

Tourism on the Verge

Roman Egger *Editor*

# Applied Data Science in Tourism

Interdisciplinary Approaches,  
Methodologies, and Applications

 Springer

# Tourism on the Verge

## **Series Editors**

Roman Egger, Innovation & Management in Tourism, Salzburg University of Applied Sciences, Urstein (Puch), Salzburg, Austria

Ulrike Gretzel, Annenberg School for Communication and Journalism, University of Southern California, Los Angeles, CA, USA

Tourism on the Verge aims to provide a holistic understanding of various phenomena that shape tourism and hospitality in profound and lasting ways, approaching research topics and practical issues from multiple perspectives. Each volume in the series will address transformations within a particular area, in order to advance both our theoretical understanding and practical applications. Books should be conceptual in nature and make highly relevant contributions to explaining these phenomena. Attention should also be drawn to cutting-edge methods, in order to stimulate new directions in tourism research. The series will publish works of the highest quality and which follow a logical structure, rather than merely presenting a collection of articles loosely related to a topic. Book editors will be asked to write a strong introductory chapter that offers a comprehensive overview of the selected topic areas / fields. Presenting a unique blend of scholarly research, the series will be an unparalleled reference source.

More information about this series at <https://link.springer.com/bookseries/13605>

Roman Egger

Editor


# Applied Data Science in Tourism

Interdisciplinary Approaches, Methodologies,  
and Applications



Springer

*Editor*

Roman Egger   
Innovation & Management in Tourism  
Salzburg University of Applied Sciences  
Urstein (Puch), Salzburg, Austria

ISSN 2366-2611

ISSN 2366-262X (electronic)

Tourism on the Verge

ISBN 978-3-030-88388-1

ISBN 978-3-030-88389-8 (eBook)

<https://doi.org/10.1007/978-3-030-88389-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

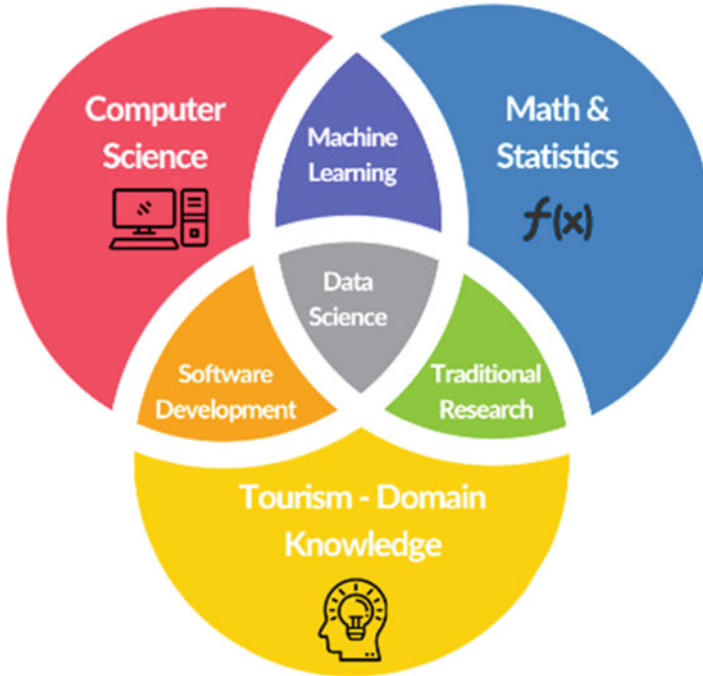
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

Data science (DS), ambient intelligence (AI), machine learning (ML), and other similar buzzwords found within this setting are used almost habitually in our linguistic repertoires nowadays; as such, I found countless working titles for this book to be insufficient, and the struggle to come up with the final title was a long-standing dilemma. Data science is a “big word” that, for most people, at least I assume, brings with it a high level of expectation. The term has an avant-garde feel to it, and there is something mystical about it, most likely due to the fact that its methods and procedures may appear complex, incomprehensible, and cryptic at first glance.

It took me about two years to wrap my head around this project since I am not by profession a data scientist myself. However, I would almost say that my *not* being a data scientist is precisely why I started this book project in the first place. As a non-computer scientist (I am a communication scientist with a strong background in tourism education and sound statistical and methodological skills), I have tried to acquire a solid knowledge base over the past few years, which has ultimately motivated me to pass on the experiences of my learning curve to others. This very fact has allowed me, at least I hope, to develop and design a book in such a way that it may be of the greatest possible use to the readers, without overwhelming them with the content or demanding too much from them. Of course, there are colleagues in my scientific community who, you could say, are more apt to publish such a book; they would be able to discuss the presented algorithms in a mathematically flawless way and, thus, could, from a computer science or mathematical perspective, provide deep insights into the subject matter at hand. However, DS also combines levels of statistics and domain knowledge in addition to the computer science aspect of it. I suspect most readers, who take the time to digest this book, will have expertise in these areas as well, either coming from the tourism industry, in which case tourism-specific expertise is their forte, or coming from academia, in which case they also most likely have an abundance of prior knowledge in the area of statistics. I personally only know a few people who are equally well-rounded in all three fields



**Fig. 1** Elements of data science

of competence (see Fig. 1), and if you are one of them, you can safely put this book aside.

There is much to suggest that DS will not be the exclusive domain of a small elite in the future. Currently, a democratization of DS is underway, and the numerous visual programming tools, including RapidMiner, KNIME, Orange, or WEKA, among others, have exemplified that extensive data analyses can be carried out even without requiring relevant programming knowledge. In addition, the rapid development of cloud-based auto ML solutions is making machine learning, text and image analysis, etc., accessible to everyone. Nonetheless, it is important not to be tempted by the act of simply loading data into one of these systems, pressing a few buttons, and publishing the results. A basic understanding of what to do, when, and how is an indispensable part of DS.

## **Purpose of the Book and Potential Audience**

The purpose of this book is to provide a first comprehensive overview of DS and its methods and procedures by presenting them in a well-structured and easy-to-understand manner. The fact that this has been applied to the context of the tourism

industry is almost incidental; everything you will read in this book is exemplified by tourism, but can also be transferred to other industries without any problems. Assumably, most readers of this book will be academics and practitioners in the tourism industry with little or no experience with DS. It is therefore essential to disentangle the individual topics, which is why you will hardly find any mathematical formulas or complex equations in each separate chapter. Instead, you will receive the most important and necessary pieces of information/knowledge needed to enable you to open the doors to the field of DS.

This book thus enables researchers to dramatically expand their methodological repertoire by moving beyond the borders of traditional quantitative methods. The amount of data available for analysis continues to increase dramatically, and the use of appropriate methods opens up an entirely new world of possibilities for analysis with far-reaching implications, also in regard to the research questions that can be asked. Naturally, tourism students are not likely to become data scientists, but they can, nonetheless, be taught individual procedures in a targeted manner. For me, it is always truly amazing to see the enthusiasm with which my master's students approach the subject and their willingness to try out various methods. The feedback from my students shows that they understand the relevance and the opportunities these new methods can provide in their later working lives and recognize it as a competitive advantage in the job market. As the Industry Insights from Data Scientists (the Q&A Session at the beginning of the book) demonstrates, the application of new forms of analytics poses major challenges to the tourism industry. Larger companies, but also DMOs, have started to add data scientists to their market research departments, or at least hire people who have the basic knowledge so that they can work on an equal footing with external data scientists and agencies. Finally, this book can also be seen as valuable reading for consultants.

## **What This Book Is Not!**

The audience of this book is not particularly geared toward data scientists, and if you mainly lack domain-specific knowledge in tourism, then this book will not help much. In this case, tourism management literature would be better suited. This book is also not satisfactory if your goal is to understand the individual algorithms in detail since, as already mentioned, this book tries to avoid complex mathematical explanations and formulas. Furthermore, the chapters do not cover the individual layers of data platforms (hardware, operating systems, data storage, and resource management) and data architectures, and the legal principles are restricted to data crawling only. Overall, this book cannot and does not intend to cover DS in its entirety; such a far-reaching and complex subject must be limited to merely its essential elements if the aim is to present an introductory and rudimentary reading.



## Features of This Book

In contrast to most edited books, which often resemble a collection of loose contributions to a topic, an anthology that pursues the task of introducing a topic step by step must ensure a very clear and stringent structure with regard to the individual chapters. As such, the first part of this book discusses some basic theoretical foundations. Thereafter, from the second part (Machine Learning) onward, each chapter contains two levels: the **theoretical basics** followed by a **practical demonstration** and/or a **research case**. The idea behind this is that the reader is guided through the analysis process by means of a practical use case in tourism. Thus, in each of these chapters, an attempt is made to bridge the theoretical background with a practical, comprehensible application. Corresponding **accompanying material** in the form of the datasets that are analyzed, Python code, R code, or other workflows, is simultaneously available on the book's GitHub profile <https://github.com/DataScience-in-Tourism/>. The reader is thus provided with a “playground,” so to say, where he or she can gain initial experience and insights into the practical implementation of a specific method. Jupyter notebooks are primarily used for this purpose, providing executable code and, at the same time, explaining the individual code elements by means of markdowns.

Generally speaking, each chapter begins with a list of **Learning Objectives**, outlining the subsequent material, and concludes with a **Service Section**, intended to give the reader a quick overview of the chapter (method), outline areas of application, show advantages and disadvantages, and point out possible combinations with other approaches. **Further readings** are also provided as literature recommendations in case one would like to deepen their knowledge of the subject matter. On that note, without further ado, happy reading!

Urstein (Puch), Austria  
August 2021

Roman Egger

# Acknowledgments

Every book project turns into a challenging task that is massively underestimated at the beginning; at least that's how I always feel, even though this is already my 19th (edited) book. That's why the help of a whole ensemble of people is invaluable to get such a project off the ground successfully. First and foremost, I would like to thank my family, both my wife and my parents, who granted me the time and countless hours I needed to complete this project, especially in addition to my regular work. Without you and your support, this project would have never even crossed my mind.

Once the idea of a project matures, the next step is to find a suitable partner with regard to the publication aspect. In this case, I was lucky enough to be working with Ulrike Gretzel on the series *Tourism on the Verge*, which we took over from Pauline Sheldon and Daniel Fesenmeier; so, it was clear to me that the book would appear in this series and, thus, at Springer. Thanks to Ulrike and Prashanth Mahagaonkar from Springer, who gave me the green light for the project, I could start compiling a list of "desired authors" and approach the first ones (exactly 12 months ago). For most of the chapters, I received a positive response straightaway, allowing me to develop the book concept with considerable authors, and, in turn, have a guarantee for qualitatively high-quality content, relatively fast. I had given very detailed specifications for each contribution, and I would like to thank each author for their highly disciplined and professional cooperation, especially considering that the project was carried out in the midst of the global COVID-19 pandemic. You brought the book to life and made it what it is!

A very special thank you goes out to Michelle Mattuzzi. Michelle proofread every chapter and with her gift for adding linguistically sharpening contributions increased the quality of the book even further. Almost all authors explicitly expressed what an outstanding job you did—thank you Michelle! I would also like to thank (Joanne) Chung-En Yu. Joanne, in addition to being the coauthor of two chapters, also helped me keep track of the formal framework of the project. Joanne, it was a special pleasure to work with you on this project as well as all our other projects!

Lastly, I would also like to thank my university and especially the dean, Eva Brucker, who recognized the relevance of data science in tourism (also for

educational purposes) and supported my project. Eva always gives me the space, means, and support to dive into new topics and express myself, and for that I am extremely grateful.

Many thanks to each and every one of you for your appreciation, support, and professional cooperation!

August 2021

# Contents

<b>Introduction: Data Science in Tourism</b> . . . . .	xxxii
<b>Industry Insights from Data Scientists: Q&amp;A Session</b> . . . . .	xlvi
<b>Part I Theoretical Fundaments</b>	
<b>AI and Big Data in Tourism</b> . . . . .	3
Luisa Mich	
<b>Epistemological Challenges</b> . . . . .	17
Roman Egger and Joanne Yu	
<b>Data Science and Interdisciplinarity</b> . . . . .	35
Roman Egger and Joanne Yu	
<b>Data Science and Ethical Issues</b> . . . . .	51
Roman Egger, Larissa Neuburger, and Michelle Mattuzzi	
<b>Web Scraping</b> . . . . .	67
Roman Egger, Markus Kroner, and Andreas Stöckl	
<b>Part II Machine Learning</b>	
<b>Machine Learning in Tourism: A Brief Overview</b> . . . . .	85
Roman Egger	
<b>Feature Engineering</b> . . . . .	109
Pablo Duboue	
<b>Clustering</b> . . . . .	129
Matthias Fuchs and Wolfram Höpken	
<b>Dimensionality Reduction</b> . . . . .	151
Nikolay Oskolkov	

<b>Classification</b> . . . . .	169
Ulrich Bodenhofer and Andreas Stöckl	
<b>Regression</b> . . . . .	209
Andreas Stöckl and Ulrich Bodenhofer	
<b>Hyperparameter Tuning</b> . . . . .	231
Pier Paolo Ippolito	
<b>Model Evaluation</b> . . . . .	253
Ajda Pretnar Žagar and Janez Demšar	
<b>Interpretability of Machine Learning Models</b> . . . . .	275
Urszula Czerwinska	
<b>Part III Natural Language Processing</b>	
<b>Natural Language Processing (NLP): An Introduction</b> . . . . .	307
Roman Egger and Enes Gokce	
<b>Text Representations and Word Embeddings</b> . . . . .	335
Roman Egger	
<b>Sentiment Analysis</b> . . . . .	363
Andrei P. Kirilenko, Luyu Wang, and Svetlana O. Stepchenkova	
<b>Topic Modelling</b> . . . . .	375
Roman Egger	
<b>Entity Matching: Matching Entities Between Multiple Data Sources</b> . . . . .	405
Ivan Bilan	
<b>Knowledge Graphs</b> . . . . .	423
Mayank Kejriwal	
<b>Part IV Additional Methods</b>	
<b>Network Analysis</b> . . . . .	453
Rodolfo Baggio	
<b>Time Series Analysis</b> . . . . .	467
Irem Onder and Wenqi Wei	
<b>Agent-Based Modelling</b> . . . . .	481
Jillian Student	
<b>Geographic Information System (GIS)</b> . . . . .	513
Andrei P. Kirilenko	

Contents	xiii
<b>Visual Data Analysis</b> .....	<b>527</b>
Johanna Schmidt	
<b>Software and Tools</b> .....	<b>547</b>
Roman Egger	
<b>Glossary</b> .....	<b>589</b>
<b>Index</b> .....	<b>599</b>

# Notes on Contributors



**Rodolfo Baggio** holds a degree in Physics (MPhys) and a PhD in Tourism Management. After an extensive experience in the information technology industry, he is currently a professor at Bocconi University (Milan, Italy), where he coordinates the digital strategies area for the Master in Economics and Tourism, and a research fellow at the Dondena Center for Research on Social Dynamics and Public Policy. He is also a professor at the Tomsk Polytechnic University in Tomsk, Russia. Rodolfo actively researches and publishes in the field of information technology and tourism on the application of complexity theory and network analysis methods for the study of tourism destinations. Rodolfo is a fellow of the Royal Geographical Society, past vice-president of IFITT—International Federation for Information Technology and Travel & Tourism, founding member of the Italian chapter of the Internet Society, and member of the Italian Physical Society. In 2017, he was awarded the Hannes Werthner Tourism and Technology Lifetime Achievement Award from IFITT.



**Ivan Bilan** is a data scientist and engineering manager who works on building production-grade NLP systems with a focus on aspect-based sentiment analysis. His educational background is in general and computational linguistics (DDPU, Ukraine; LMU, Germany), as well as technology management (CDTM, Germany). Ivan has professional experience as both a data engineer and a data scientist and works in fields such as cloud computing and real-time data pipelines as well as neural networks for NLP. The main fields of Ivan’s academic research revolve around transformer models for NLP and the application of NLP to low-resource languages. He is also the creator and maintainer of an open-source NLP knowledge resource called “The NLP Pandect.”



**Ulrich Bodenhofer** joined the School of Informatics, Communications and Media at the University of Applied Sciences Upper Austria in September 2020 as a Professor of Artificial Intelligence (AI). Prior to that, Dr. Bodenhofer held several positions all related to research, teaching, and applying machine learning. Most notably, Ulrich Bodenhofer worked as Associate Professor for Bioinformatics and Machine Learning in Sepp Hochreiter’s lab at Johannes Kepler University Linz for 12 years, but he has also been engaged in building an AI startup company, QUOMATIC.AI, since 2018. Ulrich Bodenhofer has been active in teaching machine learning and related subjects since 1999 at the University of Applied Sciences Upper Austria, Johannes Kepler University Linz, and other international universities. He has published approximately 100 scientific articles, held numerous talks at international scientific events, authored 8 publicly available software packages, and received seven scientific awards.





**Urszula Czerwinska** Born in Poland, Urszula pursued her education in France and Singapore. Currently, she is based in Paris, where she obtained her PhD. She has always been fascinated with complex systems. During her undergraduate studies, she discovered statistics and programming and was introduced to Python and R from the start. At a consulting firm, Saegus, she worked on a detailed guide, dedicated to data science professionals, on using interpretability tools with ML models and also wrote a medium post on the topic. She is now working as a data scientist specialized in NLP at the French Supreme Court in which the goal is to accelerate the open data strategy by automatizing the pseudonymization of justice documents. Occasionally, she is invited to speak about her work or some of her projects/technologies at tech conferences. Do not hesitate to contact her for a collaboration.



**Janez Demšar** teaches programming at the Faculty of Computer and Information Science as well as the Faculty of Education, both at the University of Ljubljana. His teaching activities have earned him several awards for best professor, according to students, and he is dedicated to teaching algorithmic thinking to primary school children. While pedagogy is his main research interest, he spends a lot of time developing the open-source software Orange. He focuses on visualization, data mining, machine learning, and bioinformatics.



**Pablo Duboue** is passionate about improving society through technology. He has a PhD in Computer Science from Columbia University. He splits his time between teaching machine learning, doing open research, contributing to free software projects, and consulting for startups.



**Roman Egger** is a full Professor at the Salzburg University of Applied Sciences at the Department of Innovation and Management in Tourism, where he is the head of eTourism, and head of key competencies and research. His research focus lies on new technologies in tourism and their adoption from a user-centric perspective, as well as on methodological issues in tourism research. Roman has published 19 books so far and a large number of articles and chapters in international journals and edited books, is coeditor of the *Journal of Tourism Science* (De Gruyter), series editor of “Tourism on the Verge” (Springer), and board member of a number of journals. He is a member of IFITT, Aiest, DGT, and a fellow of the ICE. Roman has received more than a dozen awards in his career.



**Matthias Fuchs, PhD**, is a full professor of tourism studies at Mid Sweden University, Östersund, Sweden. His research areas include electronic tourism (i.e., business intelligence and data mining in tourism), customer-based destination brand equity modeling, socioeconomic impact analysis, and critical tourism economics. Matthias serves on the editorial board of the *Journal of Travel Research*, *Annals of Tourism Research*, the *Journal of Hospitality & Tourism Management*, and *Tourism Analysis*. He is also associate editor of the *Journal of Information Technology & Tourism*. Matthias Fuchs has been the research track chair for the conference *ENTER@2014* and the overall chair for the conference *ENTER@2018*.



**Enes Gokce** is an NLP data scientist working in the tech industry. Enes Gokce is also a PhD student at Lifelong Learning & Adult Education program at Pennsylvania State University. His research interest is exploring causes of low math literacy by using natural language processing tools. Enes enjoys experimenting with machine learning tools, and posts articles on the Medium about them.



**Wolfram Höpken, PhD**, is a professor of business informatics at the University of Applied Sciences Ravensburg-Weingarten and director of the Institute for Digital Transformation. His main research fields are data science and big data analytics as well as ICT systems in tourism. He has been involved in several research projects in the area of knowledge discovery and big data analytics within tourism destinations as well as semantic web and seamless data interchange in tourism (EU-funded projects Harmonise, Harmo-TEN, Euromuse, and HarmoSearch). Wolfram Höpken has been the research track chair for the ENTER conference 2009 and overall chair for *ENTER@2014* as well as associate editor for the *Journal of Information Technology & Tourism*.



**Pier Paolo Ippolito** is a SAS Data Scientist and MSc in Artificial Intelligence from the University of Southampton. He has a strong interest in research areas such as data science, machine learning, and cloud development. Outside his work activities, he is a freelancer and writer for *Towards Data Science*.



**Mayank Kejriwal** is a research lead at the University of Southern California (USC) Information Sciences Institute and a research assistant professor at the Department of Industrial and Systems Engineering. Dr. Kejriwal's focus is on developing AI technology for addressing high-impact challenges, such as human trafficking and disaster response, in complex systems. His work has been funded by multiple organizations, including the US Defense Advanced Research Projects Agency, corporations, and foundations. Dr. Kejriwal has delivered talks, tutorials, demonstrations, and workshops at over 20 international academic and industrial venues in addition to having published more than 50 peer-reviewed articles, book chapters, and papers. His most recent book is *Knowledge Graphs*:

*Fundamentals, Techniques and Applications* (MIT Press, 2021). He is the recipient of a Key Scientific Challenges Award from the Allen Institute for Artificial Intelligence, a 2019 Yahoo! Faculty Research Engagement Award, and a named finalist of the AAAS Early Career Award for Public Engagement with Science.



**Andrei P. Kirilenko** is Associate Professor in the Department of Tourism, Hospitality and Event Management at the University of Florida. He received his PhD in Computer Science and held positions at the Center for Ecology & Forest Productivity, Russia; European Forest Institute, Finland; US Environmental Protection Agency laboratory, OR; Purdue University; and University of North Dakota. His research interests include big data analysis, data mining, tourism analytics, climate change impacts, and sustainability issues.



**Markus Kroner** is working as a lawyer in Salzburg, Austria, admitted in Austria and Germany. His law firm is specialized in national and international civil and commercial law, especially in the field of tourism law and real estate law. He has already published several publications in the field of tourism law and is acting as lecturer at the Salzburg University of Applied Sciences.



**Michelle Mattuzzi** is a linguistics student in the Language and Cognition Research Master Program at the University of Groningen, the Netherlands. Her main interests and research areas lie in psycho/neurolinguistics, language development, and language loss as well as multilingualism and second language acquisition. Michelle currently works as an intern/research assistant for the “Dynamic analysis of language learning in the third age (DYNAGE3)” project in collaboration with the University of Salzburg and the University of Zurich. Moreover, she is a student assistant to Prof. Dr. Roman Egger and Prof. Dr. Barbara Neuhofer

at Salzburg University of Applied Sciences and enjoys proofreading, translating, and teaching/tutoring English as well.



**Luisa Mich** is Associate Professor of Computer Science at the University of Trento, Italy. Her research interests include natural language processing and creativity techniques for requirements engineering and web presence strategies. She is the author of more than 200 papers that have appeared in journals, conferences, and workshops, and she has also served on the organizing and program committees of several conferences and workshops, including Enter, NLDB, RE, and REFSQ. She has lectured at and collaborated with several Italian and foreign universities and currently teaches tourism information systems, enterprise information systems, and web presence. Luisa Mich is a member of the IEEE Computer Society, the Association for Computing Machinery (ACM), the International Federation for Information Technology and Travel & Tourism (IFITT), and the Associazione Italiana per l'Informatica ed il Calcolo Automatico (AICA). She has been an initiator of many didactic initiatives, for introducing computer science into different degree programs at the University of Trento.



**Larissa Neuburger** is a Professor in Marketing in the Institute for Tourism, Wine Business and Marketing at the IMC University of Applied Sciences Krems. Larissa earned her PhD in Recreation, Parks and Tourism in the Department of Tourism, Hospitality, and Event Management from the University of Florida in spring 2021. Larissa has industry experience in tourism and marketing. During her doctoral studies, she worked as a graduate teaching assistant and instructor. Her research interest includes e-tourism, destination marketing, social media, and immersive technologies (augmented and virtual technology) in the context of tourism experience. She is a member of the International Federation for Information Technology and Travel & Tourism (IFITT) and the Travel and Tourism Research Association (TTRA) and a research fellow at Tourism RESET.



**Irem Onder** is an Associate Professor at the Department of Hospitality and Tourism Management at the University of Massachusetts in Amherst. She obtained her master's degree in information systems management from Ferris State University, Michigan, and her PhD from Clemson University, South Carolina, where she worked as a research and teaching assistant from 2004 until 2008. Her main research interests include information technology and tourism economics, specifically big data analysis, smart destinations, decision support systems, blockchain, and tourism demand forecasting.



**Nikolay Oskolkov** received two PhDs in Soft Condensed Matter Theoretical Physics and Statistical Physics (2007) from the University of Ulm and Moscow State University. His postdoctoral fellow positions were held at Lund University (Sweden), University of North Carolina at Chapel Hill (USA), and Technical University of Denmark (DTU Nanotech, Denmark) during the years of 2007–2012. In 2012, he changed his research direction from natural to life sciences, bioinformatics, and biomedicine and worked as a bioinformatician at the Department of Clinical Sciences at Lund University (Sweden). There, he examined the genetics of type 2 diabetes mellitus up until 2016 when he joined the SciLifeLab bioinformatics platform, a National Bioinformatics Infrastructure Sweden (NBIS) within the Long-Term Support (LTS, ex-WABI) team. He is currently working on diverse projects varying from historical DNA to biomedical multi-OMICS data analysis as well as data science.



**Ajda Pretnar Žagar** is a PhD candidate at the Department of Ethnology and Cultural Anthropology, Faculty of Arts, University of Ljubljana, and a researcher at the Laboratory for Bioinformatics, Faculty of Computer and Information Science. Her research focuses on the methodology of interdisciplinary and multidisciplinary research as well as the uses of machine learning and data mining in the humanities and social sciences. She teaches at the Higher School of Economics in Moscow,

Russia, and at the Ljubljana Doctoral Summer School at the School of Economics and Business, University of Ljubljana.



**Johanna Schmidt** received a PhD in data visualization in 2016 at the TU Wien, Austria. Her current research focuses on the visual analysis of extensive data, mainly manufacturing data from industry companies and time series data. Johanna Schmidt is the head of the group Visual Analytics at the VRVis research center (VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH) in Vienna.



**Svetlana Stepchenkova** is Associate Professor at the Department of Tourism, Hospitality and Event Management at the University of Florida. Her research interests are in the area of marketing communications, branding, and positive image building. She studies tourism behavior and the effectiveness of destination promotion in situations of strained bilateral relations between nations. She is also interested in usability of user-generated content for managerial decision-making in destination management.



**Andreas Stöckl** is head of the Digital Media Department at the School of Informatics, Communications and Media at the University of Applied Sciences Upper Austria. After studying mathematics at the University of Linz, he was first an assistant at the Institute of Mathematics before becoming a professor at the University of Applied Sciences Upper Austria. He has published several books and scientific articles on data science topics and, since 1996, has also established several startup companies, like Cyberhouse GmbH, Contextity AG, and 506 Data & Performance GmbH (506.ai).



**Jillian Student** is fascinated by human–environment interactions and the uncertainty that results from such ongoing feedbacks. In this capacity, the tourism sector is of particular interest as it relates to a variety of other economic, policy, and NGO sectors. She developed a dynamic approach to understanding emerging vulnerabilities in tourism and applied it to coastal tourism. This approach included agent-based modeling, which helped capture key factors that can contribute to ecological and socioeconomic vulnerability. Since completing her dissertational work, she has worked as a postdoctoral researcher at the Environmental Policy Group in Wageningen. She will continue her collaboration there with her new role of furthering transdisciplinary research at the Wageningen Institute for Environment and Climate (WIMEK) research graduate school.

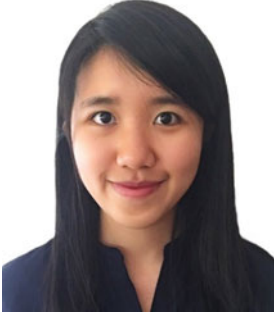


**Luyu Wang** received her PhD in Recreation, Parks and Tourism from the University of Florida in 2021. Luyu worked as a teaching assistant at the Department of Tourism, Hospitality and Event Management. Her research interests include tourism and social media and tourism data analytics. She is also interested in travel experiences in nature parks based on user-generated content as well as cultural differences and tourism.



**Wenqi We** is a second-year doctoral student from Isenberg School of Management at the University of Massachusetts in Amherst. Her research interests include data analytics and technology applications in hospitality and tourism management.





**Joanne (Chung-En) Yu** is a research assistant and a master graduate of innovation and management in tourism at the Salzburg University of Applied Sciences. Her research interests center on destination image, consumer experience, human–robot interaction, social media data analysis, and emerging technologies in tourism and hospitality.

# Abbreviations and Acronyms

ABM	Agent-based model
ADF	Augmented Dickey–Fuller test
ADR	Average daily rate
AI	Artificial intelligence
AIC	Akaike’s information criterion
AICc	Corrected Akaike’s information criterion
ANN	Artificial neural network
API	Application programming interface
ARIMA	Autoregressive integrated moving average
AUC	Area under curve
Auto ML	Automated machine learning
BERT	Bidirectional encoder representations from transformer
BI	Business intelligence
BIC	Bayesian information criterion
BILSTM	Bayesian bidirectional long short-term memory
BOW	Bag-of-words
CA	Classification accuracy
CBOW	Continuous-bag-of-words
CNN	Convolutional neural network
CRISP-DM	Cross-industry standard process for data mining
DARPA	Defense Advanced Research Projects Agency
DBSCAN	Density-based spatial clustering of applications with noise
DIG	Domain-specific insight graph
DL	Deep learning
DS	Data science
EA	Error analysis
EDA	Exploratory data analysis
EMA	Exploratory modeling and analysis
EPA	Evaluation-potency-activity
ER	Entity resolution
ETS	Error trend seasonal

EU	European Union
FE	Feature engineering
FFN	Feedforward neural network
FLOPS	Floating-point operations per second
FN	False negative
FP	False positive
GAN	Generative adversary network
GBM	General boosted model
GDP	Gross domestic product
GDPR	General Data Protection Regulation
GIS	Geographic information system
GloVe	Global vector embeddings
GMM	Gaussian mixture model
GPT	Generative pre-trained transformer
GPT-3	Generative pre-trained transformer 3
GPU	Graphics processing unit
GWR	Geographically weighted regression
ICT	Information and communications technology
IE	Information extraction
IoT	Internet of things
IR	Information retrieval
ISI	Information Sciences Institute
KB	Knowledge base
KG	Knowledge graph
KGC	Knowledge graph construction
KNN	K-nearest neighbors
LASSO	Least absolute shrinkage selector operator
LDA	Latent Dirichlet allocation
LDA	Linear discriminant analysis
LIME	Local interpretable model-agnostic explanation
LOD	Linked open data
LOO	Leave-one-out
LSA	Latent semantic analysis
LSI	Latent semantic indexing
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MDA	Mean decrease accuracy
MDS	Multidimensional scaling
ML	Machine learning
MSE	Mean squared error
NDCG	Normalized discounted cumulative gain
NER	Named entity recognition
NFL	No free lunch theorem

NLG	Natural language generation
NLP	Natural language processing
NLTK	Natural language toolkit
NLU	Natural language understanding
NMF	Non-negative matrix factorization
NN	Natural network
ODD	Overview, design concepts, and details
ODD+D	Overview, design concepts, and details + decision-making
OTA	Online Travel Agency
OWL	Web ontology language
PCA	Principal component analysis
POI	Point of interest
POS	Part of speech
QA	Question answering
RAM	Random access memory
RDF	Resource description framework
RDFS	Resource description framework schema
RF	Random forest
RMSE	Root mean square error
RNN	Recurrent neural network
ROC	Receiver operating characteristic
SA	Sensitivity analysis
SES	Single exponential smoothing
SHAP	Shapley additive explanations
SMBO	Sequential model-based optimization
SVM	Support vector machine
SW	Semantic web
TF-IDF	Term frequency—inverse document frequency
TN	True negative
TP	True positive
TPE	Tree-structured Parzen estimator
TPOT	Tree-Based Pipeline Optimization Tool
TPU	Tensor processing unit
tSNE	t-Distributed stochastic neighbor embedding
UGC	User-generated content
UMAP	Uniform manifold approximation and projection
US	United States
USC	University of Southern California
VAR	Vector autoregression
WSD	Word-sense disambiguation
XAI	Explainable artificial intelligence
YAGO	Yet another great ontology

# Introduction: Data Science in Tourism

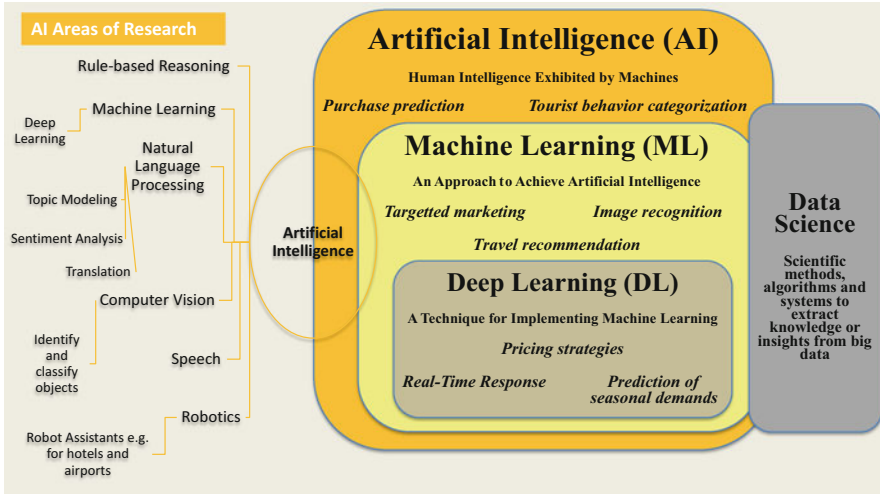
## A Brief Introduction and the Structure of This Book

Roman Egger

Artificial intelligence (AI), machine learning (ML), data mining, big data, and smart data are just some of the many buzzwords that have taken on a dominant position in science, business, and the media (Papp et al., 2019) in recent years. Although these key phrases may seem like merely a hype, there is, nonetheless, a certain magic to them mainly because they have undoubtedly permeated our everyday lives, whether it be in one's private or professional sphere. The rapidly advancing digitalization of our society has laid the foundations for this (Egger, 2007; Neuburger et al., 2018); increasing computing power, greater storage capacities, faster Internet connections, the rapid development of powerful algorithms, and the availability of vast amounts of data for analysis purposes are just some of the driving forces that have and continue to enable us to apply new analytical methods and generate useful knowledge for science, business, and, ultimately, society in general.

The multitude of the respective names often causes confusion, and, in fact, the individual concepts frequently overlap, making it difficult to distinctly separate one expression from another in terms of definition. Artificial intelligence (AI) is generally regarded as an umbrella term to which all other subject areas are subordinated, as can be seen in Fig. 1. Kok et al. (2009) define AI as an area of study that is concerned “with the development of computers able to engage in human-like thought processes such as learning, reasoning, and self-correction” (p. 271). As such, the AI areas of research include rule-based reasoning, machine learning (ML), with deep learning being a further layer thereof, subordinate ML techniques, natural language processing (NLP), computer vision, speech analytics, and robotics.

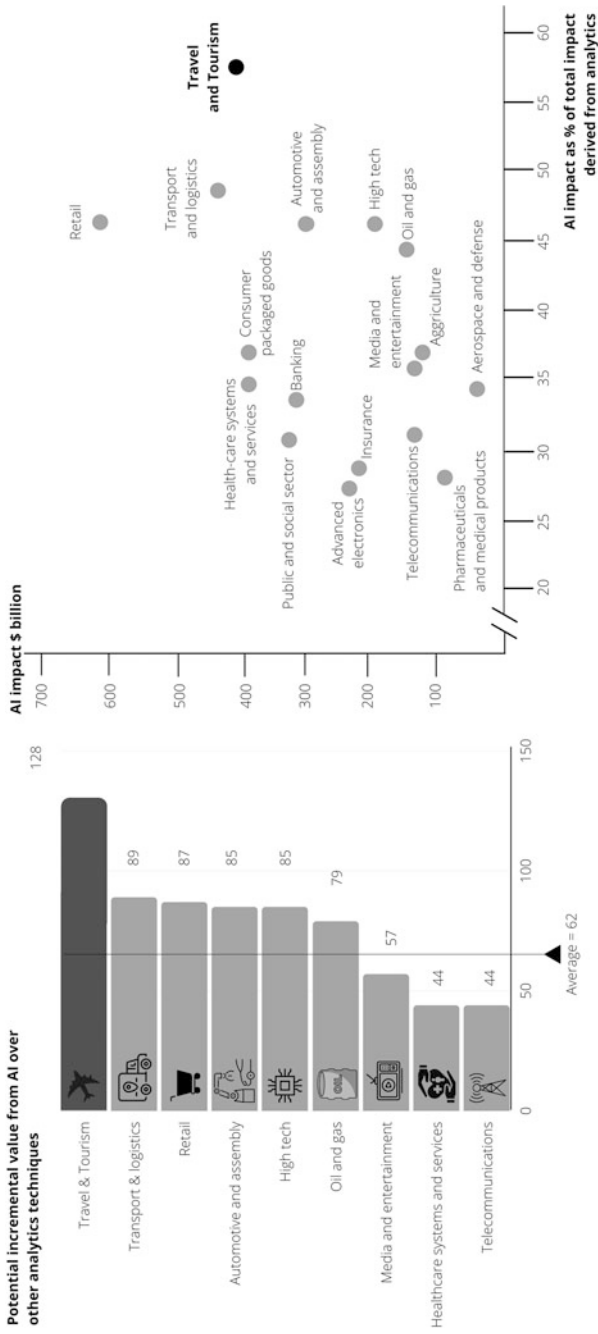
AI can be seen as a key driver of innovative solutions for businesses of all sizes and industries (Mich, 2020), including the tourism domain. Tourism is a highly complex industry characterized by a multitude of service providers, intermediaries,



**Fig. 1** Artificial intelligence and its various research niches. Source: adapted and modified from Vollmer (2018)

and customers, which requires a great deal of communication and coordination (Boes et al., 2015; Hofstaetter & Egger, 2009). At the same time, the co-creation of memorable experiences necessitates a comprehensive understanding of the needs, requirements, and wishes of individual travelers to support them with personalized recommendations in real time (Xiang & Fesenmaier, 2017). To try and cater to these needs, a type of tourism known as “smart tourism” (Gretzel et al., 2015), in which (big) data and the power of AI are used to enable new forms of (co-)value creation, sustainability, and well-being through technology, has formed. Gretzel and her colleagues define three levels of smart tourism where AI can fully exploit its potential. In a world where data is continuously recorded by sensors, i.e., our behavior is tracked and traced by the multiple devices we use in both the real and digital spheres, technology can be understood as an infrastructure. Therefore, intelligent technology, which should be able to grasp a certain situation and react accordingly (Worden et al., 2003), comprises the backbone of smart destinations and the first level of smart tourism. It bridges the physical and the digital world (Koo et al. 2016), where the Internet is transformed into an Outernet (Neuburger et al., 2018). The second level of smart tourism is that of smart experiences. This is where AI can enhance technology-mediated tourism experiences by personalizing and adapting the service to the guest’s preferences using context-sensitive and real-time data. Lastly, the third level is concerned with the highly dynamic tourism ecosystem in which data are collected, shared, harmonized, and synchronized between all stakeholders in order to be processed via AI.

According to a report by McKinney (2018), AI can be substantially beneficial to those sectors of the economy where sales and marketing, in particular, are driving the value. Tourism can indeed be viewed as such an industry, explaining why AI has extraordinary potential for tourism (see Fig. 2 below).



**Fig. 2** AI’s potential to create value across sectors. Source: adapted and modified from McKinsey Global Institute analysis (2018)

In the remainder of this book, the term AI itself will only be used in a very limited way since, rather than referring to this umbrella term, specific reference to its individual components will be made.

Data science (DS), as a significant aspect of AI, can be seen as a comprehensive set of methods, algorithms, and systems that are applied to various sectors of an interdisciplinary field. As depicted in Fig. 3, DS combines computer science, mathematics and statistics, and domain-specific (tourism) knowledge to gain valuable insights from large sets of structured, semi-structured, and unstructured data (George et al., 2016). This consequently helps to explain and understand phenomena and processes in the present and, with its predictive power, to a certain extent, also the future.

This data-driven approach is often criticized as undermining the meaning of theory or making theory obsolete since researchers seem to no longer be dependent on existing theories (Egger & Yu, 2022a). However, can research even exist without theorizing and hypothesizing, and is looking at correlations and patterns in the data

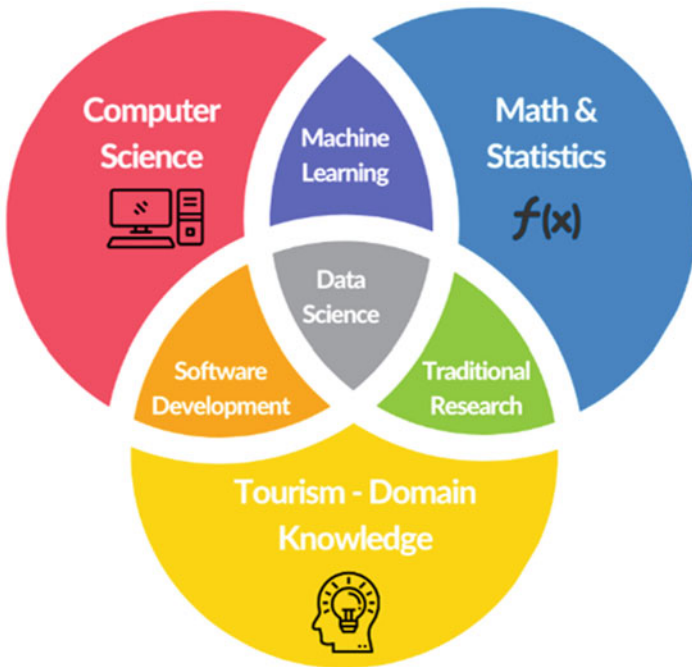


Fig. 3 Data science—an interdisciplinary field



enough to understand, explain, and predict? Over the past few years, there have been heated discussions about possible epistemological paradigm shifts, and, indeed, a certain potential for change in such a shift is observable. In Chapter 2 (Epistemological challenges), the fact that radical positions and black-and-white thinking should no longer be expedient here and, additionally, that this touchy topic needs to be viewed in a very differentiated and individual way will be explained in more detail.

In any case, there is no doubt about the power of change that DS has brought, and continues to bring, along with it (Steinberg & Aronovich, 2020). After all, the goal of data science, the data-based improvement of decision-making capabilities, is used in vast areas of modern society (Power, 2016; Provost & Fawcett, 2013). Seen from a micro-level perspective, and encompassing the core topics of this book, DS is concerned with big data, data mining, natural language processing, machine learning, image analysis, social network analysis, agent-based modeling, and data visualization, among others (Egger, 2022a).

## **Data Science and Tourism**

Tourism is an extraordinarily interdisciplinary field of research in which sociologists, geographers, economists, communication scientists, psychologists, computer scientists, etc., come together in various ways. Each of these disciplines brings its own repertoire of methods to the table, providing tourism research with a colorful bouquet of methodological options for observing and analyzing a phenomenon from a wide variety of perspectives (Egger & Yu, 2022b). Just as classical methods of empirical social research have gradually professionalized tourism research, DS methods are likely to be increasingly applied in order to further understand and solve tourism issues. Each method has its individual strengths and weaknesses and is predestined to add value to specific problems and issues. By opening up this treasure chest of methods, tourism can discover previously unknown patterns and correlations, analyses can be carried out in real time, and even deeper insights can be revealed in order to better understand and explain phenomena, systems, processes, structures, and behavior, to mention but a few. Above all, the predictive power of some particular approaches makes it possible to answer future-oriented questions (Weihs & Ickstadt, 2018). Examples of data science applications in tourism include route optimization, predictive analysis and forecasting, personalization and recommendation, opinion mining and sentiment analysis, alerting and monitoring systems, and much more (Egger, 2022a). This leads to an improved basis for decision-making

and better planning. Furthermore, services can be optimized and individualized service provision becomes possible, resulting in better customer experience and, consequently, to competitive advantages.

In addition to the existing methods, models, and algorithms, the quality and quantity of data are of particular importance (Taleb et al., 2018). Tourism is an information-intensive industry (Buhalis & Amaranggana, 2015; Egger & Herdin Thomas, 2007), and, accordingly, a wide array of different data sources and data formats are available. Since we as humans are tracked and traced in both the real and the physical world, each and every individual produces vast amounts of data per day without explicitly knowing it. The most diverse sensors continuously measure and deliver data, and through our active participation in social networks, we contribute to the growth of user-generated content (Nicholas Wise & Hadi Heidari, 2019). As such, data from systems and processes are unintermittedly recorded and issued for analysis in a wide variety of formats. They are available as numerical values, as text, images, or videos, have geographical or temporal references, or describe, as meta-data, other data (further details on the topic of data extraction and collection can be found in Chapter 5). However, the fundamental question is what can we and do we really want to extract from the data, which questions can we answer, and which problems can be solved with the insights gained?

At this point, it is clear that a completely theory-free approach will never be effective, and the importance of domain-specific knowledge becomes apparent. Typically, raw data can rarely be used without preprocessing, and meaningful features have to be engineered and selected to serve as input data (see Chapter 7). Therefore, whoever decides to work in this area carelessly and without well-founded preconsiderations will begin his or her analysis with bad data and, in turn, obtain poor or wrong results. The same applies to the selection and tuning of algorithms.

In order to get an overview of the current research landscape, the *Scopus* and *Web of Science* databases were searched by using the term “tourism” in combination with those terms that are also central to this book: “machine learning,” “sentiment analysis,” “topic modeling,” “network analysis,” “support vector machine,” etc. This resulted in a total of 2,832 papers, which were then analyzed further with VOSViewer (van Eck & Waltman, 2010). Figure 4 provides an overview of the particular topics, along with their relevant size and importance, and shows how they are related to each other.

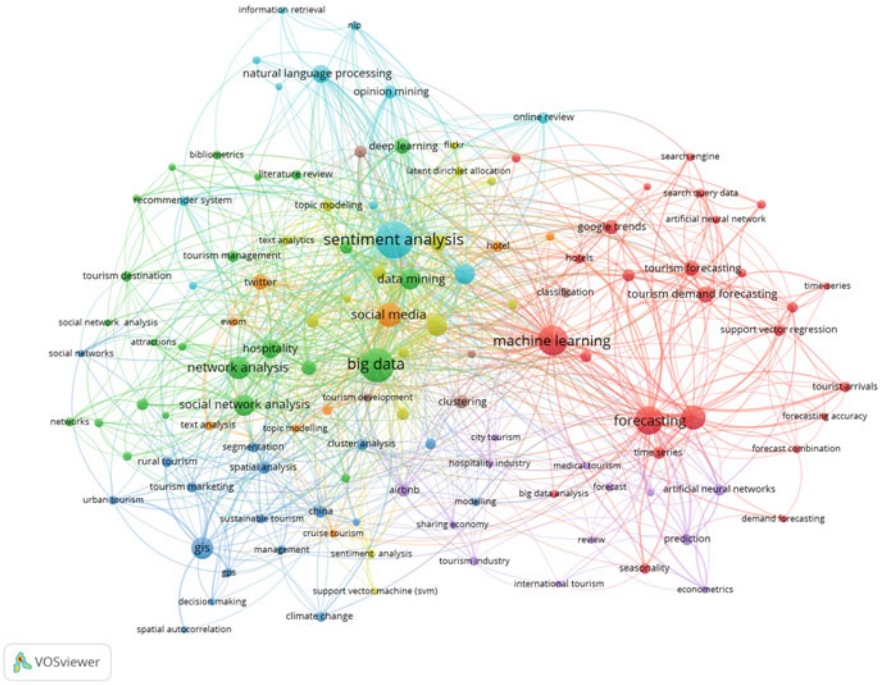


Fig. 4 Bibliographic analysis—data science in tourism

At first glance, we can clearly observe four major indicators in relation to tourism, namely, “sentiment analysis,” “big data,” “machine learning,” and “forecasting,” with “social media,” “natural language processing,” “gis,” and “(social) network analysis” closely lagging behind. On the left-hand side, the significance of “(social) network analysis” and its connection to terms like “tourism destination,” “destination management,” “tourism demand,” and “tourism forecasting” can be noted, while on the right-hand side, “time series,” “forecasting,” “tourist arrivals,” etc., seem to be of large significance for tourism. Specific links to important terms such as “big data,” “Google Trends,” and “seasonality,” but also to methods and algorithms like “ARIMA” and “support vector regression,” can also be seen. Moreover, an additional cluster on top mentions various NLP methods, placing a dominant role on “sentiment analysis.” As expected, the topic “machine learning” also presents itself as very central and powerful, with links to almost all other areas including “recommender system,” “forecasting,” “smart tourism,” and “natural language processing.” The field of “topic modeling,” in contrast, is rather underrepresented. Although it is evident that topic modeling can be found in combination with sentiment analysis and that “latent Dirichlet allocation (LDA)” is the preferred applied algorithm (Egger,

2022b), this analysis method still seems to have found relatively little use in tourism. Similarly, in the lower left-hand corner, the term “gis,” despite being linked to “social networks,” “climate change,” and “destination marketing,” among others, is otherwise relatively isolated. Overall, one important aspect highlights the fact that the close relationships of data sources such as “Tripadvisor,” “Twitter,” and “social media” with other notable items suggests the preferential analysis of user-generated content in tourism. Thus, keeping all these statements in mind, this book seeks to present most of the relevant individual concepts mentioned above in a systematic and sequential manner. A descriptive overview and structure of this attempt is provided in the following section below.

## **Data Science in Tourism and the Structure of This Book**

To start off this book, it is important to understand current problems in society along with their relevance in research as well as to draw attention to these issues. By interviewing data scientists from different areas of the tourism industry, the **Q&A Session** aims to outline the true significance of the relationship between data science and tourism, to address its challenges, and to capture a glimpse into the future of this interdisciplinary field.

### ***Chapters 1–5: Theoretical Fundamentals***

The first group of chapters then goes on to discuss relevant **fundamentals on a meta-level**. In **Chapter 1**, *Luisa Mich* dives into the topic of **AI and Big Data in Tourism** and lays down the terminological foundations thereof, defining terms such as AI, ML, big data, and DS and emphasizing their importance for tourism.

As such, it comes as no surprise that big data, the availability of powerful hardware, and the rapid development of new algorithms have ignited a transformation within the entire research process. This paradigm shift brings about **epistemological challenges**, which are discussed in **Chapter 2** by Roman Egger and Chung-En Yu.

The successful completion of DS projects requires competencies in computer science and statistics as well as a high proportion of domain-specific knowledge; yet, rarely do organizations or individuals possess all three of these competencies sufficiently and satisfactorily. Therefore, Roman Egger and Chung-En Yu call for a higher degree of **interdisciplinarity in data science**, which they elaborate on and justify in **Chapter 3**.

New forms of data generation, processing, analysis, and interpretation also go hand in hand with far-reaching ethical requirements and issues. In **Chapter 4, Data science and ethics**, Roman Egger, Larissa Neuburger, and Michelle Mattuzzi explore these challenges in more detail.

Every DS project requires data as a basis for analysis and, in contrast to classical research methods which often use survey data for quantitative analyses, DS projects primarily use the vast amount of data available online. Web mining and data crawling is the first step for many DS projects to crawl user-generated content and data from websites and retrieve data via application programming interfaces (APIs) or from open data sources. Thus, in **Chapter 5**, written by Roman Egger, Markus Kroner, and Andreas Stöckl, an overview of the different types of data in tourism is specified, legal aspects of web crawling are discussed, and tools and packages available for **web scraping** are presented. Furthermore, a practical demonstration illustrates the process of how to crawl and parse a website with Python.

## *Chapters 6–14: Machine Learning*

As **machine learning (ML)** is one of the central elements of data science, the next several chapters aim to zoom in on this field, outlining the ML process and presenting and discussing the most important algorithms and methods in further detail. Firstly, however, **Chapter 6, Machine learning: a primer**, is intended to provide an introduction/overview of this area. *Roman Egger* thus illustrates the intuition behind machine learning and discusses the different ML paradigms and algorithms and their application in tourism.

Following up on this, *Pablo Duboue* explicitly addresses the topic of **feature engineering** in **Chapter 7**. He shows that a “dialogue between the data scientist and the computer” is necessary in order to implement domain expertise into the data and its preparation. Pablo explains the procedural steps of feature selection, expansion, and homogenization. Moreover, in the practical demonstration, he walks us through a pricing example using Airbnb data.

**Chapter 8** is the first chapter that deals with a concrete application of ML algorithms. As such, *Matthias Fuchs* and *Wolfram Höpken* turn to **clustering** approaches, which are a part of **unsupervised ML**. They first explain the conceptual foundations of the most important clustering approaches and then apply clustering methods to an example with tourism data in a step-by-step demonstration using RapidMiner.

Besides clustering, **dimensionality reduction** is the second important category under the **unsupervised ML** algorithms. In **Chapter 9**, *Nikolay Oskolkov* takes the reader into the realm of high-dimensional data, explains the phenomenon of the “curse of dimensionality,” and presents methods such as PCA, tSNE, and UMAP. The practical demonstration uses annotated Instagram images of Austria that were

posted by tourists and shows how to perform dimensionality reduction on the 100-dimensional Doc2Vec vectors.

Thereafter, *Ulrich Bodenhofer* and *Andreas Stöckl* discuss the most important **supervised ML** algorithms in the next two chapters. In **Chapter 10**, they focus on **classification**, presenting the most relevant methods and explaining the evaluation of classification results. In the practical demonstration, they apply different algorithms to classification tasks based on tourism data.

In addition to classification, **regression** is one of the most commonly used **supervised ML** approaches. In **Chapter 11**, once again *Andreas Stöckl* and *Ulrich Bodenhofer* present widely used algorithms such as linear regression, regression trees, random forests, gradient tree boosting, support vector machines, and neural networks. The chapter concludes with a practical demonstration of the application of the algorithms on the basis of a tourist use case.

ML algorithms have so-called hyperparameters, settings that can be used to tune the performance of the algorithms. In **Chapter 12**, *Pier Paolo Ippolito* describes the intuition behind **hyperparameter tuning** and presents different tuning possibilities. Using the “Flight Delays and Cancellations” dataset, he takes us through a concrete example and discusses the requirements and challenges of hyperparameter tuning.

Closely related to hyperparameter tuning is the evaluation of ML models. *Ajda Pretnar* and *Janez Demšar* present the possibilities of **model evaluation**, explain the evaluation scores in detail, and point out the threats of overfitting or underfitting your models in **Chapter 13**. In the practical demonstration, they present a typical model evaluation workflow based on a hotel bookings dataset using Orange 3.

Indisputably, ML is significant for offering us far-reaching possibilities to gain interesting insights from data. At the same time, the **interpretation of machine learning** models often poses challenges as it is hard to understand how certain results were obtained. *Urszula Czerwinska* addresses this problem in **Chapter 14** and demonstrates how a machine learning model can be explained. Urszula also uses the “hotel booking demand” dataset to demonstrate her theoretical explanations in a practical example.

## *Chapters 15–20: Natural Language Processing*

In addition to the processing of numerical data, natural language processing has also emerged as a particularly relevant application area for data science. In this way, *Roman Egger* and *Enes Gokce* present the basics of text analysis in **Chapter 15**, **Introduction to natural language processing**. This chapter serves as a preface to the following chapters, which deal explicitly with specific areas of NLP. Since all of the methods presented in the following text analytics chapters require preprocessed data, this chapter is primarily devoted to the preprocessing procedure.

In many cases, one wishes to perform further calculations with text or apply algorithms that require numerical values as input data. In **Chapter 16**, *Roman Egger* discusses possibilities of such **text representations and word embeddings** and illustrates how text data can be vectorized. He starts by presenting very simple, but often sufficient, methods like BOW and TF-IDF, then moves on to more powerful embedding approaches like Word2Vec, Glove, Fasttext, and ELMO, and concludes with a discussion on BERT, the current state-of-the-art embedding approach.

**Sentiment analysis** is undoubtedly one of the most important methods when it comes to text analysis as it tries to measure and quantitatively capture people's feelings of joy, anger, sadness, and so on. In **Chapter 17**, *Andrei P. Kirilenko*, *Luyu Wang*, and *Svetlana O. Stepchenkova* present the most important approaches used in sentiment analysis and illustrate typical applications in tourism. Finally, the individual methodological steps can be noted during the practical demonstration with two research cases.

**Topic modeling** is another very commonly used text analysis technique. In **Chapter 18**, *Roman Egger* discusses the main intuition behind the most relevant topic modeling approaches. He contrasts classical methods such as latent Dirichlet allocation and non-negative matrix factorization but also presents alternative approaches such as CorEX, Top2Vec, and BERTopic, which have hardly been used in tourism research so far. At the same time, he points out the numerous hurdles and pitfalls that can lead to poor results with topic modeling. In the practical demonstration, different algorithms are applied in order to extract topics from a dataset relating to Airbnb experiences.

A common problem in text analysis involves entity matching between multiple data sources. *Ivan Bilan* describes the theoretical foundations of this approach and the steps necessary to build an **entity matching** pipeline in **Chapter 19**. At the end of this chapter, he additionally outlines how to engineer an end-to-end entity matching pipeline.

The development of **knowledge graphs** has progressed enormously over the past few years, making an increasingly significant mark on tourism. Thus, *Mayank Kejriwal* illustrates the fundamentals of knowledge graphs in **Chapter 20** and discusses their growing relevance and possible application areas in the field of tourism. In the practical demonstration and research case, he comprehensibly describes the implementation of a knowledge graph.

## ***Chapters 21–26: Additional Methods***

After numerous chapters focusing on the topics of machine learning and natural language processing, the next section pays more attention to methods that have been developed for very specific analysis tasks. One of them being network analytic

methods, which provide insights into complex systems by describing their structures and characteristics. In **Chapter 21**, *Rodolfo Baggio* introduces us to the basic concepts and methods of **network analysis** and shows which skills are necessary to visualize networks and compute main measures. In the practical demonstration, he illustrates how to perform a network analysis with the help of Gephi.

**Time series analysis** is an additional set of methods that has a high value in tourism, for example, when it comes to analyzing and forecasting tourism demand. Thus, *Irem Onder* illustrates univariate and combined forecasting methods in **Chapter 22** and demonstrates how the forecasting accuracy of time series models can be evaluated. In the practical demonstration and research case, a step-by-step time series analysis is provided.

Like other economic sectors, tourism is characterized by a high degree of complexity, interactions, heterogeneity, nonlinearity, and uncertainty. **Agent-based modeling** allows to analyze the complex interactions between people and their environment. In **Chapter 23**, *Jillian Student* outlines the opportunities presented by agent-based modeling and provides examples that guide the reader through the entire analysis process, from developing the research question to interpreting the analysis. Her practical demonstration and research case give the reader a clear illustration of the entire process.

Spatial data play a special role and are, naturally, of particular relevance in the context of tourism. In **Chapter 24**, *Andrei P. Kirilenko* describes different types of data and the main functions of **GIS analysis** and discusses the most important concepts of spatial data analysis. Using a practical example, Andrei exemplifies how GIS data can be processed.

The visualization of data plays a significant role, not only for the presentation of results. As such, *Jonanna Schmidt* describes in **Chapter 25** why, when, and how **visual data analysis** can be applied within the data analysis workflow. Moreover, she also provides us with an excellent overview of the latest data visualization libraries and software solutions.

Countless frameworks, **software solutions**, and tools are available on the market to perform DS tasks, i.e., helping to process, analyze, and visualize data. Finally, in the last **Chapter (26)**, *Roman Egger* presents the most important solutions for tourism cases. In addition, *all authors* have compiled the most important tools for their topics in their respective chapters, which are presented in the form of an overview table at the end of this chapter. Figure 5 provides an overview of the structure and outline of the book.



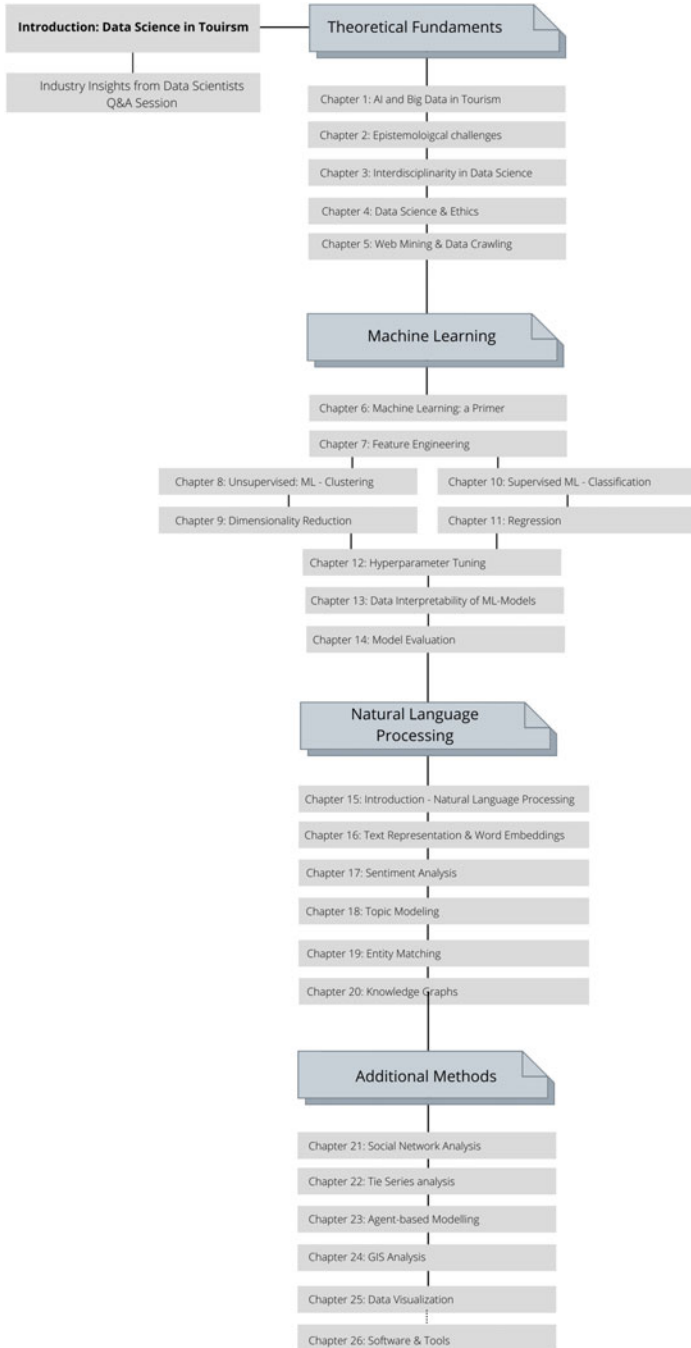


Fig. 5 Structure and outline of the book

While the first chapters focus on discussing the content in purely theoretical terms, all other chapters (from the machine learning section onward) contain two parts. On the one hand, the theoretical basis, where necessary background information is conveyed, will always be established at the beginning of each chapter. Thereafter, a practical demonstration and/or a research case with a complete step-by-step walkthrough of the process will follow. Thus, the reader is not only provided with specific knowledge of the topic but can also practice/go through the respective methods and approaches using an example. In addition, most of the chapters come with supplementary electronic material. For instance, sample datasets and corresponding parts of code can be accessed at <https://github.com/DataScience-in-Tourism/>. At the end of each chapter, a “service section” provides a quick overview of the topic by outlining typical areas of application, comparing advantages and disadvantages, and listing related and complementary methods.

## Conclusion

Data science has brought marvelous opportunities to many industries, and tourism is no exception. Although tourism is known as an interdisciplinary field, spanning across sociology, economics, geography, psychology, and the communication sciences, tourism researchers have long been constrained by the classical repertoire of research methodologies (Egger & Luger, 2015). Besides the widely applied quantitative and qualitative approaches, advancements, especially in quantitative methods, could be observed as time passes. Nowadays, in this era of digitization, data comes in new unstructured forms, and this, along with traditionally structured datasets, has resulted in the rise of big data. Meanwhile, advancements in computing and the rapid development of algorithms have led to the emergence of advanced analytics, going beyond conventional business intelligence to gain deeper insights and make future predictions.

All in all, data science is more than just a plethora of new methods and tools that can elevate the typical ways of doing empirical research and allow researchers to find answers to previously unknown questions; rather, it is a rich, interdisciplinary area, aiming to obtain business-related insights on data and containing a unique set of methods that can, in the future, most certainly be sufficiently utilized in tourism. Nonetheless, DS is yet to be embraced by tourism scholars to its fullest potential, in part due to the vastness, messiness, and unstructured nature of the data that fuels confusion and uncertainty. At the same time, because DS has altered epistemological foundations to a certain extent, the interplay between data science and theory deserves much more attention and approval.

## References

- Boes, K., Buhalis, D., & Inversini, A. (2015). Conceptualising smart tourism destination dimensions. In *Information and communication technologies in tourism 2015* (pp. 391–403). Cham: Springer. [https://doi.org/10.1007/978-3-319-14343-9\\_29](https://doi.org/10.1007/978-3-319-14343-9_29)
- Buhalis, D., & Amaranggana, A. (2015). Smart tourism destinations enhancing tourism experience through personalisation of services. In *Information and communication technologies in tourism 2015* (pp. 377–389). Cham: Springer. <https://doi.org/10.1007/978-3-319-14343-9\textunderscore>
- Egger, R. (2007). Cyberglobetrotter - Touristen im Informationszeitalter. In R. Egger & Herdin Thomas (Eds.), *Tourismus Herausforderung Zukunft* (pp. 382–393). Münster: LIT.
- Egger, R. (2022a). Data Science in Tourism. In D. Buhalis (Ed.), *Encyclopedia of Tourism Management and Marketing*. [S.l.]: Edward Elgar Publishing.
- Egger, R. (2022b). Topic Modeling. In R. Egger (Ed.), *Tourism on the verge. Applied data science in Tourism* (pp. 375–403). Cham: Springer.
- Egger, R., & Herdin Thomas (Eds.) (2007). *Tourismus Herausforderung Zukunft*. Münster: LIT. Retrieved from <https://scholar.google.com/citations?user=9jwau7qaaaaj&hl=en&oi=sra>
- Egger, R., & Luger, K. (Eds.) (2015). *Tourismus und mobile Freizeit. Lebensformen, Trends, Herausforderungen*: LIT.
- Egger, R., & Yu, C.-E. (2022a). Epistemological Challenges. In R. Egger (Ed.), *Tourism on the verge. Applied data science in Tourism* (pp. 17–34). Cham: Springer.
- Egger, R., & Yu, C.-E. (2022b). Data Science and Interdisciplinarity. In R. Egger (Ed.), *Tourism on the verge. Applied data science in Tourism* (pp. 35–49). Cham: Springer.
- George, G., Osinga, E. C., Lavie, D., & Scott, B. A. (2016). Big data and data science methods for management research. *Academy of Management Journal*, 59 (5), 1493–1507. <https://doi.org/10.5465/amj.2016.4005>
- Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: foundations and developments. *Electronic Markets*, 25(3), 179–188. <https://doi.org/10.1007/s12525-015-0196-8>
- Hofstaetter, C., & Egger, R. (2009). The importance and use of weblogs for backpackers. *Information and communication technologies in tourism 2009* (pp. 99–110).
- Kok, J. N., Boers, E. J. W., Kusters, W. A., van der Putten, P., & Poel, M. (2009). Artificial intelligence: Definition, trends, techniques and cases. *Artificial Intelligence*, 1, 270–299.
- McKinsey Global Institute analysis (2018). *Artificial intelligence (AI) has the potential to create value across sectors*. Retrieved from <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020>

- Mich, L. (2020). Artificial Intelligence and Machine Learning. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-Tourism* (pp. 1–21). Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-030-05324-6textunderscore>
- Neuburger, L., Beck, J., & Egger, R. (2018). The ‘Phygital’ Tourist Experience: The use of augmented and virtual reality in destination marketing. In M. A. Camilleri (Ed.), *Tourism Planning and Destination Marketing*. Emerald Publishing Limited. <https://doi.org/10.1108/978-1-78756-291-220181009>
- Nicholas Wise, & Hadi Heidari (2019). Developing smart tourism destinations with the internet of things. In *Big data and innovation in tourism, travel, and hospitality* (pp. 21–29). Singapore: Springer. [https://doi.org/10.1007/978-981-13-6339-9\\_2](https://doi.org/10.1007/978-981-13-6339-9_2)
- Papp, S., Weidinger, W., Meir-Huber, M., Ortner, B., Langs, G., & Wazir, R. (Eds.) (2019). *Handbuch data science: Mit datenanalyse und machine learning Wert aus Daten generieren*. München and © 2019: Hanser. <https://doi.org/10.3139/9783446459755>
- Power, D. J. (2016). Data science: Supporting decision-making. *Journal of Decision Systems*, 25(4), 345–356. <https://doi.org/10.1080/12460125.2016.1171610>
- Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking* (1st ed.). Sebastopol, CA: O’Reilly Media. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=619895>
- Steinberg, D., & Aronovich, E. (2020). Thoughts on data science in business and industry. *Applied Stochastic Models in Business and Industry*, 36.1, 36–40. Retrieved from [https://www.researchgate.net/profile/eddie-aronovich-2/publication/338692827\\_thoughts\\_on\\_data\\_science\\_in\\_business\\_and\\_industry/links/5e7955674585158bd501b45a/thoughts-on-data-science-in-business-and-industry.pdf](https://www.researchgate.net/profile/eddie-aronovich-2/publication/338692827_thoughts_on_data_science_in_business_and_industry/links/5e7955674585158bd501b45a/thoughts-on-data-science-in-business-and-industry.pdf)
- Taleb, I., Serhani, M. A., & Dssouli, R. (2018). Big data quality: A survey. In *2018 IEEE International Congress on Big Data - IEEE BigData Congress 2018: Part of the 2018 IEEE World Congress on Services: 2–7 July 2018, San Francisco, California, USA: Proceedings* (pp. 166–173). Piscataway, NJ: IEEE. <https://doi.org/10.1109/bigdatacongress.2018.00029>
- Van Eck, N. J., & Waltman, L. (2010). *VOSViewer: Visualizing scientific landscapes* [Computer software]. Retrieved from <https://www.vosviewer.com>
- Vollmer, M. (2018). *How to make it simple to explain AI, ML, DL and Data Science?* | LinkedIn. Retrieved from <https://www.linkedin.com/pulse/how-make-simple-explain-ai-ml-dl-data-science-dr-marcell-vollmer/>
- Weih, C., & Ickstadt, K. (2018). Data Science: The impact of statistics. *International Journal of Data Science and Analytics*, 6(3), 189–194. <https://doi.org/10.1007/s41060-018-0102-5>
- Worden, K., Haywood, J., & Bullough, W. A. (2003). *Smart technologies*. River Edge, NJ: World Scientific. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=235756>
- Xiang, Z., & Fesenmaier, D. R. (2017). *Analytics in Smart Tourism Design*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-44263-1>

# Industry Insights from Data Scientists: Q&A Session

Theory-based scientific research should be transparent and in close contact with society so as to ensure that current problems and their relevance are thoroughly understood, and attention is paid to these issues. For this reason, I decided to interview data scientists with extensive experience in the tourism field. Their responses should sharpen our view of the current state of data science in the industry and provide us with the business perspective needed to serve as a framework for academic research as well as allow us to ask timely and relevant questions. In order to obtain such a holistic picture, I asked five data scientists, all of whom work in different areas of the tourism industry (namely, a destination, an online travel agency, a review platform, the hospitality sector, and an airline), some questions regarding their work, experiences, and insights regarding the relationship between data science and tourism. Their qualitative statements intend to outline the true significance of data science in tourism, to filter out its challenges, and to capture a glimpse into the future.

## Interview 1

The first interview was conducted with **Holger Sicking**, head of the research department at the **Austrian National Tourist Office**. He studied business administration in Freiburg, Wisconsin, and Vienna and specialized in market research and statistics. Before working for the Austrian National Tourist Office, he worked as Head of Statistics for a Vienna-based market research agency and as a consultant for price modeling in different industries, and he was also a lecturer for market research at FH Wien, WU Wien, and MCI Innsbruck.

**Q:** *Holger, you told me in a conversation the other day that the Austrian National Tourist Office will become a “data-driven company.” What drives a DMO toward this, and what effects will this have on the organization?*

**A:** As we as a company are more like a creative agency and not, for example, a telecom company, I think we won't take this concept of a data-driven company to the limits; but yes, we are heading in this direction. We are currently developing a data strategy for our company including topics like data architecture, data management, and data factory. The overall goal is to base our operative and strategic decisions on data. We believe that using more data and better models (e.g., forecasting) has the power to make us more efficient and more agile and can bring better value to the tourism sector. We are not starting from scratch. For example, in the field of marketing performance management, we have already built quite a comprehensive system over the last two or three years. We are able to monitor the success of all our worldwide digital communication activities through one central dashboard by connecting more than twenty different data sources. In my opinion, the long-term effects of the implementation of our data strategy will be a cultural one especially. It changes the culture of work if you have absolute transparency with data. As an employee, you might also need new skills when working with data.

**Q:** *Within your research department, you have employed a data scientist for several months who is technically a physicist (by training). How does physics fit/work together with different tourism aspects, and what can you learn from having a physicist in your department?*

**A:** This is a very interesting question. The fact is that I was never actively looking for a certain university background other than the qualification and job experience as a data scientist. Hence, the applications are always quite diverse. Our data scientist has a PhD in physics and is a specialist in complex systems and chaos theory. Tourism is a complex system; there is not only one single product like in other business areas. The holiday product consists of a complex set of tangible and intangible parts. Therefore, he uses concepts like network theory to better understand this complexity. I constantly learn from our data scientist's new perspectives on working with data or, more generally, speaking about problem solving. He has a much broader perspective on problem solving than others and always tries to understand complex connections, which I believe has to do with his background in physics.

**Q:** *Austria is a relatively small country, and our tourism industry consists almost exclusively of non- and medium-sized businesses, which are mostly family-run. Furthermore, tourism has a long tradition in our country and is internationally known for having high qualities and being very professional. Why is the Austrian National Tourist Office, with its small structured market, trying to be a pioneer in the field of data science, and what do you believe are the biggest challenges resulting from these old, grown structures?*

**A:** We try to pioneer in the field of data and data science because, at present, we do not see anybody else who would do it with the whole Austrian tourism sector in mind. I guess this has to do with the situation you described, and this is also the

biggest barrier I see. The sector in Austria is very atomistic with a long history of family-run small and medium-sized businesses. They are doing a fantastic and professional job but are mostly stuck in their daily operations. It is not always very clear why it might make sense to invest money into data and algorithms, as you might not see direct returns from investments for your business. Therefore, we have the chance to showcase the value and future potential of data and data science. To give an example: we are currently doing a prototype project with the idea of using real-time mobile phone data at tourist hotspots in four regions in Austria. Tourism managers in these regions can monitor in real time the visitor volume and, based on this data, develop solutions for the management of visitor flows. Managers from the prototype tourism regions play an active part in the project; they make sense of the data, and we have joint workshops every three weeks to learn from each other. We initiated and financed the prototype project because we see much potential in this kind of data for the mobility and capacity management areas in tourism regions in the upcoming years.

## Interview 2

Thank you, Holger, for taking the time to answer these questions and for sharing your insightful thoughts on data science in Austria. Much appreciated! Up next is **Dr. Liliya Lavitas**. Liliya works as a senior data scientist at **Tripadvisor** and is responsible for creating hotel lists that best match travelers' needs. Since Liliya also has a strong academic background, some questions will focus more on this aspect.

***Q:** Liliya, you earned your PhD in statistics at Boston University. Do you believe that typical data science approaches like machine learning will become part of the standard repertoire for academic researchers in the same way as statistics currently is?*

**A:** These days, machine learning research is still primarily concentrated in the computer science field as implementation of a machine learning model requires technical skills and qualifications that traditionally are not considered necessary for statisticians. This being said, I firmly believe that advanced machine learning will be playing a larger role in applied research in statistics as deep knowledge and understanding of probability and estimation theory will allow the development of new generations of machine learning models.

***Q:** Auto ML, as discussed in Chapter 6 of this book, is becoming more and more popular. Quite a variety of tools is available on the market, and developers are trying to make them available to everybody, even those without programming knowledge. Where do you think these developments will lead in the future?*

**A:** Auto ML is indeed very promising and can be shown to be useful in some cases. Unfortunately, based on the tools that I've had a chance to explore, I still think that the use of auto ML is somewhat limited at the time. My opinion is based on the modeling framework that involves three stages: data generation, data exploration, and the actual model building/tuning. Let's consider how auto ML can be used for each of these stages. First of all, any model is only as good as the underlying data. I'm not entirely sure how auto ML can effectively solve a problem relating to training data generation; thus, data still needs to be generated by a subject matter expert. In my personal experience, proper data generation that contains enough signal to build a model is still the most time-consuming step of model building. Even more, data generation frequently requires multiple iterations to ensure the presence of all the signals that we are hoping to train the model on. I think that the use of auto ML is limited for this step. For the second stage, data exploration, auto ML tools are quite useful. Building insights for the generated data allows for the most automation, in my experience. Powerful and flexible tools for extracting insights from the data and for data visualization are indeed extremely useful in a data scientist's/ML engineer's work. Proceeding to model building itself, auto ML can be useful for benchmarking and comparing model accuracy across various model types. This being said, the set of model types that data scientists/ML engineers might consider is frequently limited by factors not related to the data itself. For example, implementation constraints can easily dictate what model types can be used. This leaves a data scientist/ML engineer with a pretty narrow choice of models for consideration. Finding the optimal model that meets both accuracy and implementation constraints is less complicated. One can argue that model tuning can be much easier with auto ML, and I think that this is a very solid argument for auto ML. Indeed, an auto ML tool is expected to tune models quite well. This being said, similar optimization results can be achieved by tuning the model "by hand" in a comparable timeframe.

**Q:** *In your opinion, do you think that at least the basics of data science approaches like NLP or machine learning should be incorporated into the curriculum for tourism studies in the future? How could/should we (better) prepare our students?*

**A:** I think that basic introductory interdisciplinary courses are extremely valuable for all industries that are currently benefiting and will be benefiting in the future from machine learning and data science. The main challenge I see is in building a course that explains complex mathematical concepts of machine learning to students of non-technical specializations. In my opinion, use cases and examples from applied machine learning in the tourism industry can be leveraged. One can consider first introducing examples of how machine learning and data science are being used in tourism now and then moving toward explaining the mechanisms of these applications. Finally, covering what other problems can and cannot be solved using machine learning and data science might be useful for students in tourism.



### Interview 3

Thank you very much for these incredibly interesting remarks on machine learning and data science, Liliya! My next questions are addressed to **Mike O'Connor**, a senior data scientist at **booking.com**. Mike has much experience in the tourism sector as he previously worked for Visit Baltimore and the Ritz-Carlton Hotel Company as well. Therefore, I would like to take this opportunity to ask him about his views on tourism as an industry sector in combination with data science.

**Q:** *Mike, would you consider the tourism sector to be a special field for data science? If so, how does it differ from other industries?*

**A:** There is no hard line separating hospitality analytics from other industries, but there are aspects in this business that are very different from other industries. In most cases, there is no physical product that a person takes home with them and puts on a shelf. The thing we sell is often intangible and ephemeral, and our customers' satisfaction is reflected not by some directly observable metric but by things tied to emotion. So much of hospitality is about ensuring that you sleep well after checking in, how much fun you had while seeing an attraction, or how friendly and helpful the flight attendants were. Our customers trust us to deliver an experience that lives up to their expectations, and they rely on us to build a relationship with them that lasts long after any money is exchanged.

For these reasons, data science in hospitality is focused on finding ways to measure the quality of those experiences and the depth of that trust, and it can be a very challenging thing to measure. In the DMO world, that might mean measuring the "value" of tourism in a way that is meaningful for lawmakers and local businesses. At a hotel, it could mean developing check-in and check-out processes that lead to better guest satisfaction and loyalty. At Booking, we combine a/b testing with our knowledge of guest preferences to pair users with travel products that make them happy and keep them coming back.

Of course, these are not challenges unique to the hospitality industry, and many others seek to answer the same questions. Music streaming, for example, strives to deliver a tailored experience that caters to the tastes and preferences of the listener. That relationship is cultivated by understanding what people enjoy and giving them more of it. But unlike with music, where I might listen to thousands of songs, people only travel a few times a year. The signals are faint and the stakes are high, especially when money and sleep are involved, and while many industries have only just begun embracing the "experience" as an opportunity to innovate, hospitality has been thinking about it since the very beginning.

**Q:** *You know the tourism industry very well from the hotel, destination, and OTA perspective. Looking at each individual sector, which ones do you believe need most improving or have the most catching up to do?*

**A:** None of the areas I've worked in have it totally figured out. I think they can all learn from each other, although cooperation among them can sometimes be harder than it seems. Hotels and OTAs have historically viewed each other with suspicion, given the nature of their relationship. This friction is unlikely to go away, but I think more cooperation is always possible and beneficial. In the DMO world, friction often comes in the form of local government or business owners who are unsure what return they get from their investment. Data has and will continue to play a very important role in solving these problems, but cooperation must come first.

When thinking about opportunities in the OTA sphere, I believe a tighter integration with accommodation providers is important because we control very little of the experience once a guest has checked in. This means thinking about products and services from both perspectives and building alliances that are more than purely transactional. In data science, it means putting ourselves in the shoes of the revenue manager or the concierge and solving problems for those audiences using the wealth of data we have available to us.

For accommodation suppliers, I can only speak from the perspective of a large chain. At those brands, experience has always dominated the conversation, and the brand identity is often built upon it. This has been a very valuable asset for hotel owners and given them a lot of flexibility to tailor the traveler experience precisely how they see fit. I do believe, however, that hotels may not be utilizing the “data-driven” approach to its fullest extent. Most big hotel chains and countless independent hoteliers have been operating since long before the Internet and so they may not be capitalizing on the data they have available to them. This means investing in digital platforms and the tools necessary to analyze them. Hotels should be building a fully integrated view of each guest's experience across every touchpoint, from the reservation process to the stay to the review after checkout.

Of all the areas I have experience, I think the DMOs are the ones with the biggest challenge ahead of them. Since the very beginning, they have been tasked with proving whether the budgets and membership dues are worth it to their constituency. To succeed, they must get a better grasp of the “impact” of destination marketing. It's monumentally difficult to get concrete numbers on the “value” of the hospitality industry in a city. Most DMOs point at jobs, small businesses, or the tax revenues collected by hotels as success metrics, but this is an incredibly complex attribution problem, and the data is sparse. Even if you solve it, the task of explaining that impact to lawmakers is equally challenging because DMOs are inextricably linked to the political machinery of the destinations they represent. A data scientist in this situation must be technically talented but also a very good communicator. Making impact easier to see and understand (especially for legislators and local business) will always be of paramount importance.

**Q:** *Chapter 3 of this book deals with the need for interdisciplinary studies/approaches. In your opinion, where will future data scientists for the tourism industry come from, which competencies are they expected to have, and how important will interdisciplinary teams be in the future?*

**A:** In my opinion, the ideal core competencies of a data scientist are less about specific technical knowledge and more about problem-solving skills in a business context. Data science itself is a rather broad discipline, and surveying a sample of people in my department illustrates just how diverse the backgrounds and skill sets are. I work with astrophysicists, computer scientists, biochemists, economists, business school graduates, college dropouts, and experts in countless subjects. The specific skills these people have aren't often what makes them good at their jobs, though; instead, they are those who see and understand an abstract problem and map it to a quantitative solution. There are many ways to solve most problems but what matters most is knowing how and when to solve them with whatever tools are appropriate and available. That is why a diversity of skills and experience is absolutely critical for any successful data science organization, hospitality, or otherwise.

When I interview people for data science positions, I don't tend to use esoteric coding tests or math problems to measure ability because it's not a good reflection of my typical workload. My day involves product managers asking me how to best measure customer loyalty, whether I can predict if a guest will cancel, or how to map the best hotels to go for wine tourism. So when hiring data scientists, I look for more than just technical knowledge; I want to know how well a candidate can solve the problem in the *right context* and how they *communicate* their ideas.

I've been very fortunate to see tourism from several angles, and that experience has helped me tremendously. With this in mind, I believe that successful data scientists in hospitality will be ones who understand the nature of the industry itself. This requires exposure to all aspects of travel, not just the narrow scope of a product, project, or individual company. Cross-pollination of ideas is where real innovation happens. But I also believe that data science in any field requires a mindset of experimentation and continuous learning. Tools and techniques are constantly evolving and improving, and the distance between raw data and actionable insight is shrinking quickly. This is why I value and rely on the perspectives of my peers from other disciplines. Data scientists are able to answer more complex questions than ever before, so the challenge of the future is using broad industry knowledge to choose which questions are worth answering and what *questions* to ask next.

## Interview 4

Great talking to you, Mike, thank you for your detailed responses! Data analysis is also becoming increasingly important in the hotel industry, providing a basis for decision-making in both operational and strategic aspects. I am therefore delighted to interview **Alex González Caules** next. Alex is a data scientist at **Meliá Hotels International**, a brand that operates over 390 hotels in more than 40 countries, and has a scientific background with a degree in economics and a master's in big data analysis.

**Q:** *The hospitality industry is a very fragmented market; whether it be leisure hotels, city hotels, individual hotels, or hotel chains, they all depend on data as a basis for decision-making. What questions does the chain hotel industry typically try to answer using data science methods?*

**A:** There are many areas where data science and machine learning models can help uncover hidden insights. Often the questions that can be answered with these methods take the form of “how can we do X best, given our knowledge of the environment?”. Some examples would be the optimization of digital marketing and advertising strategies (e.g., how can we reach our customers more efficiently without spending unnecessary advertising budget?) or segmentation methods (e.g., how can we best separate customers or products into groups so that we can find the product–customer combination that will potentially create the most value?). These methods help to homogenize the various fragments of the industry since it allows decision-makers to view each segment under a different lens. It is also very common to try to predict the probability of an event happening, such as a reservation or cancellation, or to apply optimization techniques to revenue management, even though this area is still quite underdeveloped, and there is a lot of room for research in dynamic pricing systems.

However, there is still a large space to be filled with data science models as their applications are nearly limitless due to the fact that such methods can be used in many different areas and for several various needs. Tons of data are generated everyday that might be recorded into databases but ultimately not put to profitable use. This is another aspect of the data science role: to understand the business environment and be able to reply with potential solutions should any stakeholder have questions regarding how to address a business need. Finding new questions to be answered with data is almost as crucial as being able to answer them.

**Q:** *What types of analyses do you think hold potential in the future but are currently barely being used in the hospitality industry?*

**A:** NLP is, to my knowledge, very underused in the industry. While it is one of the main techniques used in other sectors, such as finance, the hospitality and tourism industry has not yet found a golden use case for this technique as it is mostly limited to review and sentiment analysis. There are some chains testing automated assistants, concierges, and receptionists, but none have reached maturity yet. AI-powered voice and text assistants are starting to get traction as they can be implemented in some businesses, such as hotels, with similar ease as implementing one in a home. Deep learning techniques are also uncommonly found in the hospitality and travel industry although some more complex models often borrow concepts from this, such as deeply layered neural networks (convolutional or recurrent) for personalization models.

**Q:** *Alex, at what scale does it make sense for the hotel industry to get involved with data science, and what recommendations can you give smaller companies?*

**A:** Data science, machine (and deep) learning, artificial intelligence. . . This is what differentiates cutting-edge and innovative companies from others. An organization, whether big or small, will generate large amounts of data. With an IT team, any company can gather these data and exploit it beyond reports and dashboards with the use of data science methods. This is possible even with staff that are not scientifically trained in statistics or modeling since many third-party vendors now offer statistical software licenses that are easy to use for any individual with working knowledge of computers. Thanks to this democratization of machine learning, it is increasingly easy to prototype, triage, and discard models in various different applications. As a result, the marginal cost of developing every new model continues to decrease, making it a profitable endeavor for any company, regardless of its size.

## Interview 5

Excellent, Alex! Truly appreciate your remarks and thoughts on the hotel industry. Last but not least, I had a chance to sit down and talk to **Jeroen Mulder**. Jeroen studied mathematics (PhD) at Leiden University and has worked for more than twenty years as an operations research specialist and data scientist for **Air France–KLM**, both in Paris and in Amsterdam. Currently, he is working for the technology innovation office inside the Air France KLM branch, looking into emerging technologies and how these can contribute to a more sustainable data- and AI-driven company.

***Q:** Jeroen, to initiate this discussion, could you please talk a bit about typical analysis scenarios within the airline industry and where data science methods are applied? What kind of questions do you ask, and which methods are mainly applied to prompt an answer?*

**A:** The airline industry has three types of business models, which, next to passenger travel, are transporting cargo and maintaining aircraft. Each of those domains has its own problem characteristics and its own responsibilities in solving problems. However, each domain is closely linked to the others, resulting in, next to the domain-specific types of problems, transversal and interdependent problems. Hereafter, I will give a short sketch of each of these domains and will explain their dependencies.

As an airline, we accommodate passenger's travel. In order to be profitable, we want to sell our passengers the right tickets at the right moment and for the right price. In order to do so, we need to be able to accurately forecast the number of expected passengers for the different destinations. Since we provide different types of services at different prices, we need to differentiate these expected numbers by the type of passenger, for example, leisure passengers vs. business passengers. As a result, we need to be able to forecast the expected number of different types of passengers together with the expected behavior of these passenger types. Such

different behaviors can be seen in the moment tickets are booked: leisure passengers will most likely book their flights long before their departure, once they booked their holiday accommodations. Business passengers usually book their flights closer to departure.

This different behavior also has to be taken into account when ensuring that there are still enough available seats for business passengers closer to departure; the right balance needs to be found between accepting all leisure passengers long before departure with no seats left for business passengers and accepting only smaller number of leisure passengers long before departure in order to have enough seats left for business passengers. In the first case, you will have full flights but miss the opportunity to serve your business demand; in the latter case, you might have emptier flights because your leisure demand booked their flights with competitors. In both cases, as an airline, you will have missed revenue, but, even worse, you missed the opportunity to provide the best service for your passengers.

In all of the above cases, we need to analyze the historical behavior of our passengers based on past data and enrich these with data from actual bookings that have already been made. To learn as much as possible from the actual data, the techniques used are traditional forecasting methods in combination with ML models. These range from linear regression techniques through exponential smoothing techniques and ARIMA and ANOVA modeling, where the focus lies on the explainability of the results of such techniques, to ML techniques like random forests and others, where one wants to discover as much behavior as possible from the actual bookings so far. It is good to point out the different types of data used when modeling: actual bookings are concerned with passengers that might still cancel or not show up for their flight, and historical bookings involve “final data,” that is, data that will not change anymore. These different types of data require different modeling techniques because of the range of uncertainty thereof.

Finding the right balance between the different types of passengers with respect to the seats that are to be kept “available” is a classical optimization problem that needs much computational power to be solved. Often more heuristic approaches are used to ease computational efforts where more modern ML techniques are used. Here, it is good to point out that, as an airline, you want to accommodate all types of passengers, meaning that you will need to forecast and optimize for types of passengers that you may have never seen before. As a result, optimizing the future available seats for that many types of passengers becomes a huge performance challenge; again, the right balance needs to be found between this computational challenge (and its costs) and the risk that you did not anticipate for certain types of passengers, leading to a lower service for your passengers.

Next to accommodating passengers, travel airlines also transport cargo. In essence, this is the same problem or challenge as with passengers, only with different characteristics. For cargo, other dimensions are relevant, like weight and volume. The types of cargo are also more diverse; they range from live animals to pharmaceuticals to dangerous goods, and many others. The types of problems are the same and require similar techniques and methods but with other dimensions, which from time to time can make the problem more difficult to solve. There is one big difference

between passengers and cargo that should be mentioned. Passengers either show up for their flight or they do not. Cargo is different in that the actual weight or volume of the cargo is most likely to be different from what was booked, meaning that the cargo measurements/dimensions can be more or less than originally planned. This means that typical models used for forecasting passengers' behavior can't be translated one-to-one for models that accurately forecast the "behavior" of cargo. The other big difference between transporting passengers vs. cargo is their so-called booking window; passengers can book their flight one year in advance, while cargo has a booking window of approximately two weeks, rendering the modeling of the underlying behavior a different game.

There is also a dependency between passenger travel and transporting cargo because the aircrafts used are mostly the same. Some airlines use dedicated freighters to transport their cargo, but many airlines also transport their cargo in the belly of a passenger aircraft. Such dependencies between the passenger world and the cargo world add additional complexity to the problems that need to be solved.

Maintaining the aircraft and improving the aircraft's "health" requires a different organization compared to transporting passengers and cargo. "Customers" can also refer to other airlines for which we maintain their aircrafts or its components, with a strong focus, of course, on preventing component failures and aircraft incidents. Over the past few years, the emphasis has shifted toward preventing this type of maintenance, resulting in so-called predictive maintenance. For predictive maintenance, a combination of domain knowledge from aircraft engineers and data scientists' expertise on predicting relevant indicators that will signal component deterioration is required. The main challenge there is that, due to current strict safety regulations on aircrafts, there are not that many incidents, making it hard to detect patterns within the data and to link them to the components' unusual behavior that led to these incidents in the first place. Here, data scientists rely heavily on the domain knowledge of aircraft engineers, where their knowledge of the circumstances that can lead to the deterioration of components has to be translated into relevant indicators. The holy grail is to find ML techniques that will detect from the data what the best predictive models should be, so, more or less, relying on the expertise of aircraft engineers.

One aspect here needs to be mentioned and stressed: aircraft maintenance is a strictly regulated business, where there are clear regulations for how components should be treated and are certified by independent regulators. Applying ML techniques without the involvement of humans, like aircraft engineers, poses a new challenge: how to certify such techniques according to existing regulations.

The domain of maintenance also impacts the other two domains of transporting passengers and cargo. The more one flies, the more its aircrafts are utilized, which, in turn, impacts the state of the aircraft and its components. In other words, better service for your passengers and cargo feeders can conflict with the maintenance service needed. On the other hand, improving your predictive maintenance skills will improve the availability of your aircraft fleet. Again, the right balance between the costs and benefits of predictive maintenance and the impact on your fleet needs to be found. Incorporating predictive maintenance in your fleet scheduling in order to

create more robust schedules with healthier and more available aircrafts requires a cross-over between traditional optimization and state-of-the-art data science.

**Q:** *What recommendations can you give to companies wanting to set up a data science department?*

**A:** Airlines have been here for a long time already, so the need for forecasting techniques in order to predict the right number of passengers for your flight or the right weight and volume of your cargo has also been present for many years. Most airlines have been focusing on so-called operations research expertise to do so. Profiles with such expertise are quite close to what we, nowadays, call data scientists. Depending on the size of a company, it is sometimes better to have a centralized organization of such experts or a decentralized organization, when a central department becomes too big. What has not changed over the years is that all organizations need to be agile and adapt to new developments, as we have seen from the rise of big data and AI. Working together closely with researchers from universities and senior profiles from “niche” providers of data science and AI is the “best practice.” For smaller companies, some advice would be to have a central organization specialized in data science but with close links to other expertise like data analytics and data engineering. The experience so far has been that working in a multidisciplinary team with these different roles really helps organizations to move forward. In case companies are too small to afford such a central organization, it is worthwhile to work together with external providers of such expertise.

**Q:** *Are there any specific processing steps when carrying out data science projects that you consider to be particularly prone to errors?*

**A:** Any successful data science project starts with good data. For any company, this will be the main challenge. The first “easy” step is modeling your data science model on a small and “fixed” data set. At that moment, a model needs to be industrialized and to run in production; the troubles begin when new and unseen data come in, causing inconsistent data, or even erroneous data, and leading to models with unexpected and unwanted behavior. Most of the time, we define upfront test cases that can be used to detect inconsistencies and errors in the data. However, we often forget to define test cases that can be used to detect unexpected and unwanted behavior. Especially with respect to ML techniques, we often fail to prevent unwanted behavior like biasness. Therefore, the most important processing steps that need to be secured are the industrialization of data feeds and setting up tests to detect unwanted behavior in our models.

**Q:** *Lastly, I would be interested in how ethical issues can be confronted and ensured in data science projects. What are your thoughts on this?*

**A:** There are many ethical aspects to be considered. Each airline has a huge responsibility when it comes to securing the safety protocols for flying and



maintaining an aircraft. This comes with a cost, whereas the objective of an airline is to be profitable, leading to conflicting interests. One means to become more profitable is to rely more and more on AI and data science in our daily operations, which, if not done properly, could increase the risk of missing unexpected and unwanted behavior. One way to ensure safety protocols is to certify machine learning models in a similar way as we train and certify our pilots and crew. It requires independent protocols and standards, for example, as agreed with the SAE International (previously known as the Society of Automotive Engineers, a US-based, globally active professional association and standards-developing organization for engineering professionals in various industries).

Next to certification, an awareness of such ethical issues also needs to be raised. For example, it is relatively simple to add fairness constraints to your ML models so as to prevent overfitting, which can cause biased behavior; yet, if one is not aware of such fairness constraints, the resulting models will still turn out biased and unfair. Currently, we are thinking of implementing a code of conduct for our data scientists and other data-related profiles, which should provide a minimal set of “best practices” to help create fairer and less-biased models. Our aim is to do so not only as an airline, but together with other airlines and data science providers. Also an ethical committee, who assesses all major projects dealing with data and looks at the different ethical aspects thereof, could be another option.

## **Concluding Remarks**

I would like to thank all the experts for their comments and the time they took to answer my questions. I am certain that the interaction between employees with domain knowledge/expertise in tourism and data scientists with their mathematical-technical understanding will be crucial for the successful implementation of future data science projects in tourism. This requires a mutual understanding of prerequisites, needs, and requirements. With this interview, I hope to have provided a small glimpse into the “behind the scenes” that can serve as inspiration for practitioners as well as researchers in the field of tourism.

**Part I**  
**Theoretical Fundamentals**

# AI and Big Data in Tourism



## Definitions, Areas, and Approaches

Luisa Mich

### 1 Introduction

Classical definitions of Artificial Intelligence (AI) date back to the 1950s (McCarthy 2007), all including the concept that AI can enable computers to accomplish tasks and activities that are regarded as intelligent, i.e. requiring human-level intelligence. Given the difficulties in defining human intelligence, a more operational definition can refer to the abilities and capabilities AI aims to automatize:

- communication, in all forms and including all types of media (text, picture, audio, video);
- perception, which has attracted a considerable amount of attention with recent developments in new input and output devices (e.g., sensors, the Internet of Things (IoT), and cyber-physical systems);
- knowledge, making it storable, retrievable, and processable for a variety of applications;
- planning, as a backup for decision-making and responding (e.g., in robotics or autonomous driving);
- reasoning, simulating human thinking and learning processes.

As all these capabilities are interconnected, so are the corresponding subfields of AI, characterized by different approaches and applications. A list of the traditional core areas of AI research includes problem-solving, intelligent agents, natural language processing (NLP), speech recognition, computer vision, robotics, knowledge representation, and machine learning.

Despite its ups and downs, including the so-called AI winters (Lim, 2018), AI progress is evident in all the above-mentioned areas. The use of AI technologies and systems is so widespread that discussions about their applications, performances,

---

L. Mich (✉)

Department of Industrial Engineering, University of Trento, Trento, Italy

e-mail: [luisa.mich@unitn.it](mailto:luisa.mich@unitn.it)

and impact are quotidian. However, the question as to whether computers are indeed intelligent remains to withstand a positive answer. For example, even in NLP, an AI area with impressive achievements, computers do not really “understand” the content of a text. In fact, they cannot support fully fledged interactions in natural language yet; existing conversational chatbots use large datasets (big data) of exchanged sentences but do not actually “know” the content of such sentences, nor could they explain their answers. The same happens in automatic translation systems.

Without diminishing the significant progress that has been made in AI, the most recent and realistic view is summed up in the term “Intelligence Augmentation” or IA, considering AI technologies as augmenting human intelligence (Jordan, 2019). According to IA, humans have to be kept in the loop of automatization processes, adopting a semi-automatic approach whenever technologies are unable to fully replace human intelligence. Such a vision has been embraced also by big companies, as, for example, IBM, whose AI system, a.k.a. Watson, won Jeopardy!, a general culture quiz, in 2011. The victory was sealed thanks to a combination of question-answering, knowledge representation, and strategy-planning modules (Ferrucci et al., 2010). Such an achievement, followed by some successful game-winning AI systems—(AlphaGo<sup>1</sup>), Libratus (poker) (Hsu, 2017), AlphaZero (chess) (Silver et al., 2017)—have contributed to revitalizing AI industrial investments and research funding. According to a McKinsey online survey, half of the responding organizations had adopted AI for at least one function in 2019 (Balakrishnan et al., 2020). Other interesting data can be found using the Global Vibrancy Tool, which supports cross-country comparisons for up to 26 countries across 22 indicators.<sup>2</sup>

Moreover, the 2021 AI index report from the Human-Centered AI institute at Stanford University reports a major growth in AI projects. Unsurprisingly, global corporate investments in AI have increased from US \$12,751 million in 2015 to nearly 68,000 in 2020. Another indicator of the success of AI involves the number of AI patents published worldwide, which in the past two decades has steadily increased from 21,806 in 2000 to more than 101,876 in 2019. The industry’s role in AI development is also confirmed by an increase in AI Ph.D. graduates going into industry: in North America, a 65% growth in 2019, up from 44.4% in 2010, could be observed. In addition, governments are also defining their AI strategies. For instance, Canada published its strategy in 2017, followed by more than 30 other countries and regions as of December 2020 (Zhang et al., 2021). Investments in AI and big data for the tourism sector can benefit considerably from such strategies.

---

<sup>1</sup> <https://deepmind.com/research/case-studies/alphago-the-story-so-far>

<sup>2</sup> <https://aiindex.stanford.edu/vibrancy>

## 2 AI, Machine Learning, and Data Science

Many of the recent advances in AI are due to Machine Learning (ML) so that the two terms are often used as if they were synonymous. As a matter of fact, ML is a subfield of AI, and its goal is to realize computational models that make computers learn “what to do” instead of having to tell them “how to do” it. An essential step in the history of computers is connected to the concept of stored programs, in which instructions on how to solve a problem are registered in a memory and executed on the input data in order to produce the output data. Thus, programs were the result of the translation of algorithms using an (artificial) programming language. As such, programs implement algorithms to automatize well definable activities or procedures. ML, however, is based on a different approach, with the basic idea being that of making the computers learn from examples. Consequently, instead of explaining step-by-step how to solve a problem, the ML models learn which is the correct output based on given input or an optimization procedure. A traditional example in image recognition tasks is classifying cat and dog pictures; for the traditional programming approach, this is a challenging goal. We could start with a (graphical) model of cats and dogs, a textual description, etc., but achieving good results with these types of methods is out of reach. On the contrary, in ML, a learning-by-example approach would show a (large) number of cat and dog pictures, giving the correct classification for each of them, and, in this way, training the “computer” to recognize which animal is depicted in unseen pictures.

While the concept of learning by example is by no means new—as a teaching method it was referred to by Pierce (1931–1958)—its realization and exploitation as a new computational model has only been made possible thanks to the progress in computer hardware technologies. The reason for this being that ML systems require large datasets and large memories as well as processing resources (Mich, 2020a). For example, the adoption of accelerator chips designed for ML reduced the training time of image classifier systems based on ImageNet<sup>3</sup> from 6.2 min in 2018 to 47 s in 2020.<sup>4</sup> Besides learning by example, numerous other ML models or algorithms exist and can be classified into three main categories: supervised, unsupervised, and reinforcement based. Some ML systems combine models of different kinds or include new ideas, like that of attention (Bollinger, 2021). Identifying which one is most suitable for a given task is typically not a trivial problem; its solution does require expertise and knowledge both in ML and the subject matter at hand (for more details about ML, see chapter “Machine Learning in Tourism: A Brief Overview”).

One major incentive for ML research and its applications has involved big data. The variety of input sources, storage devices and systems, the Internet and the Web, faster networking services, and sensors have all contributed to the incessant creation of large sets or reservoirs of data (see, for example, statistics given in Petrov (2021)). Differing in format, type, quality, and structure, this data, or big data, has fueled the

---

<sup>3</sup><https://www.image-net.org>

<sup>4</sup><https://mlcommons.org/en/training-normal-07>

need to mine and interpret them. Yet, traditional techniques for data processing, database models, and information system functionalities face many challenges when confronting datasets that are many orders of magnitude larger than usual.

AI and ML not only take advantage of big data, but also offer new solutions via a bidirectional relationship. Through this, progress in AI and ML has contributed to a new research and application area known as Data Science (DS) (Murtagh & Devlin, 2018). DS aims to interpret available data, uncovering hidden relationships and patterns. As an interdisciplinary field, DS projects require a skilled approach in statistics, mathematics, computer science, machine learning, data visualization, communication, and presentation skills. Furthermore, all these capabilities must be used in different domains, making domain expertise a mandatory aspect for any sound and successful data exploitation and data management activity (Chauhan & Sood, 2021). Another concept that helps to understand the critical role of domain expertise is that of data vs. information vs. knowledge. To be used as input for ML algorithms, data must first be pre-processed (e.g., classified or labeled), usually with the help of domain information. Subsequently, the output of the ML algorithms can be utilized to support knowledge-based processes, such as in business intelligence systems, customer relationships management systems, or in many other applications (Sarker, 2021). Classical principles of information systems suggest, just like other advancements in digital technologies and applications, that ML and DS solutions allow companies and organizations to improve their business processes, to transform their business models, and to support new ones. The big tech companies, or the so-called FAANGs (Facebook, Apple, Amazon, Netflix, and Google), are the clearest examples of how data can be used to create business value. Many of their recent successes are due to services that use AI and big data solutions.

### 3 AI for Big Data

In an effort to characterize big data, and to answer the question “how *big* are big data?” a mainstream definition introduces three parameters: volume, variety, and velocity. All of these parameters are relevant to understanding why AI models and tools can be useful for leveraging big data.

- *Volume* A high volume of data can be collected from a variety of sources and tools, including business transactions, smart (IoT) devices, industrial equipment, mobile devices, radio-frequency identification (RFID) readers, wireless sensor networks, social networks, etc. The size and number of available datasets have increased beyond expectations, but storing it with new and cheaper storage options and platforms has also become much easier now as it was in the past.
- *Velocity* Data are produced and used at high speed. Cyber-physical systems, embedding hardware and software components in complex contexts (e.g., in driverless cars, robotics), and many other widely used applications (e.g., chatbots

or personal assistants) produce fast data streams and frequently require near real-time processing.

- *Variety* Data comes in many types of formats. Only a small fraction of them consist of structured data and can be managed with classical databases. The vast majority of data are unstructured in the form of texts, pictures, videos, sensors and ticker data, etc. A given input, e.g., a webpage, usually includes a combination of different formats. In addition, there are no shared standards for many of the new data types (e.g., for home automation systems), contributing to the complexity of the processing problem.

Other models for big data include a wider variety of parameters. Among the most frequently added is veracity followed by variability (also called volatility):

- *Veracity* Data have to be quality checked. Since the very first computer programs, the relevance of the quality of data has been highlighted through the term, “Garbage In Garbage Out” (GIGO). In fact, problems in the input data always impact the output, and all the parameters characterizing big data make the quality of data a difficult and multifaceted problem. The fact that there are many data points, many sources, many changes, many formats, etc., challenges any application using big data, but especially ML computational models that use such data to self-train.
- *Variability* Data flows change often and vary greatly. This characteristic is frequent in tourism related data; for example, tourist flows change, sometimes dramatically, depending on events, weather conditions, daily or seasonal supply, and demand. Such changes have to be addressed promptly to deal with peak data loads as well as to extract valuable knowledge to guide companies’ subsequent decisions.

Big data are also described using the term “Value” as a key dimension to underline the fact that data have to be interpreted by taking their nature, i.e., statistical, hypothetical, etc., into account; but more importantly, Value means that “having access to big data is not good unless we can turn it into value” (Marr, 2015). According to the second definition, Value is not an intrinsic characteristic of big data and, thus, had to be modeled at a meta-level, not including it in the Vs set. All the “Vs” of big data involve challenges faced by companies. Big data can also be seen as a moving target, depending on the available computational resources and capabilities—a trend shared by any technological innovation: in other words, new technologies, and AI progress in particular, are continuously expanding the scope of big data applications.

The synergy of AI and big data can be described based on the five main steps of a general DS process model (even if not all DS projects require big data):

- Problem framing
- Data gathering
- Data cleaning
- Data processing
- Data exploitation

Like any other process, the five steps can be completed linearly, one after the other, or one of the life cycle models defined in software engineering can be applied (Sommerville, 2018). AI models and tools may be incorporated in all the steps.

### ***3.1 AI for Problem Framing***

This step is arguably the most critical one, as results strongly depend on the initial questions and stated goals. Big data by itself has no value unless you can extract meaningful results—value—from it. To this end, it is first necessary to ask a meaningful question, framing a business problem, or defining a company strategy (*from problem to AI-based solution*). On the other hand, the discipline of information systems teaches that, like any other technology, AI solutions can suggest new ways of using big data to add value to a company’s products or services (*from AI system to addressable problem or business strategy*). In this way, some standard questions could state the following: “How could <name\_of\_an\_AI\_tool\_or\_models> be useful for gathering data and supporting our strategies?” or “How can the company use new types of data for <name\_of\_an\_object\_or\_relationship>?” Lastly, AI techniques can also support creativity in order to explore new and innovative ideas to state the problems to be tackled using big data (Boden, 1998).

### ***3.2 AI for Data Gathering***

Data is, by definition, a representation of a fact or phenomenon; with the new generation of AI technologies, such representations are also obtainable in terms of big data. AI contributes to the “datafication” of the real world, i.e., to the process of objects, people, and processes being “transformed” into digital data (Southerton, 2020). For example, online interactions allow for the gathering of data that is useful to study behavior and social life. IoT and Industry 4.0 technologies create large datasets due to the automatization of exchanges between machines and processes. The process to download content from a website or from a social networking platform is also called data scraping (see chapter “Web Scraping”), and some of the available tools even apply NLP (Lane et al., 2019) or computer vision techniques (Brownlee, 2019). In some cases, big data are owned by different organizations or companies, which is the case for smart city projects. Legal property, but also privacy issues and security, have: there are three subjects here! be properly addressed in order to obtain and manage the needed data (Bartneck et al., 2021). In addition, more often than not, AI tools are also used to this end (e.g., <https://securiti.ai>), in a loop that, to be effective, has to take on humans—government, companies, people (Zhou et al., 2020). Vice versa, for governments or large enterprises, the question to ask should be “Who should own big data initiatives that affect the different



organizations?” Lastly, although not quite as widespread, AI techniques can also help in identifying legal or privacy requirements (Zeni et al., 2015) or look for relevant data (i.e., search engine capabilities using n-grams: <https://books.google.com/ngrams/info>).

### 3.3 *AI for Data Cleaning and Preparation*

The quality of data is a major concern in any automatic system and, therefore, even more so when it comes to big data. Data cleaning includes errors and duplicated elimination, a task requiring different methods for the different input formats and dealing with different standards and access rules. In some cases, it is necessary to filter, transform, or integrate available data coming from different sources and stored across many systems. The entire collection of a company’s data—structured and unstructured—is also named “data lake.” Usually stored in a cloud platform, data lakes are types of “warehouses” collecting copies of an organization’s data; and, as for *warehouses*, data lakes have to be adequately managed so as to be able to retrieve the datasets that are useful for the next steps. Typically, cleaned data must be prepared in order to be used as input for the analysis step; thus, this step is strictly intertwined with the following step and is also described as part of a data wrangling process (Lakshmanan et al., 2020). AI, and especially ML, can help in many of the tasks related to data cleaning and preparation, e.g., by using pattern recognition techniques to complete missing data (Ilyas, 2020).

### 3.4 *AI for Data Processing*

Data pre-processed in the previous step can be explored and analyzed using ML algorithms and other AI techniques, for example, via network analysis (see chapter “Network Analysis”) or agent-based algorithms (see chapter “Agent-Based Modeling”). In addition, depending on the type of datasets involved in the analysis and the related AI areas, there are many various approaches that can be adopted. First of all, data visualization techniques can be used to explore the data and look for outliers or regularities, also with support from traditional statistical models. The final goal is to define a strategy where valued information can be extracted from the data. For example, if the input includes large sets of textual product reviews to be scored automatically, a subset of them can be pre-processed using semantic networks, conceptual models (Harmelen et al., 2010), or other linguistic methods. Subsequently, ML algorithms can be applied to define scoring rules corresponding to identified linguistic patterns or conceptual relationships. If the dataset includes images, on the other hand, then other areas of AI can contribute. The same applies to IoT data streams, etc. Furthermore, AI can support analytical tasks of increasing complexity, from descriptive to predictive and prescriptive. The problem, however,

lies not only in the very choice of how the AI method or system can be applied, but also in the complexity thereof. For this reason, experts in data analysis, AI, and the application domain are all necessary, underlining the need for a multidisciplinary and collaborative team for big data projects. On the plus side, many libraries and platforms support handling big data and data science methods, including ML (Krensky et al., 2021).

### 3.5 *AI for Data Exploitation*

Output obtained through big data analyses can be used to take action or decisions (Sharda et al., 2021) and to define companies' strategies, e.g., identify new target markets, optimize production processes, reduce energy consumption, and forge ahead with preventive maintenance and all the traditional information systems (inventory management, production management, and supply chain, among others). There is seemingly no limit to the applications of AI-based solutions, and each sector and task could, in one way or another, be supported by intelligent systems using big data. In healthcare, AI improves diagnostic accuracy and efficiency by using millions of cell phone and sensor data to support evidence-based medicine or virus tracking, while in agriculture, AI enables growth monitoring through the use of satellite systems or drone data. Moreover, financial analytics support trading and data available through e-learning systems can be used to improve teaching activities and to tailor it to individual students. Among the most recent and widespread AI-based tools, recommendation systems (e.g., on Amazon), virtual assistants (e.g., Siri, applying NLP and speech recognition systems), fraud detection systems (e.g., those used for credit cards), and traffic visualization (e.g., on Google maps, using different sources of data from smartphones or GPS) have made their marks. Progress in NLP makes it possible to automatically generate news, support web sentiment analysis, and automatically generate news to support web sentiment analysis, reputation monitoring, and fake news detection. Augmented and virtual reality also offer new ways to communicate, e.g., with 3D yourself or with wearable devices that respond to your thoughts.<sup>5</sup> Additional examples include computer vision that supports facial recognition, robotics that are applied to driverless automotive systems, and so on.

However, having such a variety of uses highlights the need for operational standards, and, thus, to this end, the ABOUT ML<sup>6</sup> initiative has introduced transparency guidelines. Other problems and risks are those related to explainability and accountability issues as many ML algorithms fail to justify their output and, in turn, to establish the responsibility of the involved subjects (Garigliano & Mich, 2019). Lastly, another primary concern regarding the use of AI is algorithmic bias. All sorts

---

<sup>5</sup><https://news.mit.edu/2018/computer-system-transcribes-words-users-speak-silently-0404>

<sup>6</sup><https://www.partnershiponai.org/about-ml-2021>

of human prejudices and opportunistic behaviors can be found in AI systems and ML applications (O’Neil, 2016), challenging researchers, companies, and governments to form an ethical vision of big data (European Economic and Social Committee, 2017) and AI (<https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence>).

## 4 AI and Big Data in Tourism

Tourism is a complex and multidimensional sector, and all its features contribute to making it an ideal space for the application of AI systems exploiting big datasets. First, due to the large number of people, processes, and activities involved, tourism data are often big data. Furthermore, there seems to be no limit when it comes to the types of data sources and AI technologies that could be rendered useful in investigating and improving both the supply and demand of tourism, including marketing strategies and overtourism management. These technologies could even facilitate recovery from crisis periods and much more. Consequently, despite the unfeasibility of a review of all significant experiences and projects in this field, some guidelines can be given:

- AI and big data projects can take advantage of “*external*” data sources, i.e., not only those traditionally considered in tourism applications (Li et al., 2018). For example, in addition to user-generated content (UGC), weather forecasts and climate change data could be used to address the problem of seasonal tourism supply. In addition, the two aspects of negotiation experience and data harmonization knowledge are as critical for successful initiatives as they are for smart city projects.
- Innovative tourism strategies can only be designed with new visions and ideas. Creativity techniques can subsequently be applied by considering insights gained from analyzing big datasets. Alternatively, new ways to visualize them could be used; among the most recent principles of circular economics, sustainable development and environmental economics can give inspiration to government and destination managers, local tourism boards, and tourism stakeholders in general.
- Regarding the many data gathering technologies, those which identify people’s location, or equally, any kind of phenomenon are very relevant to tourism geography. New (action-)tracking technologies and techniques, including GPS, mobile, sensors as well as drone data, can be used to manage overtourism issues, e.g., for route optimization, traffic management, customized guided tours, etc. (Mich, 2020b).
- AI and big data could also help reduce the fragmentation of tourism (mobile) apps as, often times, many apps are too specialized and not personalized enough. One of the reasons for this being that each is based on a single dataset and is not designed to satisfy actual stakeholders’ requirements.

- AI progress must be constantly monitored to improve any of the quasi-classic applications of big data further. For example, systems analyzing UGC for customer recommendations or web reputation monitoring can take advantage of some of the advancements in NLP while also exploiting techniques for image classification, face recognition, video analysis, and voice identification,<sup>7</sup> in addition to the more recently developed emotion recognition.
- Given the central role of communication in tourism, on top of content analysis, it is possible to apply the so-called generative everything AI systems to support content generation. Texts, audio, and images created by these systems are not easily recognizable as being automatically produced (and require new solutions to copyright). In regard to content quality, AI can also use datasets to carry out fact checking and deepfake detection.
- Finally, new and innovative AI and big data applications can also take advantage of traditional (analogical) documents of any kind. There are large amounts of documents that, when digitalized, could be integrated with existing datasets, adding precious knowledge to tourism destinations, e.g., regarding their cultural and historical heritage. Crowdsourcing initiatives, in this sense, could also help to build up communities, which are key to tourism.

To conclude, it is essential to highlight that the hype around obtaining unexpected results through mining big datasets via AI is prominent in academic and professional sources. However, such results require considerable investment and specialized knowledge as well as caution regarding social and political implications (see, e.g., Subirana (2020) for privacy risks of voice analysis) in a systemic and networked sector such as tourism (Baggio et al., 2010).

## Further Readings

- Alpaydin, E. (2021). *Machine learning* (Revised and Updated Edition). MIT Press Essential Knowledge Series.
- Baggio, J. A., & Baggio, R. (2020). *Modelling and simulations for tourism and hospitality: An introduction*. Channel View Publications.
- Balakrishnan, T., Chui, M., Hall, B., & Henke, N. (2020). The state of AI in 2020. *McKinsey*. <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020>
- Iafrate, F., & ISTE (Londyn), & John Wiley & Sons. (2018). *Artificial intelligence and big data: The birth of a new intelligence*. ISTE.
- Ilyas, I. F., & Chu, X. (2019). *Data cleaning*. ACM Books <https://doi.org/10.1145/3310205>

---

<sup>7</sup><https://www.gartner.com/smarterwithgartner/7-digital-disruptions-you-might-not-see-coming-in-the-next-5-years>

- Laudon, K. C., & Laudon, J. P. (2020). *Managing information systems: Managing the digital firm* (16th ed.) Pearson.
- O’Neil, C., & Schutt, R. (2013). *Doing data science: Straight talk from the frontline*. O’Reilly Media.
- Russell, S. J., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
- Sigala, M., Rahimi, R., & Thelwall, M. (Eds.). (2019). *Big data and innovation in tourism, travel, and hospitality: Managerial approaches, techniques, and applications* (1st ed.). Springer Publishing.
- Voulgaris, Z., & Bulut, Y. E. (2018). *AI for data science: Artificial intelligence frameworks and functionality for deep learning, optimization, and beyond*. Basking Ridge.
- Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., Lyons, T., Manyika, J., Niebles, J. C., Sellitto, M., Shoham, Y., Clark, J., & Perrault, R. (2021). *The AI index 2021 annual report*. AI Index Steering Committee Human-Centered AI Institute, Stanford University <https://hai.stanford.edu/research/ai-index-2021>

## Other Sources

- AI Trends—The Business and Technology of Enterprise AI: <https://www.aitrends.com>
- Becoming Human—Exploring Artificial Intelligence & What it Means to be Human: <https://becominghuman.ai>
- Big data Centres of Excellence: <https://www.big-data-value.eu/skills/bigdata-centres-of-excellence>
- IBM—Artificial Intelligence: <https://www.research.ibm.com/artificial-intelligence>
- Machine Learning Department Research—Carnegie Mellon University: <https://www.ml.cmu.edu/research>
- Stanford Artificial Intelligence Lab—SAIL: <http://ai.stanford.edu>

## References

- Baggio, R., Scott, N., & Cooper, C. (2010). Improving tourism destination governance: A complex science approach. *Tourism Review*, 65(4), 51–60.
- Bartneck, C., Lütge, C., Wagner, A., & Welsh, S. (2021). Privacy issues of AI. In *An introduction to ethics in robotics and AI* (SpringerBriefs in ethics). Springer. [https://doi.org/10.1007/978-3-030-51110-4\\_8](https://doi.org/10.1007/978-3-030-51110-4_8)
- Boden, M. A. (1998). Creativity and artificial intelligence. *Artificial Intelligence*, 103(1–2), 347–356. [https://doi.org/10.1016/S0004-3702\(98\)00055-1](https://doi.org/10.1016/S0004-3702(98)00055-1)
- Bollinger, T. (2021). Why AlphaFold is not like AlphaGo. *Academia Letters*, Article 728. <https://doi.org/10.20935/AL728>

- Brownlee, J. (2019). *Deep learning for computer vision. Image classification, object detection, and face recognition in python*. <https://machinelearningmastery.com/deep-learning-for-computer-vision>
- Chauhan, P., & Sood, M. (2021). Big data: Present and future. *Computer*, 54, 59–65. <https://doi.org/10.1109/MC.2021.3057442>
- European Economic and Social Committee. (2017). *The ethics of Big Data: Balancing economic benefits and ethical questions of Big Data in the EU policy context*. <https://www.eesc.europa.eu/sites/default/files/resources/docs/qe-04-17-306-en-n.pdf>
- Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefter, N., & Welty, C. (2010). The AI behind Watson—The technical article. *AI Magazine*. <http://www.aaai.org/Magazine/Watson/watson.php>
- Garigliano, R., & Mich, L. (2019). Looking inside the black box: Core semantics towards accountability of artificial intelligence. In M. ter Beek, A. Fantechi, & L. Semini (Eds.), *From software engineering to formal methods and tools, and back* (LNCS) (Vol. 11865). Springer. [https://doi.org/10.1007/978-3-030-30985-5\\_16](https://doi.org/10.1007/978-3-030-30985-5_16)
- Hsu, J. (2017, January). *Meet the new AI challenging human poker pros*. IEEE Spectrum. <https://spectrum.ieee.org/automaton/artificial-intelligence/machine-learning/meet-the-new-ai-challenging-human-poker-pros>
- Ilyas, I. F. (2020, February). AI should not leave structured data behind! How AI can solve the notorious data cleaning and prep problems. *Towards Data Science*. <https://towardsdatascience.com/ai-should-not-leave-structured-data-behind-33474f9cd07a>
- Jordan, M. I. (2019). Artificial intelligence—The revolution hasn't happened yet Harvard. *Data Science Review*, 1(1). <https://doi.org/10.1162/99608f92.f06c6e61>
- Krensky, P., Idoine, C., Brethenoux, E., den Hamer, P., Choudhary, F., Jaffri, A., & Vashisth, S. (2021, March). *Gartner magic quadrant for data science and machine learning platforms*. Gartner. <https://www.gartner.com/en/documents/3998753>
- Lakshmanan, V., Robinson, S., & Munn, M. (2020). *Machine learning design patterns: Solutions to common challenges in data preparation, model building, and MLOps*. O'Reilly.
- Lane, H., Howard, C., & Hapke, H. M. (2019). *Natural language processing in action*. Manning.
- Li, J., Xu, L., Tang, L., Wang, S., & Li, L. (2018). Big data in tourism research: A literature review. *Tourism Management*, 68, 301–323. <https://doi.org/10.1016/j.tourman.2018.03.009>
- Lim, M. (2018). History of AI winters. *Actuaries Digital*. <https://www.actuaries.digital/2018/09/05/history-of-ai-winters>
- Marr, B. (2015). *Big data: Using smart big data, analytics and metrics to make better decisions and improve performance*. Wiley.
- McCarthy, J. (2007, November). *What is artificial intelligence?* Stanford. <http://jmc.stanford.edu/articles/whatisai/whatisai.pdf>
- Mich, L. (2020a). Artificial intelligence and machine learning. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-Tourism* (pp. 1–21). Springer. [https://doi.org/10.1007/978-3-030-05324-6\\_25-1](https://doi.org/10.1007/978-3-030-05324-6_25-1)
- Mich, L. (2020b). Systems and technologies for fluxes management. In M. Franch & R. Peretta (Eds.), *Tourism, fragilities and emergencies* (pp. 107–134). McGraw Hill.
- Murtagh, F., & Devlin, K. (2018). The development of data science: Implications for education, employment, research, and the data revolution for sustainable development. *Big Data and Cognitive Computing*, 2(2), 14. <https://doi.org/10.3390/bdcc2020014>
- O'Neil, C. (2016). Weapons of math destruction: How big data increases inequality and threatens democracy. *New York Time*.
- Petrov, C. (2021, July). 25+ impressive big data statistics for 2021. *Techjury*. <https://techjury.net/blog/big-data-statistics/#gref>
- Pierce, C. S. (1931–1958). In: C. Hartshorne & P. Weiss (Eds.), *Collected papers of Charles Sanders Peirce* (Vols. 1–8). Harvard University Press.
- Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *Sn Computer Science*, 2, 160. <https://doi.org/10.1007/s42979-021-00592-x>

- Sharda, R., Delen, D., & Turban, E. (2021). *Analytics, data science, & artificial intelligence: Systems for decision support*. Pearson.
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, *550*, 354–359.
- Sommerville, I. (2018). *Software engineering* (10th Global Ed.). Pearson.
- Southerton, C. (2020). Datafication. In L. Schintler & C. McNeely (Eds.), *Encyclopedia of big data*. Springer. [https://doi.org/10.1007/978-3-319-32001-4\\_332-1](https://doi.org/10.1007/978-3-319-32001-4_332-1)
- Subirana, B. (2020). Call for a wake standard for artificial intelligence. *Communication of ACM*, *63*(7), 32–35. <https://doi.org/10.1145/3402193>
- Van Harmelen, F., Lifschitz, V., & Porter, B. (2010). *Handbook of knowledge representation*. Amsterdam Elsevier.
- Zeni, N., Kiyavitskaya, N., Mich, L., Cordy, J. R., & Mylopoulos, J. (2015). GaiusT: Supporting the extraction of rights and obligations for regulatory compliance. *Requirements Engineering*, *20*(1), 1–22.
- Zhou, T., Shen, J., He, D., Vijayakumar, D., & Kumar, N. (2020). Human-in-the-loop-aided privacy-preserving scheme for smart healthcare. *IEEE Transactions on Emerging Topics in Computational Intelligence*. <https://doi.org/10.1109/TETCI.2020.2993841>

# Epistemological Challenges



## Is the Future Theory-Driven or Data-Driven?

Roman Egger and Joanne Yu

### Learning Objectives

- Understand the difference between a knowledge- and data-driven approach
- Appreciate epistemological challenges of data-driven approaches
- Comprehend the significance of theory even in data-driven approaches
- Acknowledge epistemological challenges and pitfalls in order to successfully execute data science projects

## 1 Introduction

When applying data science techniques for scientific discovery, what is particularly important, but often ignored, is the nature of theoretical knowledge (Rizk & Elragal, 2020). Epistemology, or sometimes also known as the theory of knowledge, is a branch of philosophy concerned with knowledge, logic, and reasoning (Swan, 2015). Before diving deeper, however, it is equally important to get a glimpse of what theories are and what they are for. In essence, theories consist of accumulated knowledge constructed in a systematic way so as to provide guidance for practice,

---

R. Egger (✉)

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

e-mail: [roman.egger@fh-salzburg.ac.at](mailto:roman.egger@fh-salzburg.ac.at)

J. Yu

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

Department of Tourism and Service Management, Modul University Vienna, Vienna, Austria

e-mail: [joanne.yu@modul.ac.at](mailto:joanne.yu@modul.ac.at)



offer a (new) lens/perspective on interpreting a set of phenomena, develop testable propositions for empirical research (Gregor, 2006), and, eventually, challenge state-of-the-art knowledge (Simsek et al., 2019).

However, the existing ambiguity regarding whether tourism should be considered an independent discipline, in addition to its fragmented knowledge construction (Darbellay & Stock, 2012; Tribe, 2010), pushes tourism even further away from a uniform explanation or a solid “theory of tourism” (Tribe, 2010). Instead, knowledge and methodological approaches from a wide range of disciplines have come together to try and explain the multifaceted phenomenon of tourism (Oviedo-García, 2016). As described in chapter “Data Science and Interdisciplinarity”, understanding tourism as a cross-sectional domain requires the following: deep knowledge of the various methods used in its sub-disciplines, research collaboration, and the development and testing of suitable instruments to answer new types of questions. Akin to the propositions of theories, they serve their purposes of analysis, explanation, prediction, design, and action (Gregor, 2006). In particular, owing to the increasing proliferation of online research in tourism, addressing questions in a new way is of high relevance (Song & Zhu, 2016). Hereby, a discussion of the various methods that are most likely to be applied in the upcoming years will provide further information about the professional and correct use of the Internet as a data source and survey tool.

The act of collecting and analysing the vast amount of information and data available online, though not necessarily big data, has been viewed as data-driven science (Kitchin, 2014), which, consequently, has decreased the value of theories in general. Yet, some scholars hold opposite viewpoints by arguing that knowledge construction is fragile in a theory-free context (Harford, 2014). For instance, when looking at Google Flu Trends, the algorithm only provides one with the correlation between influenza cases rather than trying to solve the more urgent and realistic concern of what causes what (Harford, 2014). At an epistemological level, data-driven research has led to a paradigm shift in knowledge discovery (Balazka & Rodighiero, 2020) and offers new ways for us to make sense of the world (Kitchin, 2014). As such, understanding epistemological issues in data analytics is critical in that they fundamentally influence research design and theory construction (Elragal & Klischewski, 2017).

Distinct from traditional deductive approaches for testing theory, advanced analytics, often in empirical research, lays the groundwork for entirely new epistemological perspectives, where insights are “born from the data” (Kitchin, 2014). Nevertheless, there are ongoing debates concerning whether the rise of data science is really a new paradigm or simply just an innovative way of adopting tools for scientific enquiry (Rizk & Elragal, 2020). Some scholars claim that the spirit of inductive reasoning remains in data science research by preserving experiential knowledge (Mazanec, 2020), whereas others argue that data science research should rather involve an integration of inductive and deductive approaches (McAbee et al., 2017). These criticisms emerge from some of the epistemological challenges that are inherent to research with data science techniques. For instance, although the ubiquitous nature of user-generated content (UGC) eases the process of identifying data

sources, the development of research questions might be tailored according to “where” the data exists (Elragal & Klischewski, 2017). Moreover, other scholars argue that the data-rich environment opposes the concept of sampling procedures (Balazka & Rodighiero, 2020). Likewise, the data-driven paradigm shifts the focus from causality to inductive reasoning and correlation (Mayer-Schönberger & Cukier, 2013).

Whilst acknowledging that epistemological reflections should go hand in hand with knowledge generation, only about 0.5% of publications in the data science discipline have explicitly mentioned/discussed epistemological issues (Balazka & Rodighiero, 2020). The sophistication of data science itself is what makes it exciting, yet it leads to researchers shying away from (or simply ignoring) the challenges posed by data analytics. To adequately address the (potential) pitfalls that derive from data science, interdisciplinary collaboration (see chapter “Data Science and Interdisciplinarity”) should be highly valued and sought after (Symons & Alvarado, 2016).

As information and communication technologies continuously foster the development of the tourism industry (Xiang, 2018), the purpose of this chapter is to address the epistemological issues in the area of tourism research where data science techniques are used as methods in the hopes of providing an overview of potential challenges tourism scholars and practitioners may face. The remainder of this chapter proceeds as follows: Sect. 2 outlines the paradigm shift in knowledge discovery, whilst Sect. 3 discusses the epistemological challenges that merit attention for each step involved in the data science process. Finally, Sect. 4 highlights the chapter’s conclusions and summarises the knowledge-driven and data-driven approach for epistemological reflection.

## 2 Epistemological Evolution

In a nutshell, the epistemology of data science research has evolved from the Petabyte Age, where data spoke for itself (Anderson, 2008), to the fourth paradigm (Hey et al., 2009), to data-driven science (Kitchin, 2014), and, finally, to lightweight theory-driven approaches (Elragal & Klischewski, 2017).

In the early 2000s, the rise of the digital economy aroused the interest of some large companies, such as Google, Facebook, and Amazon, to act as leaders in disentangling our lives based on a massive amount of data (Harford, 2014), thereby opening the doors to the Petabyte Age (Anderson, 2008). For instance, Google, which heavily relies on applied mathematics and statistics, believes that the accumulation of knowledge, without tracing back to their root causes, is sufficient to make solid assumptions and conclusions (Anderson, 2008). In this way, due to the agnostic nature of dimensionality (Levallois et al., 2013), the only thing scientists need to do is await statistical algorithms that search for trends and patterns within the datasets (Prensky, 2009). Characterised by “the end of theory” (Anderson, 2008), petabytes of data support the notion that correlation is superior to causal knowledge

(McAbee et al., 2017). In practice, scholars explain this reasoning as causality being too difficult or impossible to address; therefore, statisticians would rather opt for correlations (Gregor, 2006). As such, typical scientific approaches such as hypothesising, modelling, and testing data were rendered obsolete (Prensky, 2009). From an epistemological stance, although Anderson (2008) claimed that the large volumes of data are enough to speak for themselves, interpretation always goes back to human understanding (Hannigan et al., 2019). Such beliefs may also be prone to the bigness bias (Lukosius & Hyman, 2019) as it does indeed hold true that “n” could never be equivalent to “all” (Harford, 2014).

Since the enormous growth of data posed serious challenges in various disciplines (Levallois et al., 2013), scientists subsequently proposed the postulate of a “fourth paradigm” (Gray & Szalay, 2007). The fourth paradigm, which gives birth to data-intensive science at a later stage (Kitchin, 2014), transcended the previous experimental, theoretical, and computational approaches through the use of more advanced database technologies and analytics tools (Levallois et al., 2013). Further elaborated by Hey et al. (2009), whilst the fourth paradigm of science is detached from theories, the paradigm goes beyond empiricism and simulation as it constitutes more complex methods of data analysis (e.g. grid computing, visualisation, and other novel analytical methods). Nevertheless, there are ongoing debates concerning its epistemology (Rizk & Elragal, 2020). Some scholars relate the fourth paradigm to the rebirth of empiricism (Kitchin, 2014), whilst others criticise that it simply adds new methodological approaches for scientific inquiry based on existing paradigms (Müller et al., 2016). More recently, social scientists have argued that the high volume of data does not reflect idiographic knowledge, leading to an incomplete representation of reality (Singleton & Arribas-Bel, 2021).

Despite the prominence of the fourth paradigm, most scholars to date support the idea that data is not inherently meaningful (Balazka & Rodighiero, 2020; Kitchin, 2014; Rizk & Elragal, 2020), claiming that even if analytical processing could be fully automated, data are neither theory (Gregor, 2006) nor can they speak for themselves (Redman, 2014). Interpretation always requires the data to be framed within specific business purpose(s) (Redman, 2014) and contextualised in particular domain(s) for researchers to give meaningful insights and to avoid generalisation bias (Gregor, 2006), confounding bias, and ecological fallacies (Rizk & Elragal, 2020), amongst many others. As data continues to revolutionise the humanities and social sciences, using pure induction for theory generation or deduction for theory confirmation (McAbee et al., 2017) is no longer adequate. In fact, theory does not necessarily come before data (McAbee et al., 2017). Kitchin (2014) developed data-driven science as a new paradigm for epistemology to further extract additional insights based on a combination of inductive, deductive, and abductive research principles. Abductive reasoning balances the two approaches and allows researchers to move back and forth throughout the data analysis process (Hauer & Bohon, 2020). That is, rather than solely relying on pre-defined conceptual or theoretical frameworks (Symons & Alvarado, 2016), data-driven science is novel in that hypotheses, insights, and relevant theories result from the data using inductive reasoning to guide the research design, which can then be tested deductively at a later stage (McAbee et al., 2017).

Lately, scholars have noted that, although Kitchin (2014) emphasised the necessity of taking theories into consideration, theories have been used mainly in the data analysis process rather than tackling the initial framing of research projects and overall directions (Elragal & Klischewski, 2017). Even though process-driven approaches have been commonly adopted in information systems research, they always have to be accompanied by consolidated theories for better prediction, and vice versa (Ali et al., 2019). In fact, the aforementioned arguments rationalise the notion that domain knowledge is the key to shaping research questions and reaching goals in data-driven science (Canali, 2016). Since “*humans cannot not hypothesise*” (Mazanec, 2020), even just out of curiosity in everyday life, researchers are always bound to implicit assumptions or expectations. Akin to the lightweight theory-driven approach proposed by Elragal and Klischewski (2017), it enables researchers to move beyond pure quantitative analytics by incorporating different domain expertise so as to answer the “how” of knowledge acquisition. With the use of theories to guide research design and analysis, the quality of predictive analytics can be improved and fine-tuned (Miah et al., 2017). As such, we argue the significance of interdisciplinary collaboration for building a priori knowledge of the data to ensure the robustness of the results and conclusions.

### **3 Epistemological Challenges: Data Science in Tourism Research**

Consequently, to overcome epistemological challenges (Canali, 2016) when applying data science in tourism research, attention should be paid to each individual step involved in the multiphasic process. As discussed in chapter “Interdisciplinarity and Data Science”, the key phases of data science projects include (1) topic formulation and its relevance for academia and industry, (2) data access and data collection, (3) data pre-processing, (4) feature engineering, (5) data analysis, (6) model evaluation and model tuning, and (7) interpretation of the results. To ensure that each step can be completed smoothly and successfully, this section discusses the potential pitfalls, highlighting their epistemological challenges, that may emerge whilst conducting a data science project. It additionally provides guidance for tourism scientists and practitioners to improve the governance and trustworthiness of their research when choosing to combine data science with tourism.

#### ***3.1 Topic Formulation and Relevance for Academia and Industry***

As in any research project, the data science process begins with formulating a topic, which should be supported by its relevance for science and industry. At this point,

the question of whether a completely theory-free and purely data-driven approach is at all possible and whether “the end of theory” (Anderson, 2008; Kitchin, 2014) will take over in the near future arises. The answer would be a relatively simple and straightforward “no”, if one assumes two premises: first, the analyst’s actions are not random, and, second, questions can come about from a fundamental curiosity that is based on rudimentary hypotheses (Mazanec, 2020).

Both the preface of this book and the introductory chapter mention the three levels of data science, with domain knowledge being one of them. Domain knowledge refers to the cumulative knowledge that one possesses and has built up over many years based on experiences. Kim and Pedersen (2010), therefore, assume that domain-specific knowledge also contains theoretical assumptions and that this knowledge also encompasses hypotheses in the sense of a “well-founded guess” (Klahr et al., 1993). If one assumes that the topic formulation and the problems and questions are target-oriented, at least a minimum amount of domain-specific knowledge must be present. Since this compiled experiential knowledge could be understood as theory, it continues to have meaning, at least if one assumes that its function is not to reinvent the wheel over and over again (Mazanec, 2020). According to this logical conclusion, a basis of theory seems to be inherent in every (meaningful) data-driven knowledge generation process.

Although the authors are not aware of any study that supports the following assumptions, it does seem to be the case that data science methods are highly welcomed in the review process. Time and again, certain methods become a trend; a few years ago, papers with structural equation modelling (SEM) were in, and, nowadays, machine learning (ML) approaches seem to be very popular. In principle, such styles are more than acceptable as long as one main common consensus exists: the method *must* be suitable for answering the research question or falsifying the hypotheses, i.e. the choice of method is downstream.

However, the availability of new datasets tempts to reverse this paradigm, and it often seems to be the case that data are collected in the first step without having a utilisation context in mind yet. Thus, one runs the risk of not aligning topic formulation with its relevance for society and the current state of research but, rather, with the data that is available. This phenomenon has become known as the “streetlight effect” (Elragal & Klischewski, 2017), meaning that researchers study phenomena involving a lot of available data (Rizk & Elragal, 2020) instead of focusing on the truly important and relevant problems at hand (Rai, 2016). In this sense, Rivera (2020) emphasises, “when it comes to Big Data Research, hospitality researchers must avoid falling for the streetlight effect. That is, looking for answers where the data is better and more accessible rather than where the truth is most likely to lie” (p. n.a.).

### ***3.2 Data and Its Access and Collection***

As such, since the streetlight effect also seems to be a temptation within tourism research, we have already found ourselves in the middle of the discussion on data,

along with its access and collection. If one looks at recent publications involving the analysis of big data, mostly the same data sources have been used. Popular ones, for example, include the analysis of online reviews based on data from TripAdvisor, the analysis of tweets, and studies based on Airbnb data. On that note, data from TripAdvisor is relatively easy to crawl, and tweets are trendy as they can be retrieved relatively easily via an API (Slota et al., 2020). As for Airbnb, data is provided, for example, by “Inside Airbnb”, an independent, non-commercial initiative that crawls Airbnb data and makes it available to download for further analysis (Egger et al., 2022a). It is often argued that social reality can be captured and described particularly well if the entire population were to be analysed. However, one must keep in mind that only a certain percentage of the population is online, with merely a section of it being active on individual channels and platforms, thus leaving traces to be collected and resulting in biased data and limited representativeness/generalisations (Hargittai, 2020; McFarland & McFarland, 2015).

The different types of data in tourism as well as the various possibilities of accessing and collecting data are discussed in more detail in chapter “Web scraping: Collecting and retrieving data from the Web”. Therefore, details on open data types, APIs, and web crawling will be omitted here. However, it is important to mention data monetarisation (Elragal & Klischewski, 2017) as a phenomenon. Companies have understood the value of their data and are earning good money with it. Again taking Twitter as an example, the social media platform offers its own Twitter API “Firehose” for business and academic research, giving Master students, PhD students, and faculty or research-focused employees access to global, real-time, and historical data. The free basic version allows 10 million tweets per month, and paid versions will be available from 2022 onwards. Selecting the right data can also become a challenge as there is simply too much data out there, and preparing and pre-processing it requires much time and many resources. It seems surprising, in the context of big data, to talk about *too* much data being a problem; but, in fact, the vast amount of data can, in some cases, also increase the noise in the data (Vargas, 2020), making it difficult to identify valuable signals (Torabi Asr & Taboada, 2019).

In practice, researchers will need to depend on the way data can be obtained. This may require technical skills that not everyone possesses and may, therefore, also be quite limiting. Elragal and Klischewski (2017) point out that sampling is particularly important in data collection, which raises epistemological challenges since goal-oriented data collection and acquisition requires appropriate theorisation. Nonetheless, Mayer-Schönberger and Cukier (2013) argue that with big data, data can be captured in its completeness and sampling; thus, concerns about the accuracy of the data become obsolete. As the entire population is examined, the problems of obtaining representative samples are eliminated (Mariani et al., 2018). Steadman (2013) even talks about the fact that, nowadays, everyone can use big data “regardless of how comfortable they are with that situation”. Hence, the problems of subjectivity raised by Max Weber more than a 100 years ago (the personal interests and values of a scientist lead to a specific understanding of objects; knowledge must be thought of as “knowledge from particular points of view”; the “criteria by which this segment is selected” are inseparable from the cultural framework through which

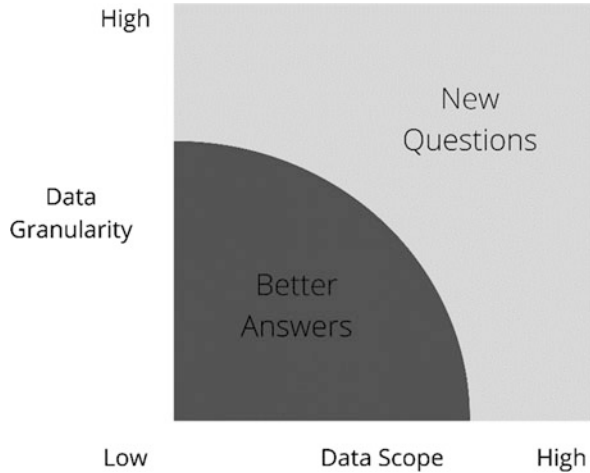
the final meaning is acquired) seem to be solved by big data (Balazka & Rodighiero, 2020). This epistemological way of thinking assumes that one can rely on data based on the motto “let the data speak for itself”, in that raw data are seemingly independent of human subjectivity (Barrowman, 2018), strive for consistency, and want to replicate themselves (DuBravac & Shapiro, 2015).

Certainly, this way of thinking has its charm, and certain advantages cannot be denied. For example, when data are collected in the context of classical empirical social research, they cannot be considered “raw” because many factors influence them. Are questions answered according to social desirability, i.e. are they biased in favour of the respondents? Is there always a common understanding between the researcher and the respondent regarding the research question? Were there incentives to participate in a survey? Moreover, if a clear method for what should be measured has yet to be established, then the question arises as to whether the operationalisation was correct. How was the data collected and under what conditions was it entered into a computer system? Was it stored correctly? Yet, even when big data has been collected, the extent to which this raw data represents the “truth” and is not subject to distortions, such as context, remains in question. Does user-generated content truly represent social reality, or are online reviews, tweets, and Instagram posts not also subject to social desirability and positive self-representation? Are GPS based trajectories subject to noise, especially when people move indoors and signals become inaccurate or are interrupted? Are data provided by companies via APIs actually raw or perhaps preselected as they are gatekeepers to the data (Schrock & Shaffer, 2017)? Ultimately, the selection and collection of data are always subject to design-specific decisions (Seaver, 2017).

Barrowman (2018) outlines this naive way of thinking, i.e. relying blindly on data, as follows: “Raw data, uncorrupted by theory or ideology, will lead us to the truth; complex problems will be solved simply by throwing enough data at them. No experts will be required, apart from those needed to produce the data and herald their findings; no theory, values, or preferences will be relevant; nor will it be necessary to scrutinize any assumptions” (p. 134). Balazka and Rodighiero (2020) also criticise such an objectivist rhetoric of big data, i.e. viewing data as neutral, omnicomprehensive, and theory-free, and argue that data collection is neither objective nor neutral, exhaustive data collection remains mostly theoretical, and interpretation of data is subjective and theoretically informed.

Big data is defined primarily by its volume, but its granularity is also of particular importance. George, Osinga, Lavie, and Scott (2016) understand granularity as the possibility of directly measuring the defining characteristics of a construct (Fig. 1). Through data science techniques, more accurate and better results to test existing theories can be obtained thanks to more precise estimates of effect sizes and their confidence intervals. For example, by aggregating data, sensors can measure in real-time not only when how many people are in a city centre but also how many of them are tourists and how the context (weather, events, time units, etc.) affects their behaviour. This provides a better understanding of how phenomena such as overtourism emerge and how prognostic methods can support visitor management and guidance strategies to avoid overcrowding (Such-Devesa et al., 2021).

**Fig. 1** Granularity and scope of data—implications for theory. Source: adapted from George et al. (2016)



Nevertheless, George et al. (2016) point out that newly obtained measures and constructs need to be linked to existing theories in order to avoid a division into literature with “small” and “large” data.

As long as data volume and granularity continue to increase, new questions can be explored. Whilst new constructs can be introduced (again, the importance of domain knowledge in the context of feature engineering should be highlighted here), existing constructs can be operationalised in a fresh and innovative way (George et al., 2016). When different perspectives in regard to a specific phenomenon are linked to the corresponding data, a better and more coherent picture can form. Accordingly, new data sources can give rise to new questions that can be derived from existing theories and eventually further developed and improved upon.

To sum up this paragraph in one, the data that are ultimately selected and analysed are highly dependent on a researcher’s epistemological mindset, the expectations of discovering patterns and associations in the data (Mazanec, 2020), a researcher’s personal preferences and technical capabilities, the streetlight effect, the impact of data monetisation (Elragal & Klischewski, 2017), and current publication trends.

### 3.3 Data Pre-processing

The fact that pre-processing data is one of the most time-consuming steps in the entire data science process also indicates that simply feeding raw data into algorithms is not effective. During pre-processing, an attempt is made to ensure the quality of the data; it is, therefore, a matter of having specific expectations concerning the data, both in terms of quantity and, above all, in terms of quality. Are the data consistent for the domain, are necessary attributes missing, are there any outliers that need removing, which data should be included, which should be



omitted, and should missing values be replaced, and, if so, how? These are just some of the many questions that must be asked in order to harmonise the data and prepare it for further processing. As described in detail in chapter “Natural Language Processing (NLP): An Introduction”, text data, in particular, is sometimes subjected to complex pre-processing steps or vectorised for further processing (see chapter “Text Representations & Word Embeddings”). At this stage, a large number of decisions need to be made, which, again, presupposes that the research purpose of the data is known/has been well thought out in advance based on theoretical and domain-specific knowledge.

### ***3.4 Feature Engineering***

Feature engineering, which is discussed in detail in chapter “Feature Engineering”, is another intermediate step in data science. Datasets often contain a very large number of variables, and it is not uncommon to have hundreds of them. Especially in exploratory studies, where the goal is theory generation, a few variables that make the largest explanatory contribution to a model have to be identified (George et al., 2016). Often times, the available variables are not even the most informative data, and only the combination and aggregation of data, generated metadata, or statistical characteristics about the data represent the most significant features. Thus, data science, again, requires extensive domain knowledge for which a theory-driven approach can be helpful. For feature selection, however, numerous metrics are available in order to evaluate the individual features so that domain expertise can be coupled with statistical evaluations.

### ***3.5 Data Analysis***

Data analysis, the heart of the data science process, also involves numerous epistemological challenges and pitfalls. First, a shift from the typical search for causality to correlations can be observed (Balazka & Rodighiero, 2020; Mayer-Schönberger & Cukier, 2013). The analysis of large datasets leads to a loss of statistical significance as even variables with the smallest effects will be significant. Meanwhile, spurious correlations are likely to appear when a large number of features are considered. In general, traditional statistical inference is said to be inappropriate for complete population studies (Alexander, 2015) since they are designed to work with selected samples. Instead, Allenby, Bradlow, George, Liechty, and McCulloch (2014) point out that Bayesian statistical approaches can be a proper solution for analysing a whole population since, here, the data are fixed and the parameters are random (George et al., 2016).

Apart from that, the fundamental question of what methods to use and which algorithms to apply for knowledge discovery remains. Furthermore, at this point, the

discussion on how algorithms are created and whether they are able to quantify social reality accordingly and accurately continues to be deliberated. Algorithms are designed by humans to minimise human bias (Egger et al., 2022b), but, at the same time, they are embedded in a social framework. For example, capitalist forces can influence algorithm fairness and bias (see chapter “Data Science and Ethical Issues”). According to Balazka and Rodighiero (2020), algorithms are understood as “constantly changing, theoryladen, and naturally selective human artifacts produced within a business environment” (p. 4).

As not all software solutions provide hyperparameters for fine-tuning, and the algorithms themselves end up changing from time to time (Leonelli, 2018), computational reproducibility embodies yet another challenge. Moreover, it can also be the case that an algorithm had originally been trained on data that are no longer valid today. As a result, a model trained on old data is applied to new data without taking any necessary changes into account. For example, customer behaviour and economic data naturally change over time, which can lead to incorrect predictions if models trained on outdated data are still used for current situations (Egger et al., 2022b).

To appropriately address epistemological challenges in the analysis phase, Elragal and Klischewski (2017) suggest that constructs used in analysis should be grounded in accepted theoretical concepts, interdisciplinary data science teams should be assembled, and a theoretical framework should be provided for the selection of techniques or models.

### ***3.6 Model Evaluation and Model Tuning***

Data analysis, model evaluation, and model tuning usually tend to be iterative processes. Depending on the algorithm, common evaluation scores such as ROC/AUC or F1 (for classification) or mean average error and mean squared error (for regression) are used, requiring cross validation or holdout methods as sampling techniques to select the best model (see chapter “Model Evaluation”). Hyperparameter tuning (see chapter “Hyperparameter Tuning”) tries to change the tuning knobs of an algorithm in such a way so that the algorithm also achieves the best possible result. However, due to insufficient available information on existing machine learning systems, choosing a set of optimal hyperparameters for an algorithm has never been an easy task (Zhou et al., 2017), especially for researchers lacking a strong background in data science (Kraska et al., 2013).

This phase not only requires a comprehensive understanding of the available hyperparameters, but the researcher’s interaction with the data is equally important. Being able to obtain a value that determines whether results are good or bad is just the first step; what is more significant for researchers is to have a full understanding of *why* such results happen in accordance to the settings of certain hyperparameters (Elragal & Klischewski, 2017). In case there is a gap in knowledge regarding individual hyperparameters and what their changes can cause, default settings are often applied. In this way, models are evaluated and assessed without even having to

turn the adjustment screws. As such, it is crucial to bear in mind what one truly wishes/aims to know, if the results meet these expectations, or if further tuning is required. In the increasingly popular AutoML approaches, this tuning is done without human intervention (Egger, 2022a), thereby assuming that a purely data-based analysis can represent reality better than a theory-based analysis (Kempeneer, 2021).

### 3.7 *Interpretation of Results*

Interpreting data and results quickly becomes a challenge due to the fact that this step is always subject to human bias (Symons & Alvarado, 2016), and, in many cases, may even be a nearly impossible task to fulfil (Rudin, 2019). For instance, when it comes to predictions in particular, ML algorithms such as random forests, support vector machines, or neural networks are superior to traditional inferential statistical methods because they can work with high-dimensional data (Arora & Gabrani, 2018), account for interaction effects between variables (Breiman, 2001), and capture nonlinear relationships between variables (Müller et al., 2016). However, this accuracy comes at the expense of interpretability; although these models produce better results, it is extremely problematic to reconstruct and understand the algorithms' decisions. As a consequence, one has to trust "black box" algorithms, which also makes it difficult to justify the results (Ribeiro et al., 2016). Hence, the majority of scientists opt for interpretability, supposing that the results are high quality and are in line with their theoretical/domain knowledge (Müller et al., 2016; Sharma et al., 2014). To address the concerns of algorithmic decision-making, currently, intensive research efforts are underway and interesting methods to make black box models more transparent with "eXplainable Artificial Intelligence" (XAI) are available (see chapter "Interpretability of Machine Learning Models"). Since algorithms are ultimately intended to help humans make decisions, explanations must be prepared in such a way that they are comprehensible to the relevant target group (managers, authorities, etc.) (Barredo-Arrieta et al., 2020; Rudin, 2019).

Especially in the context of text analysis, which was originally qualitatively oriented, the interpretation of results can quickly end up looking unclear, incorrect, and unprecise. Interpreting topic modelling results, for instance, is a good example thereof. Although Latent Dirichlet allocation (LDA) is technically a quantitative, probabilistic topic modelling method, topic modelling in general is considered a set of qualitative methods since it statistically preprocesses text data despite the interpretation of the results always being qualitative (Egger, 2022b). This means that the results include topic clusters, characterised by keywords, which have to be evaluated and interpreted by humans. Depending on the method, there are different statistical evaluation metrics, such as topic coherence, that can be applied; nonetheless, a manual topic description always remains necessary and is, therefore, rendered a subjective process (Hannigan et al., 2019).

Another significant issue brought to light by Elragal and Klischewski (2017) is the interpretation of “quick and dirty” patterns. As data scientists are able to run an analysis easily and in a time sufficient manner, spending more time on interpreting the data and looking at the results in detail (with regard to the pre-defined research objectives) appears to be rather unattractive if there are other interesting and obvious patterns that emerged from the data. In particular, recent advancements in data visualisation tools, such as Voyant Tools, provide additional quick access to tourism researchers (Nukhu & Singh, 2020), regardless of their background, for them to draw conclusions based on the observations of raw data patterns.

Overall, since interpretation in data science projects is always accompanied by human fallibility to some extent (Balazka & Rodighiero, 2020), researchers are advised to develop and apply theoretical frameworks when making meaningful conclusions, which, once again, echoes the prominence of theories in data-driven approaches.

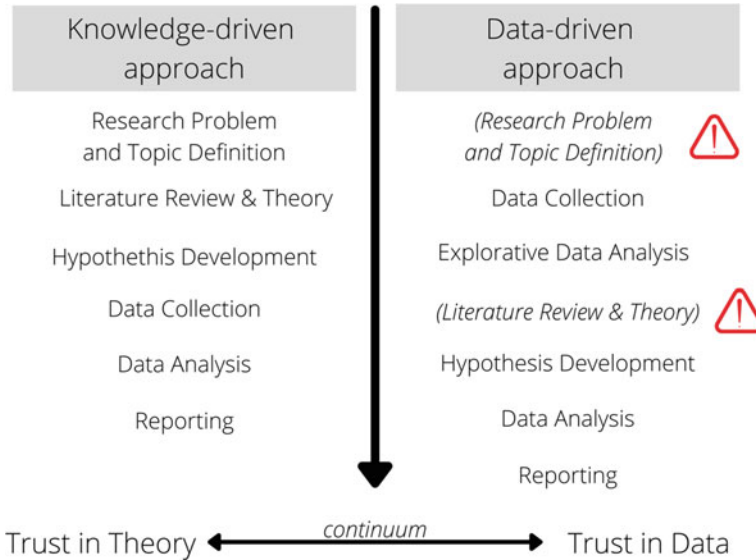
## 4 Conclusion

The expansion of data science methods in tourism has gained ground and is expected to provoke even greater attention in the upcoming years. However, if one is only attracted to the bright side of data science techniques and the idea of novel research methods, then we lose sight of the spirit of social scientists as we turn a blind eye towards its epistemological challenges and pitfalls (Bannister & O’Sullivan, 2021).

At bottom, data-driven analytics has never been a one-size-fits-all solution as it comes with its own strengths and set of drawbacks. As discussed above, numerous epistemological challenges can emerge, starting already in the initial phase, since, in the search of new contexts and patterns driven by correlations, data-intensive science often relies heavily not only on theories but also on the nature of datasets. Consequently, so as to achieve optimal results, extensive computer science know-how remains an essential element in data science research. On that note, major advantages and disadvantages associated with data-driven approaches have been summarised in Table 1.

**Table 1** Advantages and disadvantages of data-driven analytics

<b>Advantage</b>	<b>Criticism</b>
Reveals new contexts and patterns	Not theory-based
Accesses huge databases	Driven by correlations
Works with real-time data	Data does not represent social reality
Shortens the period of data collection	Complex procedures; extensive computer science know-how required
Minimises bias such as social desirability	Requirements for quality criteria are to be critically questioned
Predictive and prognostic	Results highly dependent on hyperparameter tuning



**Fig. 2** The continuum between trust in theory and trust in data. Source: illustrated by the authors

As illustrated in Fig. 2, the transition between the knowledge-driven and the data-driven approach can be understood as a continuum. For Kempeneer (2021), this continuum spans between trust in theory and trust in data. Certainly, there is nothing wrong with “letting the data speak for itself” (Anderson, 2008), yet, this should only be valid for an intermediate step in the data-driven process. What merits the most attention is the first phase of defining a research problem and topic. The act of collecting certain data and applying specific data science methods implies that one does have “something” in mind—this can be understood as a research question, a hypothesis, or even just curiosity/an incentive towards a particular phenomenon. At the same time, this suggests that one has an understanding of the field, which is consistent with the nature of knowledge-based practices (even if the theory itself may not be extensive at this point).

What typically follows next in knowledge-driven approaches, before collecting the data, are reviewing previous literature and formulating a hypothesis. These steps, however, are reversed in data-driven science (Hey et al., 2009)—based on having relevant and logical objectives in mind, researchers applying data science methods proceed with data collection first, as discussed earlier. As soon as some unknown patterns arise, researchers then become curious and analyse the why (i.e. reasons behind certain patterns and their correlations), which reflects our earlier claims that data can speak for itself. At this point, at the latest, a link to existing theories must also be established. When using data to search for patterns and correlations, researchers implicitly know that a relationship exists. This claim can be supported by Kempeneer’s (2021) study, arguing that correlations are sufficient to support a

data-driven representation of reality notwithstanding that they do not explicitly explain certain phenomena.

However, knowledge gain is simply a limited perspective, and, in science, the question goes beyond whether something holds true. Only by knowing the why and the how can we explain the world, and only through being able to explain the world can we generalise the results. Consequently, inductive reasoning also comes with some amount of theory, at least to the extent that we need to guide our research. Furthermore, in order to generate new knowledge from huge databases, they must be linked to theories (Rivera, 2020). In other words, one does perhaps have a light-weight theory in mind, which allows us to identify patterns and use results as “indicators” for further research rather than abruptly pausing at the step concerning discovering patterns via exploratory analysis.

All in all, because of data being used in prediction or to support business decision-making, the scientific community was, so to say, “thrown in at the deep end” of the process, losing all control that it traditionally had a grip on. In this sense, the fractures between business and science, on the one hand, and between business methods and research ethics, on the other, combined with additional access issues, have caused ongoing tensions at an epistemological level and continue to push science outside of academia. To best leverage data science techniques and address social inquiry in other research domains, it is important to ensure that digital methods’ needs are “sustained by an abductive, intersubjective and plural epistemological framework that allows to profitably include big data and computation within the different paradigmatic traditions that coexist in our disciplines” (Amaturo & Aragona, 2021, p. 1). In addition to the epistemological challenges that have emerged in data-driven science, these issues go hand in hand with understanding an “appropriate” way of using data and interpreting the results (Mittelstadt & Floridi, 2016).

## References

- Alexander, N. (2015). What’s more general than a whole population? *Emerging Themes in Epidemiology*, 12(1), 1–5.
- Ali, I. M., Jusoh, Y. Y., Abdullah, R., Nor, R., Nor, H., & Affendey, L. S. (2019). Measuring the performance of big data analytics process. *Journal of Theoretical and Applied Information Technology*, 97(14), 3796–3808.
- Allenby, G. M., Bradlow, E. T., George, E. I., Liechty, J., & McCulloch, R. E. (2014). Perspectives on Bayesian methods and big data. *Customer Needs and Solutions*, 1(3), 169–175.
- Amaturo, E., & Aragona, B. (2021). Digital methods and the evolution of the epistemology of social sciences. In P. Mariani & M. Zenga (Eds.), *Studies in classification, data analysis, and knowledge organization. Data science and social research II* (pp. 1–8). Springer International.
- Anderson, C. (2008). The end of theory: The data deluge makes the scientific method obsolete. *Wired Magazine*, 16(7), 16–07.
- Arora, N., & Gabrani, G. (2018). Significant machine learning and statistical concepts and their applications in social computing. In *2018 first international conference on secure cyber computing and communication* (pp. 57–61). IEEE.

- Balazka, D., & Rodighiero, D. (2020). Big data and the little big bang: An epistemological (R)evolution. *Frontiers in Big Data*, 3, 1–13.
- Bannister, J., & O'Sullivan, A. (2021). Big Data in the city. *UrbanStudies*, 1–10.
- Barredo-Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115.
- Barrowman, N. (2018). Why data is never raw. *The New Atlantis*, 56, 129–135.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Canali, S. (2016). Big Data, epistemology and causality: Knowledge in and knowledge out in EXPOsOMICS. *Big Data and Society*, 3(2), 205395171666953.
- Darbellay, F., & Stock, M. (2012). Tourism as complex interdisciplinary research object. *Annals of Tourism Research*, 39(1), 441–458.
- DuBravac, S., & Shapiro, G. (2015). *Digital destiny: How the new age of data will transform the way we work, live, and communicate*. Regnery Publishing.
- Egger, R. (2022a). Machine learning in tourism—A brief overview. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 85–107). Springer.
- Egger, R. (2022b). Topic modeling. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 375–403). Springer.
- Egger, R., Kroner, M., & Stöckl, A. (2022a). Web scraping. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 67–84). Springer.
- Egger, R., Neuburger, L., & Mattuzzi, M. (2022b). Data science & ethical issues. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 51–66). Springer.
- Elragal, A., & Klischewski, R. (2017). Theory-driven or process-driven prediction? Epistemological challenges of big data analytics. *Journal of Big Data*, 4(1), 1–20.
- George, G., Osinga, E. C., Lavie, D., & Scott, B. A. (2016). Big data and data science methods for management research. *Academy of Management Journal*, 59(5), 1493–1507.
- Gray, J., & Szalay, A. (2007). *eScience: A transformed scientific method*. National Research Council, Mountain View, CA. Retrieved from <https://www.slideshare.net/dullhunk/escience-a-transformed-scientific-method>
- Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly*, 611–642.
- Hannigan, T. R., Haans, R. F. J., Vakili, K., Tchalian, H., Glaser, V. L., Wang, M. S., Kaplan, S., & Jennings, P. D. (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, 13(2), 586–632.
- Harford, T. (2014). Big data: A big mistake? *Significance*, 11(5), 14–19.
- Hargittai, E. (2020). Potential biases in big data: Omitted voices on social media. *Social Science Computer Review*, 38(1), 10–24.
- Hauer, M., & Bohon, S. (2020). Causal inference in population trends: Searching for demographic anomalies in big data. *SocArXiv*, 1–45.
- Hey, T., Tansley, S., & Tolle, K. (2009). *The fourth paradigm: Data-intensive scientific discovery* (Vol. 317). Microsoft Research.
- Kempeneer, S. (2021). A big data state of mind: Epistemological challenges to accountability and transparency in data-driven regulation. *Government Information Quarterly*, 101578.
- Kim, H. J., & Pedersen, S. (2010). Young adolescents' metacognition and domain knowledge as predictors of hypothesis-development performance in a computer-supported context. *Educational Psychology*, 30(5), 565–582.
- Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data and Society*, 1(1), 205395171452848.
- Klahr, D., Fay, A. L., & Dunbar, K. (1993). Heuristics for scientific Experimentation: A developmental study. *Cognitive Psychology*, 25, 111–146.
- Kraska, T., Talwalkar, A., Duchi, J. C., Griffith, R., Franklin, M. J., & Jordan, M. I. (2013). MLbase: A distributed machine-learning system. *Cidr*, 1, 1–7.

- Leonelli, S. (2018). Rethinking reproducibility as a criterion for research quality. In *Including a symposium on Mary Morgan: Curiosity, imagination, and surprise* (Research in the history of economic thought and methodology, Vol. 36B, pp. 129–146). Emerald Publishing.
- Levallois, C., Steinmetz, S., & Wouters, P. (2013). Sloppy data floods or precise social science methodologies? Dilemmas in the transition to data-intensive research in sociology and economics. In *Virtual Knowledge: Experimenting in the Humanities and the Social Sciences* (pp. 151–182). MIT Press.
- Lukosius, V., & Hyman, M. R. (2019). Marketing theory and big data. *The Journal of Developing Areas*, 53(4), 1–9.
- Mariani, M., Baggio, R., Fuchs, M., & Höpken, W. (2018). Business intelligence and big data in hospitality and tourism: A systematic literature review. *International Journal of Contemporary Hospitality Management*, 30(12), 3514–3554.
- Mayer-Schönberger, V., & Cukier, K. (2013). *Big data: A revolution that will transform how we live, work, and think* (First Mariner Books edition). Mariner Books Houghton Mifflin Harcourt.
- Mazanec, J. A. (2020). Hidden theorizing in big data analytics: With a reference to tourism design research. *Annals of Tourism Research*, 83, 102931.
- McAbee, S. T., Landis, R. S., & Burke, M. I. (2017). Inductive reasoning: The promise of big data. *Human Resource Management Review*, 27(2), 277–290.
- McFarland, D. A., & McFarland, H. R. (2015). Big Data and the danger of being precisely inaccurate. *Big Data and Society*, 2(2), 205395171560249.
- Miah, S. J., Vu, H. Q., Gammack, J., & McGrath, M. (2017). A big data analytics method for tourist behaviour analysis. *Information and Management*, 54(6), 771–785.
- Mittelstadt, B. D., & Floridi, L. (2016). The ethics of big data: Current and foreseeable issues in biomedical contexts. *Science and Engineering Ethics*, 22(2), 303–341.
- Müller, O., Junglas, I., vom Brocke, J., & Debortoli, S. (2016). Utilizing big data analytics for information systems research: Challenges, promises and guidelines. *European Journal of Information Systems*, 25(4), 289–302.
- Nukhu, R., & Singh, S. (2020). Branding dilemma: The case of branding Hyderabad city. *International Journal of Tourism Cities*, 6(3), 545–564.
- Oviedo-García, M. Á. (2016). Tourism research quality: Reviewing and assessing interdisciplinarity. *Tourism Management*, 52, 586–592.
- Prensky, M. (2009). H. sapiens digital: From digital immigrants and digital natives to digital wisdom. *Innovate: Journal of Online Education*, 5(3), 1–11.
- Rai, A. (2016). Editor's comments: Synergies between big data and theory. *MIS Quarterly*, 40(2), iii–ix.
- Redman, T. C. (2014). Data doesn't speak for itself. *Harvard Business Review*, 1–5.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.
- Rivera, M. A. (2020). Big data research in hospitality: From streetlight empiricism research to theory laden research. *International Journal of Hospitality Management*, 86, 102447.
- Rizk, A., & Elragal, A. (2020). Data science: Developing theoretical contributions in information systems via text analytics. *Journal of Big Data*, 7(1), 1–26.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215.
- Schrock, A., & Shaffer, G. (2017). Data ideologies of an interested public: A study of grassroots open government data intermediaries. *Big Data and Society*, 4(1), 205395171769075.
- Seaver, N. (2017). Algorithms as culture: Some tactics for the ethnography of algorithmic systems. *Big Data and Society*, 4(2), 205395171773810.
- Sharma, R., Mithas, S., & Kankanhalli, A. (2014). Transforming decision-making processes: A research agenda for understanding the impact of business analytics on organisations. *European Journal of Information Systems*, 23(4), 433–441.
- Simsek, Z., Vaara, E., Paruchuri, S., Nadkarni, S., & Shaw, J. D. (2019). New ways of seeing big data. *Academy of Management Journal*, 62(4), 971–978.



- Singleton, A., & Arribas-Bel, D. (2021). Geographic data science. *Geographical Analysis*, 53(1), 61–75.
- Slota, S. C., Hoffman, A. S., Ribes, D., & Bowker, G. C. (2020). Prospecting (in) the data sciences. *Big Data and Society*, 7(1), 205395172090684.
- Song, I. Y., & Zhu, Y. (2016). Big data and data science: What should we teach? *Expert Systems*, 33(4), 364–373.
- Steadman, I. (2013). *Big data, language and the death of the theorist*. WIRED UK. Retrieved from <https://www.wired.co.uk/article/big-data-end-of-theory>
- Such-Devesa, M. J., Ramón-Rodríguez, A., Aranda-Cuéllar, P., & Cabrera, A. (2021). Airbnb and overtourism: An approach to a social sustainable model using Big Data. In *strategies in sustainable tourism, economic growth and clean energy* (pp. 211–233). Springer.
- Swan, M. (Ed.). (2015). *Philosophy of big data: Expanding the human-data relation with big data science services*. IEEE.
- Symons, J., & Alvarado, R. (2016). Can we trust Big Data? Applying philosophy of science to software. *Big Data and Society*, 3(2), 205395171666474.
- Torabi Asr, F., & Taboada, M. (2019). Big Data and quality data for fake news and misinformation detection. *Big Data and Society*, 6(1), 205395171984331.
- Tribe, J. (2010). Tribes, territories and networks in the tourism academy. *Annals of Tourism Research*, 37(1), 7–33.
- Vargas, R. A. (2020). The onto-epistemology of Big Data. *Sapientiae*, 5(2), 286–294.
- Xiang, Z. (2018). From digitization to the age of acceleration: On information technology and tourism. *Tourism Management Perspectives*, 25, 147–150.
- Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350–361.



Roman Egger and Joanne Yu

## Learning Objectives

- Understand the interplay between computer science, mathematics and statistics, and domain knowledge
- Outline the core competencies in data science research
- Clarify the integration of tourism and data science
- Explain the scope of available software solutions

## 1 Introduction

Tourism research originates from its constitution of knowledge involving various spheres of study (Darbellay & Stock, 2012), which include, but are not limited to, economics and business, sociology, anthropology, psychology, geography, humanities, and applied sciences (Oviedo-García, 2016). However, the fact that scholars have different areas of expertise potentially leads to an issue of fragmented knowledge construction (Darbellay & Stock, 2012). That is, depending on one's research field(s), previous research findings and conceptual frameworks might not be as well-

---

R. Egger (✉)

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

e-mail: [roman.egger@fh-salzburg.ac.at](mailto:roman.egger@fh-salzburg.ac.at)

J. Yu

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

Department of Tourism and Service Management, Modul University Vienna, Vienna, Austria

e-mail: [joanne.yu@modul.ac.at](mailto:joanne.yu@modul.ac.at)

connected to each other since they may have been developed and verified based on pre-existing models with a discipline-specific background (Xiang, 2018). In order to investigate tourism phenomena and to justify tourism science as an independent subject, knowledge integration through an interdisciplinary approach is critical (Leiper, 1981). Additionally, advancements in computational social sciences (Xu & Wu, 2018) have fostered a paradigm shift in knowledge creation from traditional research techniques based on theory to the empirical-then-theoretical approach (Lehmann, 2020). Since the era of digitalisation, this conversion can be seen through the exceptional growth of tourism studies that have applied empirical methods (Li et al., 2020; Xu & Wu, 2018) with a particular focus on data science techniques (Li et al., 2013; Mariani, 2019; Oviedo-García, 2016; Arefieva et al., 2021). Hence, with the reinforcement of information and communication technologies (ICTs) and data science, tourism research is no longer limited to conceptual frameworks rooted in other domains (Xiang, 2018). In fact, the interwoven disciplines of data science are also manifold, including computer science, mathematics and statistics, and the respective domain knowledge.

One of the central questions for long-term development in a data-rich environment is how and with whom the corresponding levels of knowledge should be connected. One such avenue can involve cross-disciplinary collaborations (Mariani et al., 2018) between tourism scholars and data scientists (Arefieva et al., 2021). Conventionally, tourism-specific domain expertise is paired with statistical knowledge, which results in traditional research. Contrariwise, people specialising in mathematics and computer science are capable of elevating traditional statistics to computational statistics (Mariani et al., 2018). Since tourism experts in academia and the industry in general often lack the necessary computer science competencies, and vice versa, interdisciplinary teams are of crucial importance when applying data science methods to tourism research. However, despite the increasing popularity and interest in integrating multifaceted knowledge, a general framework that bridges data science and tourism has yet to be developed (Mariani et al., 2018; Xiang, 2018). To assist tourism scholars in identifying and using the appropriate analytical tools, this chapter aims to bridge and theorise the importance of core competencies regarding the data science process and, thus, to advocate for interdisciplinary collaboration in hopes of delivering comprehensive and easy-to-understand scientific knowledge to tourism.

Following this brief introduction, Sect. 2 will start off with problem identification. Thereafter, Sect. 3 will theorise the interplay between statistical knowledge, domain expertise, and computer science skills, whilst Sect. 4 will then outline the specific procedures of data science research, from formulating topics and extracting data to analysing and interpreting results. Finally, Sect. 5 will present implications for tourism research projects and conclude with practical implementations.

## 2 Problem Identification

Apart from the overall growing academic interest in data science, its increasing use within the global tourism industry can also be steadily observed. For instance, ICTs and data science are beneficial to alerting and monitoring systems, personalisation and recommendation engines, and pricing and demand optimisation, amongst many others. Furthermore, as the twenty-first century has brought about many environmental crises (e.g., the Australian bushfires and the floods in Venice) alongside the COVID-19 pandemic, data science plays a critical role in helping the industry to adapt to the resumption of tourism. Since ICTs are ultimately the enablers of collaborations and value co-creation between various stakeholders (e.g., tourists, travel agencies, destination marketers, tourism technology providers, etc.) in a determined ecosystem (Buhalis, 2015; Femenia-Serra et al., 2019), businesses should explore solutions that comply with regulations and restrictions whilst maximising the level of tourist experiences through processing and analysing datasets. However, as such, a large volume of data would be undervalued if researchers were unaware of the potential of advanced data analytics, had no access to the data, or were unfamiliar with analysis processes.

Equipping researchers and practitioners with knowledge from data science is vital so that they can process enormous data chunks to streamline the development of the tourism industry. Typically, academia in tourism follows a linear direction of generating hypotheses and research propositions followed by confirming and validating findings. Big data has shifted the process of epistemological development by allowing researchers to go beyond the status quo and discover new knowledge that may have otherwise remained unrecognised using a data-driven paradigm (Song & Zhu, 2016). Approaches like natural language processing, predictive analytics, real-time analytics, and social media analytics, for instance, have the power to potentially revolutionise existing scientific knowledge (Baldassarre, 2016). Yet, although analytical technologies have already prompted some radical innovations within the tourism industry, it appears that the majority of existing studies are exploratory and ad hoc (Xiang, 2018). The potentiality of using data science to enrich research designs and to improve the generalisability of findings on a larger scale (Mariani, 2019) thus pushes for the development of a guided approach that links tourism researchers to the field of data science (Mariani et al., 2018; Xiang, 2018).

Over the last decades, we have witnessed a growing body of literature attempting to synergise the strengths of data science with tourism management and development (Arefieva et al., 2021; Li et al., 2018; Mariani et al., 2018; Mariani, 2019). Take big data as an example; the data sources range from user-generated content (e.g., Facebook, Instagram, TripAdvisor, etc.) and any type of travel website (e.g., Airbnb, destination websites, Online Travel Agencies, etc.) to sensor data (e.g., geotagging) (Bulencea & Egger, 2013; Li et al., 2018). These sources create the possibility of digital data streams that businesses can use to optimise internal operations, track company performance, improve market positions, and so forth in real-time. Nevertheless, although the above data types and sources have been widely acknowledged

in contemporary tourism, the process of data acquisition (e.g., parsing and crawling data), pre-processing, analysis, evaluation, and interpretation may present researchers with major challenges related to technical, legal, and ethical issues. For instance, unstructured data must be prepared in such a way so that further processing with classical methods from empirical social research or methods from data science is possible. Yet, without an in-depth understanding of applied algorithms (Ceri, 2017), data evaluation and interpretation can be difficult because the numerous black-box models in the field of machine learning make it hard to comprehend what happened during the computation stage (Rudin, 2019).

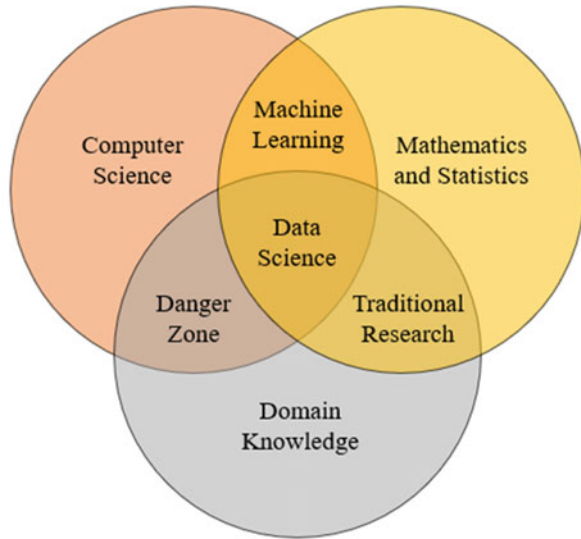
Incorporating and applying data science methods to current tourism research is on the rise, but it is mostly done without explicitly addressing the detailed methodological procedure and the analysis process. Thus, a comprehensible step-by-step description of a study's empirical analysis tends to be missing. This often seems undesirable because the competence to apply such methods leads to a unique selling point and a clear advantage when it comes to the competition of publishing activities.

### 3 Data Science Is an Interdisciplinary Area

Data science in general is concerned with the collection, preparation, analysis, visualisation, management, and preservation of huge amounts of information (Song & Zhu, 2016). Essentially, it focuses on extracting knowledge and insights from structured/unstructured data by means of various scientific methods, processes, algorithms, and systems (Dhar, 2013) in an automated manner (Song & Zhu, 2016). Ideally, data scientists should be equipped with statistical and computational competencies, paired with domain knowledge, in order to analyse and interpret raw data and assist in decision-making processes (Alarcón-Soto et al., 2019). Nevertheless, what is particularly noteworthy within data science is that having specialised knowledge from all relevant fields is not required (Emmert-Streib et al., 2016). Since people who are capable of addressing social issues based on real-world data are scarce (Song & Zhu, 2016), the combined nature of data science further intensifies the need for interdisciplinary collaboration.

Figure 1 presents the pillars of data science including mathematics and statistics, domain knowledge, and computer science (Conway, 2010). Mathematics serves as the basis for solving real-world problems, whereas statistical knowledge allows researchers to choose and apply appropriate techniques in order to extract insights from data (Alarcón-Soto et al., 2019). Having sound knowledge in math and statistics is the precursor to addressing complex issues such as statistical computation and modelling (Prevos, 2017). If one selects unsuitable methods, researchers

**Fig. 1** Data science Venn diagram



might find themselves in the danger zone and thereby face difficulties in solving the pertinent research questions.

Likewise, the discipline of computer science is equally important as it deals with the tools and algorithms for modelling and analysis. Data from the digital environment is typically provided in different formats, such as structured, semi-structured, and/or unstructured data (Song & Zhu, 2016); this is where coding comes in to aid in data extraction, manipulation, and preparation of the datasets. Thus, scientists are required to have intensive hands-on experience and knowledge in computer languages, such as Python,<sup>1</sup> R,<sup>2</sup> and MATLAB.<sup>3</sup> Moreover, researchers need to be equipped with a holistic understanding of data in order to be able to accurately derive meanings and insights based on the findings (Prevos, 2017). Domain-specific knowledge is necessary when it comes to the consideration of methodological approaches, adjustment of hyperparameters, and application of different algorithms. Overall, it is an iterative process between data analysis and data assessment of the investigated context. Without this domain knowledge, method evaluation and interpreting results would be nearly impossible.

The interplay between mathematical and statistical knowledge and domain expertise is what allows for classical empirical research to emerge in the first place. Typically, data used in traditional research is highly structured and semi-ready for analysis. Nonetheless, Conway (2010) claims that scholars often put too much effort into improving and progressing in their respective areas but invest too little time in

<sup>1</sup>Python: <https://www.python.org/>

<sup>2</sup>R: <https://www.r-project.org/>

<sup>3</sup>MATLAB: <https://www.mathworks.com/>

engaging with the latest trends and developments surrounding advanced analytical technologies. Thus, a major bottleneck during this so-called Information Age is the shortage of researchers capable of solving big data problems (Song & Zhu, 2016).

By excluding domain knowledge, the intersection of applied statistics and knowledge in computer science leads to pure machine learning. Based on the notion of learning-by-data, machine learning enables researchers to effectively uncover trends and patterns from large datasets (Song & Zhu, 2016). Whilst having sound knowledge in programming and machine learning is necessary and promising (Prevos, 2017), what is missing in this regard is a scientific motivation (Baldassarre, 2016). Since big data is intrinsically theory-laden, only a search for potential reasoning behind the observed relationships would advance science-driven understanding (dos Santos, 2016). After all, science is about discovering the unknown and advancing existing knowledge, and it is one's interest in observing and questioning real-world issues that lays the groundwork for developing predictions (Conway, 2010) of emerging social-cultural phenomena, which can be further evaluated via statistical methods.

Lastly, the overlap between computational skills and domain knowledge without sufficient understanding of mathematics and statistics can be viewed as the most serious problem in data science (Baldassarre, 2016). Although researchers might be well-versed in computing science, it is equally critical to understand the underlying mathematical meanings so as to transform statements into theorems (Emmert-Streib et al., 2016). A lack of rigorous methodological structure can be problematic when validation is required, which, in turn, can lead to incorrect analysis. Therefore, even though the sphere of these three disciplines is interdependent, the sweet spot places the realm of data science at the centre. The core value of data science is to transform data into information and actionable knowledge, and this is established by combining knowledge from statistics, programming, and the substantive expertise in the field (Baldassarre, 2016). It serves as the foundation for uncovering original insights based on the pertinent research objectives that, in the end, contribute to all relevant stakeholders in the ecosystem.

## 4 The Importance of Core Competencies in Data Science

Although the term data science might sound frightening and discourage tourism researchers who are unfamiliar with the field, the scope of available software solutions is vast, ranging from pure coding to visual programming, which does not require any coding skills. The sophistication of data science comprises a collection of scientific approaches and methods that uses algorithms to extract information from big data. One common approach involves the cross-industry standard process for data mining (CRISP-DM), which includes the steps of business understanding, data understanding, data preparation, modelling, evaluation, and deployment (Martinez-Plumed et al., 2019). In addition, there are also other methodological process concepts currently being implemented for data science research, such as

Foundational Methodology for Data Science (Rollins, 2015), Team Data Science Process (Ericson et al., 2020), and Rapid Collaborative Data Mining System (Moyle & Jorge, 2001), amongst others.

Based on the commonly adopted procedures mentioned above, seven distinct phases often involved in the data science process will be summarised here; namely, (1) topic formulation and relevance for academia and industry, (2) data access and data collection, (3) data pre-processing, (4) feature engineering, (5) analysis, (6) model evaluation and model tuning, and (7) interpretation of results. Important to note is that the steps are interconnected, and the procedure is highly iterative, involving back-and-forth testing on new features and tuning hyperparameters. There is no doubt that these steps require researchers to have various technical and nontechnical skills throughout the multiphase process. For this reason, so as to identify the knowledge and competences required to complete each step of the process successfully, this chapter examines the extent of such necessary key skills. These competencies are evaluated based on one's level of understanding in programming, which can vary from pure coding, through adopting cloud services and application programming interfaces (APIs), to using visual programming solutions without necessary programming skills. To name a few, coding refers to knowledge in programming and scripting languages such as Python<sup>1</sup>, R<sup>2</sup>, and MATLAB<sup>3</sup>; examples of cloud services and APIs for data analytics include IBM Cloud,<sup>4</sup> Google Cloud Platform,<sup>5</sup> Microsoft Azure,<sup>6</sup> and Amazon AWS<sup>7</sup>; and visual programming software can be the use of RapidMiner,<sup>8</sup> KNIME,<sup>9</sup> and Orange 3,<sup>10</sup> amongst others.

By using a purposive sampling technique, this sections draws on insights from the experience and assessments of 120 data scientists. To be more specific, email invitations were sent out to potential participants whose LinkedIn or Medium.com profiles indicated a data science job description. Upon agreement, the participants were invited to complete an online survey to provide their opinion regarding the importance of various competencies needed for data science research. The evaluation was based on a 10-point Likert scale with 1 referring to "not important at all" and 10 referring to "extremely important". The results summarised in Table 1 are structured in such a way that they show the ideal situation for each data science phase and indicate the programming competence for each of the three competence fields. An in-depth discussion of the matrix is also presented below.

The first phase, topic formulation, is critical because it serves as the basis for successful resolution of business problems (Martinez-Plumed et al., 2019). Not only do researchers need to ensure the relevance of the data for both academia and

---

<sup>4</sup>IBM Cloud: <https://cloud.ibm.com/developer/watson/dashboard>

<sup>5</sup>Google Cloud: <https://cloud.google.com/solutions/smart-analytics>

<sup>6</sup>Microsoft Azure API: <https://azure.microsoft.com/>

<sup>7</sup>Amazon AWS: <https://aws.amazon.com/>

<sup>8</sup>RapidMiner: <https://rapidminer.com/>

<sup>9</sup>KNIME: <https://www.knime.com/>

<sup>10</sup>Orange 3: <https://orange.biolab.si/>



**Table 1** Fields of competencies in data science

Data science phase	Programming competency					
	High		Medium		Low	
	Coding		Cloud services and APIs		Visual programming tools	
	Mean	SD	Mean	SD	Mean	SD
Topic formulation and relevance for academia and industry						
Computer science	6.64	2.98	5.46	2.59	4.90	2.71
Mathematics and statistics	6.80	2.75	5.39	2.56	5.40	2.58
Domain knowledge	<b>7.52</b>	2.57	<b>6.85</b>	2.67	<b>6.62</b>	2.72
Data access and data collection						
Computer science	<b>7.70</b>	2.19	<b>7.03</b>	2.41	5.35	2.71
Mathematics and statistics	6.49	2.78	5.71	2.60	5.31	2.69
Domain knowledge	7.38	2.28	7.02	2.26	<b>6.48</b>	2.45
Data pre-processing						
Computer science	<b>8.21</b>	1.98	6.55	2.44	5.74	2.72
Mathematics and statistics	7.33	2.35	5.97	2.36	5.91	2.60
Domain knowledge	7.51	2.29	<b>6.58</b>	2.41	<b>6.44</b>	2.47
Feature engineering						
Computer science	<b>8.03</b>	2.00	6.27	2.42	5.96	2.69
Mathematics and statistics	7.73	2.05	6.16	2.27	6.13	2.59
Domain knowledge	7.70	2.21	<b>6.58</b>	2.36	<b>6.67</b>	2.37
Analysis						
Computer science	<b>8.49</b>	1.89	6.23	2.50	6.23	2.70
Mathematics and statistics	8.44	1.74	<b>6.49</b>	2.35	<b>6.50</b>	2.59
Domain knowledge	7.45	2.50	6.36	2.52	6.39	2.61
Model evaluation and model tuning						
Computer science	8.30	1.89	6.30	2.35	6.15	2.62
Mathematics and statistics	<b>8.41</b>	1.68	<b>6.58</b>	2.48	<b>6.58</b>	2.62
Domain knowledge	7.54	2.26	6.45	2.39	6.48	2.54
Interpretation of results						
Computer science	7.16	2.51	6.30	2.60	6.94	2.91
Mathematics and statistics	7.87	2.28	6.57	2.62	7.05	2.76
Domain knowledge	<b>7.94</b>	2.14	<b>6.99</b>	2.41	<b>7.43</b>	2.45

industries in order to address real-life issues, but expert knowledge in the respective domain is also needed to convert data into actionable insights. Take a hotel revenue forecasting task as an example; here, the integration of domain knowledge and computational skills is essential for developing an effective revenue management system, especially when it comes to managing group reservations in tourist attractions where the distribution of guest arrivals can hardly be pre-defined (El Gayar et al., 2009). Thus, supplying in-depth domain expertise may facilitate researchers and marketers to address new problems in hospitality and tourism curricula and to achieve the desired outcomes (Ogbeide et al., 2020).

Once a study context/subject has been defined, researchers need to gather relevant data, which can be structured, unstructured, and/or semi-structured. Due to sophisticated information systems, accessing and automatically collecting data has become a trend for businesses (Damangir et al., 2018), and, therefore, computing skills are considered highly valuable for applying cloud-based services or using programming tools to extract data. For instance, APIs allow tourism researchers to automatically identify and crawl the geographical information of a service entity (Vu et al., 2019). In this regard, scholars have adopted Python to scrape all available online reviews on [Booking.com](#) (Mariani et al., 2019), Facebook posts published by destination marketing organisations (Mariani et al., 2016), and Instagram pictures shared by tourists (Arefieva et al., 2021). Nonetheless, even without coding skills, this does not necessarily mean that researchers cannot incorporate the discipline of data science into the context of tourism. A study conducted by Yu, Xie, and Wen (2020), for example, underpins the effectiveness of using visual web data extraction software (e.g., Octoparse<sup>11</sup>) to extract tourism related data (as long as data sources can be identified). In a nutshell, data sources relating to tourism are usually dominated by user-generated content (e.g., text and pictures), followed by GPS data and web search data (Li et al., 2018).

Next, data pre-processing attempts to clean, reconstruct, convert, and transform data to prepare for subsequent stages in the data science process, such as modelling, analysis, and visualisation. For instance, in natural language processing, typical techniques include stopword removal, stemming, lemmatisation, tokenisation, and vectorisation, amongst many others. Since real-world data, especially data from user-generated content, is unstructured and inconsistent in nature (Qi et al., 2018), data pre-processing tends to be one of the most salient and time-consuming phases. As such, exploratory statistics plays a necessary role in streamlining data pre-processing so as to better understand a dataset (Weihs & Ickstadt, 2018). In addition to data pre-processing using a programming language, tourism researchers have also proposed other alternatives for those unfamiliar with coding (Supak et al., 2017), such as finding solutions with a graphical user interface (e.g., Orange 3<sup>10</sup>, OpenRefine,<sup>12</sup> and DataCleaner<sup>13</sup>).

By involving domain knowledge, feature engineering can extract and generate new features from the raw data, thus improving machine learning performance. However, despite the wide range of toolkits available for the automation of this process (e.g., auto-sklearn,<sup>14</sup> AutoKeras,<sup>15</sup> and Featuretools<sup>16</sup>), manual inspection is often required to determine input features for the fitting process. An example can be taken from tourism demand forecasting; although feature engineering streamlines

---

<sup>11</sup> Octoparse: <https://www.octoparse.com/>

<sup>12</sup> OpenRefine: <https://openrefine.org/>

<sup>13</sup> DataCleaner: <https://datacleaner.org/>

<sup>14</sup> Auto-sklearn: <https://automl.github.io/auto-sklearn/master/>

<sup>15</sup> AutoKeras: <https://autokeras.com/>

<sup>16</sup> Featuretools: <https://www.featuretools.com/>

the process of constructing models by identifying the most relevant features from a long list of potential factors likely to influence outcomes (Law et al., 2019), insufficient computational competencies and the infancy of artificial intelligence (Moro et al., 2019) entail the necessity of domain knowledge from tourism experts when performing modelling techniques (Law et al., 2019). As such, this chapter particularly points out the significance of automated feature engineering. Data scientists have recently proposed the deep feature synthesis algorithm as a solution to achieve automation (Kanter & Veeramachaneni, 2015). In this way, instead of encoding knowledge about a specific context, deep feature synthesis can automatically generate features between relational data and synthesise the process of model building.

Whilst data scientists scrutinise data and examine the unknown by prototyping, modelling, and writing algorithms, data analysts identify trends and patterns within the large datasets to draw conclusions and support business decisions. As such, data analysis requires an understanding of mathematical statistics, data modelling, programming, reporting, data visualisation, and database management. In addition to the wide array of tools available for data analysis, ranging from professional to free open-source software, the increasing use of social media along with the progression of widespread Internet connectivity has made big data solutions more accessible. Therefore, data analysis emerging as a component in the development of smart tourism can already be witnessed as a trend (Xiang & Fesenmaier, 2017). For instance, a recent study used RapidMiner<sup>8</sup> for network analysis in order to develop a novel approach for intelligence practices in the hotel industry (Köseoglu et al., 2020), whilst another study adopted Orange 3<sup>10</sup> for text analysis to evaluate consumer experiences in the airline industry (Bodnár et al., 2020). Other popular techniques provided in the tourism literature include linear and nonlinear regression, time-series analysis, topic modelling, sentiment analysis, statistical analysis, clustering, text summarisation, and dependency modelling (Li et al., 2018). Note that, whilst the above examples feature visual programming software, basic statistical knowledge is still a prerequisite for choosing a suitable method.

Nevertheless, researchers need to evaluate model performance as well, with one of the goals being to avoid over-fitting or under-fitting and to achieve an unbiased estimate. The iterative process between hyperparameter tuning (see chapter “Hyperparameter Tuning” for more details) and model evaluation additionally calls for the importance of knowledge in mathematics and statistics, regardless of the programming tool used in a study. One of the common methods includes k-fold cross-validation; for example, given the large amount of data extracted from TripAdvisor, Moro (2020) applied ten-fold cross-validation and used mean absolute error (MAE) and mean absolute percentage error (MASE) as performance metrics to ensure the robustness of the results. Metrics such as F1 score, precision, recall, AUC-ROC, Gini Coefficient, Information Gain, or Log Loss, amongst others, have evolved to evaluate predictive models. Moreover, researchers have affirmed that the performance of models in tourism forecasting largely depends on the choice of hyperparameters (Abellana et al., 2020). More details regarding this topic can be found in the chapters that follow.

Finally, one needs to constantly bear in mind that data does not speak for itself (Kitchin, 2014). When it comes to data-driven approaches, interpreting results is the “human” part, thereby making expert knowledge indispensable for deriving meaningful insights (Alarcón-Soto et al., 2019). More specifically, the data interpretation phase can be best characterised as a mix of abductive and iterative research (Oliver & Vayre, 2015). All in all, researchers should equip themselves with sufficient theoretical knowledge in order to refine and interpret results as hypothetical/theoretical characters are inherent in most of the big data tourism studies (Mazanec, 2020; Egger & Yu, 2022). For example, theories grounded in emotional psychology are in support of a sentiment analysis interpretation when analysing tourism online reviews (Yu & Egger, 2021). Similarly, an understanding of psychographics and behavioural theories is needed to explain users’ information behaviour (Ma et al., 2020). Essentially, as data science communities are bound to implicit assumptions (Mazanec, 2020), it is not only the methodological understanding that is important, but having sound knowledge in relation to the study context from a content perspective is also relatively crucial.

## 5 Conclusion

With the increasing popularity of automated machine learning, data science in the twenty-first century has been more accessible than ever (Egger, 2022). Tools are becoming easier to use, and algorithms can do hyperparameter tuning on their own. In fact, some businesses have already initiated advanced analytical technologies that expand the usual scope of applications. For instance, as the emerging leader of sharing economy, Uber recently released Ludwig v0.3,<sup>17</sup> a no-code machine learning platform (Addair et al., 2020). The possibilities of Ludwig (e.g., automated hyperparameter optimisation, integration with weights and biases, and code-free interface) are changing traditional practice by making machine learning possible for everyone. Similarly, Microsoft Azure<sup>18</sup> empowers both professional and non-professional scientists to automatically build and deploy predictive models through a no-code user interface.

Realistically speaking, although automatic services have emerged in areas such as natural language processing, speech recognition, reinforcement learning, image recognition, machine learning, and much more, implications and conclusions cannot be generated in one click. The fact that data does not speak for itself goes back to the significant point of interdisciplinary collaboration. No matter how sophisticated a tool might be, models still need to be evaluated, algorithms and hyperparameters need to be fine-tuned, and, finally, results need to be compared and transferred into

---

<sup>17</sup>Uber Engineering: <https://eng.uber.com/ludwig-v0-3/>

<sup>18</sup>Microsoft Azure. Automated machine learning: <https://azure.microsoft.com/en-us/services/machine-learning/automatedml/>

practice. Indeed, as automation continues disentangling maintenance tasks, coding and programming skills might not be sought after as much by tourism scholars and practitioners. However, whilst computer scientists advance the development of algorithms and automated services, users are required to have the competencies to effectively use those systems. That said, in order to infer meanings correctly, researchers must be able to understand the machines and the intuition of algorithms.

The process of collecting and analysing consumer data and digital footprints is nothing new to tourism and hospitality (Egger, 2007). The rise of artificial intelligence with machine learning-based applications along with Internet-of-Things (IoT) devices providing us with an unimagined wealth of data, and much more, continues to transform how scientific and market research can be conducted. Researchers and businesses embrace big data analytics to uncover insights that can optimise decisions and strategies. Whilst data is the fuel that provides additional value, the results generated from data analytics need to be validated further (Lukosius & Hyman, 2019). From topic formulation to data collection and interpretation of results, the iterative nature of the process occurs in alignment with a deep understanding of the knowledge through theory matching, empirical findings, and real-life observations (Saragih et al., 2019). After all, it can be noted that, essentially, each phase in the data science process requires adequate theorisation and awareness of epistemological challenges (Müller et al., 2016) so as to properly interpret analytical findings based on the respective disciplinary backgrounds and industry practices.

In the foreseeable future, for those involved in the tourism industry and academia to benefit from data to the fullest, curricula should further equip people with the basic foundations of machine learning as well as the methodological principles of properly using machines and cloud systems. Like mathematics and statistics, which have long been the fundamentals of business and management, having sound knowledge in data analytics should be considered equally important. In this way, what is needed for scholars and practitioners in the tourism field will no longer be the coding itself but a thorough understanding of fast-developing algorithms and the various methods and techniques for different cases/scenarios. The wide range of skills needed in the modern age of data science reinforces the significance of interdisciplinary collaboration; hence, the knowledge of tourism experts should be paired with that of computer scientists and statisticians', and vice versa, in order to refine data, attain implications, and consolidate or advance new theories.

## References

- Abellana, D. P. M., Rivero, D. M. C., Aparente, M. E., & Rivero, A. (2020). Hybrid SVR-SARIMA model for tourism forecasting using PROMETHEE II as a selection methodology: A Philippine scenario. *Journal of Tourism Futures*, 7(1), 78–97.
- Addair, T., Molino, P., & Dudin, Y. (2020). *Ludwig v0.3 introduces hyperparameter optimization, transformers and TensorFlow 2 support*. Retrieved from <https://eng.uber.com/ludwig-v0-3/>

- Alarcón-Soto, Y., Espasandín-Domínguez, J., Guler, I., Conde-Amboage, M., Gude-Sampedro, F., Langohr, K., & Gómez-Melis, G. (2019). Data science in biomedicine. *arXiv preprint arXiv:1909.04486*.
- Arefieva, V., Egger, R., & Yu, J. (2021). A machine learning approach to cluster destination image on Instagram. *Tourism Management*, 85, 104318.
- Baldassarre, M. (2016). Think big: Learning contexts, algorithms and data science. *Research on Education and Media*, 8(2), 69–83.
- Bodnár, M., Jackle, F., & Linzner, T. (2020). Exploring the difference in perception of service quality of Low Cost Carrier customers through online reviews: Social Media Analysis. In *ISCONTOUR 2020 tourism research perspectives: Proceedings of the international student conference in tourism research* (pp. 231–242). BoD–Books on Demand.
- Buhalis, D. (2015). *Working definitions of smartness and smart tourism destination*. Retrieved from <http://buhalis.blogspot.co.uk/2014/12/working-definitions-of-smartness-and.html>
- Bulencea, P., & Egger, R. (2013). Facebook it: Evaluation of Facebook’s search engine for travel related information retrieval. In *Information and communication technologies in tourism 2014* (pp. 467–480). Springer.
- Ceri, S. (2017). On the big impact of “big computer science”. In *Informatics in the future* (pp. 17–26). Springer.
- Conway, D. (2010). *The data science Venn diagram*. Retrieved from <http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>
- Damangir, S., Du, R. Y., & Hu, Y. (2018). Uncovering patterns of product co-consideration: A case study of online vehicle Price quote request data. *Journal of Interactive Marketing*, 42, 1–17.
- Darbellay, F., & Stock, M. (2012). Tourism as complex interdisciplinary research object. *Annals of Tourism Research*, 39(1), 441–458.
- Dhar, V. (2013). Data science and prediction. *Communications of the ACM*, 56(12), 64–73.
- Dos Santos, R. (2016). Big Data: Philosophy, emergence, crowdledge, and science education. *Themes in Science and Technology Education*, 8(2), 115–127.
- Egger, R. (2007). *Cyberglobetrotter–Touristen im Informationszeitalter*.
- Egger, R. (2022). *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies, and applications*. Springer.
- Egger, R., & Yu, J. (2022). Epistemological challenges. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 17–34). Springer.
- El Gayar, N., Zakhary, A., Aziz, H. A., Saleh, M., Atiya, A., & El Shishiny, H. (2009). A new approach for hotel room revenue maximization using advanced forecasting and optimization methods. *Data Mining for Improving Tourism Revenue in Egypt*, 1–11.
- Emmert-Streib, F., Moutari, S., & Dehmer, M. (2016). The process of analyzing data is the emergent feature of data science. *Frontiers in Genetics*, 7, 12.
- Ericson, G., Rohm, W. A., Martens, J., Sharkey, K., Casey, C., Harvey, B., & Schonning, N. (2020). *What is the team data science process?* Retrieved from <https://docs.microsoft.com/en-gb/azure/machine-learning/team-data-science-process/overview>
- Femenia-Serra, F., Neuhofer, B., & Ivars-Baidal, J. A. (2019). Towards a conceptualisation of smart tourists and their role within the smart destination scenario. *The Service Industries Journal*, 39(2), 109–133.
- Kanter, J. M., & Veeramachaneni, K. (2015). Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE international conference on data science and advanced analytics* (pp. 1–10). IEEE.
- Kitchin, R. (2014). Big data, new epistemologies and paradigm shifts. *Big Data and Society*, 1(1), 205395171452848.
- Köseoglu, M. A., Mehraliyev, F., Altin, M., & Okumus, F. (2020). Competitor intelligence and analysis (CIA) model and online reviews: Integrating big data text mining with network analysis for strategic analysis. *Tourism Review*, 76(3), 529–552.
- Law, R., Li, G., Fong, D. K. C., & Han, X. (2019). Tourism demand forecasting: A deep learning approach. *Annals of Tourism Research*, 75, 410–423.

- Lehmann, D. R. (2020). The evolving world of research in marketing and the blending of theory and data. *International Journal of Research in Marketing*, 37(1), 27–42.
- Leiper, N. (1981). Towards a cohesive curriculum tourism: The case for a distinct discipline. *Annals of Tourism Research*, 8(1), 69–84.
- Li, N., Buhalis, D., & Zhang, L. (2013). Interdisciplinary research on information science and tourism. In *Information and communication technologies in tourism 2013* (pp. 302–313). Springer.
- Li, J., Xu, L., Tang, L., Wang, S., & Li, L. (2018). Big data in tourism research: A literature review. *Tourism Management*, 68, 301–323.
- Li, M., Lehto, X., & Li, H. (2020). 40 years of family tourism research: Bibliometric analysis and remaining issues. *Journal of China Tourism Research*, 16(1), 1–22.
- Lukosius, V., & Hyman, M. R. (2019). Marketing theory and big data. *The Journal of Developing Areas*, 53(4), 1–9.
- Ma, S. D., Kirilenko, A. P., & Stepchenkova, S. (2020). Special interest tourism is not so special after all: Big data evidence from the 2017 Great American Solar Eclipse. *Tourism Management*, 77, 104021.
- Mariani, M. (2019). Big data and analytics in tourism and hospitality: A perspective article. *Tourism Review*, 75(1), 299–303.
- Mariani, M. M., Di Felice, M., & Mura, M. (2016). Facebook as a destination marketing tool: Evidence from Italian regional Destination Management Organisations. *Tourism Management*, 54, 321–343.
- Mariani, M., Baggio, R., Fuchs, M., & Höepken, W. (2018). Business intelligence and big data in hospitality and tourism: A systematic literature review. *International Journal of Contemporary Hospitality Management*, 30(12), 3514–3554.
- Mariani, M. M., Borghi, M., & Gretzel, U. (2019). Online reviews: Differences by submission device. *Tourism Management*, 70, 295–298.
- Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez Orallo, J., Kull, M., Lachiche, N., Quintana, M. J. A., & Flach, P. A. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 1–14.
- Mazanec, J. A. (2020). Hidden theorising in big data analytics: With a reference to tourism design research. *Annals of Tourism Research*, 83, 102931.
- Moro, S. (2020). Guest satisfaction in East and West: Evidence from online reviews of the influence of cultural origin in two major gambling cities, Las Vegas and Macau. *Tourism Recreation Research*, 1–10.
- Moro, S., Esmerado, J., Ramos, P., & Alturas, B. (2019). Evaluating a guest satisfaction model through data mining. *International Journal of Contemporary Hospitality Management*, 32(4), 1523–1538.
- Moyle, S., & Jorge, A. (2001). *RAMSYS-A methodology for supporting rapid remote collaborative data mining projects*. ECML/PKDD01 workshop: Integrating aspects of data mining, Decision Support and Meta-Learning (Vol. 64, pp. 1–12).
- Müller, O., Junglas, I., Brocke, J. V., & Debortoli, S. (2016). Utilising big data analytics for information systems research: Challenges, promises and guidelines. *European Journal of Information Systems*, 25(4), 289–302.
- Ogbeide, G. C., Fu, Y. Y., & Cecil, A. K. (2020). Are hospitality/tourism curricula ready for big data? *Journal of Hospitality and Tourism Technology*, 12(1), 112–123.
- Oliver, M. A., & Vayre, J. S. (2015). Big data and the future of knowledge production in marketing research: Ethics, digital traces, and abductive reasoning. *Journal of Marketing Analytics*, 3(1), 5–13.
- Oviedo-García, M. Á. (2016). Tourism research quality: Reviewing and assessing interdisciplinarity. *Tourism Management*, 52, 586–592.
- Prevos, P. (2017). Lifting the ‘big data’ veil. Creating value through applied data science. *Water E-Journal*, 2(1), 1–5.

- Qi, S., Wong, C. U. I., Chen, N., Rong, J., & Du, J. (2018). Profiling Macau cultural tourists by using user-generated content from online social media. *Information Technology and Tourism*, 20(1–4), 217–236.
- Rollins, J. (2015). *Why we need a methodology for data science*. Retrieved from <https://www.ibmdatahub.com/blog/why-we-need-methodology-data-science>
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215.
- Saragih, H. S., Simatupang, T. M., & Sunitiyoso, Y. (2019). Co-innovation processes in the music business. *Heliyon*, 5(4), e01540.
- Song, I. Y., & Zhu, Y. (2016). Big data and data science: What should we teach? *Expert Systems*, 33(4), 364–373.
- Supak, S., Brothers, G., Ghahramani, L., & Van Berkel, D. (2017). Geospatial analytics for park & protected land visitor reservation data. In *Analytics in smart tourism design* (pp. 81–109). Springer.
- Vu, H. Q., Li, G., Law, R., & Zhang, Y. (2019). Exploring tourist dining preferences based on restaurant reviews. *Journal of Travel Research*, 58(1), 149–167.
- Weihs, C., & Ickstadt, K. (2018). Data science: The impact of statistics. *International Journal of Data Science and Analytics*, 6(3), 189–194.
- Xiang, Z. (2018). From digitisation to the age of acceleration: On information technology and tourism. *Tourism Management Perspectives*, 25, 147–150.
- Xiang, Z., & Fesenmaier, D. R. (Eds.). (2017). *Tourism on the verge. Analytics in smart tourism design*. Springer International.
- Xu, J. B., & Wu, M. Y. (2018). Netnography as a new research method in tourism studies: A bibliometric analysis of journal articles (2006–2015). In *Handbook of research methods for tourism and hospitality management*. Edward Elgar.
- Yu, J., & Egger, R. (2021). Tourist experiences at overcrowded attractions: A text analytics approach. In *Information and communication technologies in tourism 2021* (pp. 231–243). Springer.
- Yu, C. E., Xie, S. Y., & Wen, J. (2020). Coloring the destination: The role of color psychology on Instagram. *Tourism Management*, 80, 104110.



# Data Science and Ethical Issues



## Between Knowledge Gain and Ethical Responsibility

Roman Egger, Larissa Neuburger, and Michelle Mattuzzi

### Learning Objectives

- Understand the theoretical intuition behind ethics
- Illustrate the spectrum of data science ethics
- Explain how ethics influence data science in tourism
- Discuss the pitfalls of not taking an ethical perspective into account

## 1 Introduction

Wherever data is used to predict and support decision-making processes, those decisions can affect people in many ways (Barocas & Selbst, 2016). Although the growing field of data science has brought many new possibilities for problem-solving and developing new insights based on data analysis (Saltz & Dewar, 2019), the topic of ethical challenges and the “appropriate” way of using data has only recently been starting to receive the attention it deserves. Since an overall

---

R. Egger (✉)

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

e-mail: [roman.egger@fh-salzburg.ac.at](mailto:roman.egger@fh-salzburg.ac.at)

L. Neuburger

Institute Tourism, Wine Business and Marketing, IMC University of Applied Sciences Krems, Krems, Austria

e-mail: [larissa.neuburger@fh-krems.ac.at](mailto:larissa.neuburger@fh-krems.ac.at)

M. Mattuzzi

Faculty of Arts, University of Groningen, Groningen, The Netherlands

compliance in regard to what is considered ethical vs. unethical seems to be lacking (Asadi-Someh et al., 2016), the field of data science requires a more thorough investigation. The idea of ethics involves not only human rights but also the rights of data derived from people as well as how to best handle this abundance of information for the greater good. By having a closer look at recent literature involving ethics within various sectors and branches of data science, this chapter aims at providing an overview of the ethical challenges that are currently being faced and discussed.

## 2 Ethics

Ethics can be defined as the science of morals or the moral evaluation of choices (LaFollette, 2007; Ulrich, 2008). Its most foundational interpretation “refers to the perception of something being good or right” (Saltz & Dewar, 2019, p. 198). In the context of data science, ethics illustrate the right, proper, acceptable, and socially appropriate approach to conducting research with this type of data.

Ethics can be divided into the different categories of meta-ethics, normative ethics, and applied ethics. While meta-ethics describe ethical theories, normative ethics focus on the process of reaching moral conclusions. Applied ethics are concerned with their practical application in certain contexts (Mingers & Walsham, 2010). Furthermore, ethics are based on three major ethical philosophies: the Kantian approach (Louden, 1986), the utilitarian point of view (Shaw, 1999), and the virtue model (Slote, 1992) of ethics. The Kantian approach involves honesty and responsibility and the belief that every ethical action’s foundation revolves around moral values, whereas the utilitarian approach looks for outcomes and consequences (LaFollette, 2007). The virtue model is not concerned with consequences or ethical actions but, rather, with subjective nonrational impulses (instincts) that influence how people act when clear rules are not present (Hursthouse, 1999; Merrill, 2011).

The Kantian approach (or deontology) is focused on the act itself instead of its consequences and results. “Actions are to be seen as morally right or wrong, just or unjust, in themselves regardless of their consequences” (Mingers & Walsham, 2010, p. 835). Hence, the outcome can never justify the means (in contrast to the utilitarian perspective). Kant’s philosophical approach is connected to duty and responsibility towards others and aims to universalize an ethical act for all humans. Nevertheless, the approach is criticized for being highly individually focused and limited in its universal application across different culture and belief systems (LaFollette, 2007; Mingers & Walsham, 2010).

The utilitarian standpoint, or, consequentialism, is based on choosing actions with the best overall outcomes while simultaneously minimizing any harm (origins: David Hume and Adam Smith). However, utilitarianism is limited by its power of predicting outcomes due to many unpredictable factors. Moreover, the approach is criticized for maximizing benefits for the majority while often marginalizing and

ignoring minorities and/or justifying unmoral means for the greater good (LaFollette, 2007; Mingers & Walsham, 2010; Saltz & Dewar, 2019).

Lastly, virtue ethics, as defined by Hursthouse (1999), is “agent-centered” rather than “act-centered” and emphasizes virtues and moral character in comparison to rules or duties (deontology) or consequences (utilitarianism). Although this viewpoint dates back to Plato and Aristotle, it has recently been revived; virtue ethics imposes the question of what sort of person one should/would like to be and is based on the fundamental concepts of ‘the good’ and the virtues of mind and character.

## 2.1 *Data Science and Ethics*

Ethics in data science can be viewed within the general context of computing and has been deliberated on ever since its development in the 1950s. A broader discourse was first initiated during the 1980s–1990s, leading to the introduction of the term “applied ethics.” As a result, computer ethics have been adopted into various curricula, textbooks, conferences, journals, and academic literature (see Stahl et al. (2016) or Brey and Soraker (2009) for more details) (Saltz & Dewar, 2019).

However, ethics in the field of data science have only recently been added to the ongoing debate. While ethics in general focus on humans’ decisions and choices, ethics in data science are more occupied with algorithm decisions (Mittelstadt et al., 2016). Algorithms and their parameters are specified by developers with certain results and outcomes in mind, thus prioritizing specific values over others (Kraemer et al., 2011; Nakamura, 2013). “At the same time, operation within accepted parameters does not guarantee ethically acceptable behaviour” (Mittelstadt et al., 2016, p. 1). Overall, good algorithms depend on good data and it is significant to note that algorithms have the same limitations as all data-processing methods. Algorithms, however, have often been found to be as biased as humans when it comes to minorities. Data can often inherit prejudices and reflect biases that exist as a whole in society, hereby making it challenging to identify the problem’s original source (Barocas & Selbst, 2016). Ethics in data science thus aim to provide guidance on how to interpret data and, as a consequence, what actions to implement (Mittelstadt et al., 2016).

In the context of current findings on ethical considerations in data science, Saltz and Dewar (2019) conducted a systematic literature review by inserting the search terms “data science and ethics,” “big data and ethics,” “data science and ethical,” and “big data and ethical” into six electronic databases. After applying the defined exclusion criteria, a total of 80 papers were reviewed by means of content analysis, and four key ethical themes were identified. First of all, findings revealed (1) the need for an ethics framework containing a consensus regarding terminology as well as a set of detailed regulations and policies (in contrast to a more general code of ethics). Furthermore, (2) the newness of the field concerning “ethical implications that have not been previously considered by others or even been highlighted as a potential ethical dilemma” (Saltz & Dewar, 2019, p. 202) was placed into the

spotlight. The third key theme revealed (3) data-related challenges, as for instance, privacy and anonymity issues, data misuse, and data accuracy and validity. Finally, (4) model-related challenges concerning personal and group harm, subjective model designs, and model misuse and interpretation issues encompassed the fourth key theme. These results present a bigger picture of the dispute surrounding data science and ethics. It is, however, significant to examine specific ethical issues further; thus, the following sections aim to address certain problems in a more detailed manner.

## 3 Data Science Ethics Issues

### 3.1 Privacy

Privacy rights serve as a frame for the amount of unstructured data available and the general protection of personal data and information (Schermer, 2013). In particular, privacy issues are concerned with the level of control users have in regard to their personal data, the ownership of data rights as well as the accessibility of data under which circumstances (Mateosian, 2013; Saltz & Dewar, 2019). In addition, privacy issues deal with the process of data collection, the use of the respective data, conclusions that derive from the data and actions that are taken as well as the consequences thereof (Birrer, 2005). However, while the costs of data protection have increased, the costs and barriers concerning privacy issues have decreased due to enhanced algorithms and larger datasets (Fairfield & Shtein, 2014).

Data privacy is one of the most critical issues, not only for data science but also for governments and lawmakers worldwide. For issues related to data protection and security, the following four areas can be classified (van den Hoven, 2008): (1) Protection against information-based damage, such as identity theft or fraud, due to the vulnerability of personal data in possession of parties wanting to impose harm. (2) Safeguard against informational inequality, focusing on the vulnerability of consumers themselves and their lack of knowledge or understanding that their personal data is accessible to businesses or governments (without any transparency about how these parties use this data). (3) Security involving informational injustice, which protects the use of consumers' data for causes and reasons outside the circumstances agreed upon (e.g., medical data that is accessed during the process of a job application). (4) Protection revolving around moral autonomy and moral identification concerning the rights of individuals to not be observed or controlled through their data and to create a certain distance between the outside world and the individuals themselves.

Generally, data science researchers mostly work with data that is readily available to them, and, oftentimes, they can derive and predict sensitive and personal data from already existing and accessible data (Kosinski et al., 2013; Mayer & Mutchler, 2014). A study by Mayer and Mutchler (2014) analyzed phone metadata and revealed that not only the users' identities but also their occupations, religious affiliations, or even medical conditions could be extracted. Another quite

controversial study, testing the effect of emotional contagion on social media, revealed how easy it is to access Facebook data and manipulate content without the user's awareness thereof (Kramer et al., 2014). In addition, Kosinski et al. (2013) showed that by analyzing users' Facebook likes, sensitive and personal attributes (e.g., sexual orientation, ethnicity, personality traits, age, and drug use) could be predicted. Hence, despite users' cautiousness of not revealing certain information about themselves, other data can be used to put these missing pieces together. While accessing and predicting personal data touches upon the issues of privacy, it is also significant to emphasize that the use and application of this data (e.g., for advertising purposes) can potentially result in unprecedented consequences on an individual level (e.g., advertisements and flyers for maternity products can reveal an unwanted pregnancy) (Kosinski et al., 2013).

These studies and examples demonstrate that individuals who leave traces of information online can easily be identified and categorized. In fact, big data is not necessarily required as connecting metadata from a few traces of online data is sufficient to program algorithms to derive even more information (Ananny, 2016). The issue of accessing information in such a simple manner can be compared to the problematic nature of surveillance. A "surveillance system obtains personal and group data in order to classify people and populations according to varying criteria, to determine who should be targeted for special treatment, suspicion, eligibility, inclusion, access, and so on" (Lyon, 2003, p. 20). In a bigger context, this "social sorting" of data can lead to discrimination and long-term social differences (Lyon, 2003).

While existing laws and regulations address the potential risks of protecting individual rights, they fail to focus on data protection, particularly when it comes to protecting groups of people from the impact of invasive data processing. Not only are users mostly unaware of the fact that certain data has been collected from them, but decision-makers often use big data analysis to make choices that can hugely impact either groups or individuals. In addition, these decisions are not based on data of an individual but, rather, on data that categorizes an individual as a member of a certain group. As a result, this process often leads to misrepresentation, discrimination, or bias. Hence, privacy and data protection are essential to safeguard personal rights and interests as well as to maintain the quality of society as a whole (Mantelero, 2016).

Whether or not privacy and data protection are regulated by the law, the issue of ethics remains. Moreover, regardless of certain data mining algorithms and data processes being legal, it does not necessarily mean they are ethical, especially when the respective individuals are not asked for permission (Custers, 2013). However, when talking about data in general, and data access in particular, there will always be a trade-off between privacy and security as well as control and freedom (Newell & Marabelli, 2015). "Ethics can only attempt to specify extreme boundaries of definitely unacceptable outcomes, and at the meta-level it can try to specify when the negotiation process is fair" (Birrer, 2005, p. 213).

## 3.2 *Data Validity*

Errors in data analysis may not only lead to a lack of validity but may also result in ethically problematic results with far-reaching consequences. As data form the foundation for decisions and indicate options for action, any errors that occur during the data collection, input, or processing steps can prompt results in the wrong direction (Lever et al., 2016). On the one hand, the results may appear incomprehensible or difficult to interpret, contributing to false conclusions. Worse still, they can have fatal consequences for the individual as well (Balas et al., 2015). For example, Amazon faced massive problems with an AI solution for their internal recruiting due to using a machine learning algorithm that was biased towards women. The historical training data, which served as input for the algorithm, was distorted by the male-dominated working environment of the technical world, thus discriminating against women (Vincent, 2018). Along similar lines, women were also widely discriminated against when it came to Apple's credit card. Even though both sexes were professionally equivalent, the credit line for men was set at up to 20 times the level of women (Vincent, 2019).

When it comes to issues in data science, they mainly occur in three simplified forms and may be caused by (1) a lack of validity of the data itself (Balas et al., 2015), (2) shortcomings in data processing (Kwon et al., 2014), or (3) a lack of validity concerning the created models (Raschka, 2018).

### 1. *Lack of validity of the data*

When working with publicly available data and using it as a basis for calculations, it is especially important that the data quality is sufficient (Gao et al., 2016). Often, however, quality checks and opportunities to gain a closer look into how the data was created are not possible (Gao et al., 2016). Another problem is the choice of a representative sample as researchers are often confronted with limitations and can only work with the data that is available to them (Seely-Gant & Frehill, 2015). In the introductory chapter on Natural Language Processing (see chapter "Natural Language Processing (NLP): An Introduction"), Twitter posts were used as sample data. Compared to Instagram or Facebook data, Twitter posts are easy to obtain and are, therefore, a frequently used source to analyze public opinions (Dindar & Yaman, 2018). Twitter users are typically younger, more technically savvy, and tend to have higher income, but they also vary greatly across countries (Dindar & Yaman, 2018; Seely-Gant & Frehill, 2015). In addition, an analysis of Twitter data revealed that only a small, very specific part of the population can be characterized as opinion leaders (Seely-Gant & Frehill, 2015). Hence, it should be questioned as to how representative this data is of the entire population and what conclusions can ultimately be made when relying on the analysis results of such data.

### 2. *Lack of data processing*

A lack of data processing often happens due to unfavorable decisions (Kwon et al., 2014), especially since there are numerous approaches that deal with missing values (Ngiam & Khor, 2019). Moreover, there are certain procedures

that react very sensitively to the presence of missing values and algorithms that can handle missing values well (Pratama et al., 2016). Hence, the question of whether missing data should simply be deleted, replaced with the mean or median, interpolated from adjacent data points, or simply be kept remains. Another example can be seen in the context of text analysis; in regard to dealing with irony and sarcasm or phrases that were automatically created by bots, the researcher must think about how to best handle text elements and must decide if the text can always be taken at face value (Potamias et al., 2020).

### 3. *Lack of validity of created models*

When it comes to the selection of features for machine learning (see chapter “Feature Engineering”), this process is of utmost importance (Cai et al., 2018). One must keep in mind which features are available in general and which ones should ultimately be selected in order to train a model. In machine learning, the quality of the training data itself also plays a decisive role (Raschka, 2018). If one trains a model using data from the past and the actual data used in the model is different from the original training data, the results will lead to skewed and misleading conclusions. Furthermore, customer behavior or economic data can change over time and lead to incorrect forecasts if the model does not adapt to these changes (Dergiades et al., 2018). For example, a prediction involving travel data from the last few years, especially from the time period during the COVID-19 pandemic, would massively distort the result in most cases. However, a badly trained algorithm that incorporates these outliers would be even worse than a one-time false result.

Besides the features, the selection of the appropriate model must also reflect an ethical perspective (Lever et al., 2016). The researcher should take into account if the model has been evaluated accordingly, if there is a possibility of over- or underfitting the data, and if the correct hyperparameters have been tuned (see chapter “Hyperparameter Tuning”). Which model should ultimately be chosen? In any case, it would be unethical to choose a model that has been rendered inaccurately but best fits the context’s goals (Raschka, 2018). Besides mistakes that can occur, be it from carelessness or ignorance, many subjective decisions, which can also lead to serious consequences, must be made by the researcher even during the structured processes of data science projects (Kitchin, 2014). It is therefore important to handle data and models responsibly, coupled with an awareness of the consequences of the decisions made.

### 3.3 *Algorithm Fairness and Bias*

Although data mining and other data science methods are designed to eliminate human bias, algorithms can only be as good as the data it was supplied with (Barocas & Selbst, 2014). “If data miners are not careful, the process can result in disproportionately adverse outcomes concentrated within historically disadvantaged groups in ways that look a lot like discrimination” (Barocas & Selbst, 2014, p. 673). Data

mining is highly dependent on the quality of the data; yet, often the data inherits prejudices and biases represented in society or performed in the past. In fact, algorithms can adopt these prejudices without even being programmed to do so. Hence, discrimination can either be a result of biased data or the data mining process itself. Such cases of unintentional discrimination are especially difficult to identify and mitigate (Barocas & Selbst, 2014).

While data mining algorithms can be used to detect spam or fraud (e.g., flag spam e-mail) as mostly uncontroversial binary categories, decisions about the creditworthiness or integrity of an employee, for example, cannot be uncovered so easily and depend largely on pre-defined categories and target variables specified by the data miner. Thus, the problem lies mostly in the definition of the (often subjective) class labels in addition to the possible inaccurate classification. In this way, various choices of defined class labels result in different consequences and impact certain classes (Barocas & Selbst, 2014; Gandy, 2010; Hildebrandt & Koops, 2010).

Another potential area of algorithm bias results from the actual training of the data that machine learning works with. Biased training data can lead to discriminatory models. Hence, in situations where biased cases are used as training data, the problem ends up repeating and reproducing itself. On the other hand, if the bias lies in the sample, the algorithm will repeat, eventually putting the underrepresented population at a disadvantage. Both cases can affect the training data, the algorithm, the results, and, in turn, its derived implications (Custers, 2013).

In addition, the training data and algorithm can adopt biases based on user or consumer behavior. Results from a study by Sweeney (2013) show that Google is more likely to display keyword-triggered ads of arrest records for search terms involving “black-sounding” names in comparison to “white-sounding” names. Furthermore, the study found that the reason for these results is Google’s algorithm to predict the likelihood that users will click on a certain advertisement based on historical user data. In this fashion, the algorithm learns (from the data) which ads receive the most hits and further promotes those accordingly, whereby the biased search results replicate themselves.

If the bias lies in the data itself, it can be a result of incorrect, incomplete, or non-representative datasets or data that has no records of certain classes of people, for instance, due to a lack of resources to collect this type of data (English, 2009; Tufekci, 2014; Wang & Strong, 1996). In particular, certain classes of people who live on the margins of big data may be underrepresented in the data or simply omitted completely (Lerman, 2013). “Not all data is created or even collected equally, there are ‘signal problems’ in big-data sets — dark zones or shadows where some citizens and communities are overlooked or underrepresented” (Crawford, 2013).



## 4 Big Data

When it comes to big data, it seems that the notion of “one-size-fits-all” must be discarded since the use of multidisciplinary big data affects both privacy and confidentiality and poses dilemmas in various ways, depending on the circumstances of a certain situation (Steinmann et al., 2016; O’Leary, 2016). This also leads to the question of whether big data and analytics should belong to the computer ethics category or if they should be treated as completely separate aspects (O’Leary, 2016). Although computer ethics has been a popular topic over the past decades and most experts are aware of these ethical guidelines (Saltz & Dewar, 2019), O’Leary (2016) argues that existing frameworks lack the specificity for certain technologies, thus limiting their effectiveness when applied to big data.

Furthermore, despite the availability of developed codes of conduct for big data projects, such as the IEEE Ethics and Member Conduct, the Data Science Code of Professional Conduct, or the Code of Ethics for Certified Analytics Professionals, to name but a few, the actual purpose for which the data is being used as well as the field in which the data scientists/analysts work lead to complications regarding which particular code to adhere to (O’Leary, 2016). Fairfield and Shtein (2014) additionally highlight the fact that big data techniques do not pair well with ethical approaches from the social sciences as they tend to focus strongly on individual human participants. Therefore, it seems that multiple policies or codes of ethics need to be designed explicitly for various big data techniques, technologies, and applications. In this way, researchers and organizations would also gain a clear-cut perspective and understanding of what kind of ethical behavior is expected and how the data can be handled (O’Leary, 2016).

According to Steinmann et al. (2016), big data analysis presents two main ethical concerns: the fact that big data tend to deal with populations rather than individual samples and that big data can be reused, repurposed, recombined, and reanalyzed (4R challenge). One speaks of reusing when the data is used for other purposes within the same domain in addition to what it was initially collected for (Steinmann et al., 2016). Based on the misconception that publicly available data does not impose any further harm, Metcalf and Crawford (2016) state that this ethical risk relating to individuals and communities is continuously overlooked in the field of big data. Here, it is significant to review the conditions of the initial dataset and to argue who, in such a case, is genuinely responsible for the various outputs (Leonelli, 2016).

On the other hand, repurposing involves taking the original data and using it for unrelated purposes (e.g., outside of the intended domain) in which the validity of the analysis as well as privacy and protection contexts both pose potential issues (Steinmann et al., 2016). An additional dilemma concerning a privacy breach also occurs in the case of combining and recombining data. Not only does recombining data potentially lead to uncovering an individual’s stripped identification, but the project’s underlying goal may even be to purposely involve the act of re-identifying a person. Lastly, reanalysis deals with big data archives that are stored for

longitudinal reasons, especially within the healthcare and public health sectors (Steinmann et al., 2016). All in all, the topic of 4R and privacy ultimately boils down to “where consent often amounts to an unread terms of service or a vague privacy policy” (Metcalf & Crawford, 2016, p. 2) and the management thereof.

Another pressing issue when it comes to big data is the topic of human subjects themselves. In Mittelstadt and Floridi’s (2016) meta-analysis, 11 key themes were revealed. These include, amongst others, informed consent, anonymity and data protection, ownership, and epistemology. Despite these topics being crucial, one downfall of strict data protection and distribution rights is potentially missing out on opportunities. Due to the prevention of sharing specific datasets, some researchers may end up not having access to the data they need and the information that would theoretically be appropriate and acceptable to use in the right context (Mittelstadt & Floridi, 2016; Choudhury et al., 2014). Nevertheless, it is argued that the exaggeration of the benefits of big data has masked the notion of taking ethical implications and considerations seriously, making it even more challenging to manage sensitive data. Additional future issues that have received little attention in the literature and require further investigation involve “group-level ethics, ethical implications of growing epistemological challenges [see chapter 2] [...], effects of Big Data on fiduciary relationships, the ethics of academic versus commercial practices, ownership of intellectual property derived from Big Data, and the content of and barriers to meaningful data access rights” (Mittelstadt & Floridi, 2016, pp. 468–469).

Educational research is another sector in which big data and ethics come together as a matter of contention. Daniel (2019) believes that the future application of big data in education will eventually cause complications regarding student safety and security within and across institutions. Major concerns embody themes such as maintaining research integrity and providing data to institutions without one’s permitted consent to share amongst third parties (Daniel, 2019). In this case, both national and international standards need to be set in order to address ethical issues in the field of education. As big data and ethics can be seen in many other sectors and subject matters (e.g., public health, healthcare, journalism, etc.) as well, it is crucial to consider the various branches in which data science is being incorporated and to start thinking about integrating methodological, societal, and ethical issues into an interdisciplinary approach (Delpierre & Kelly-Irving, 2018).

## 5 Artificial Intelligence and Machine Learning

As with so many technologies, the goal of Artificial Intelligence (AI) developers is to solve some of the complex issues of humanity and change the world for the better. Another vision is to overcome human subjectivity and let a system that is not influenced by emotions or personal biases judge with principles of fairness. However, the biggest challenge of AI is combining these principles with the nature of human need and instruction. Another aspect is that AI, similar to any other intelligent technology, can also be used by humans to harm other humans or even act

autonomously in destructive ways (Gabriel, 2020). Based on the three laws of robotics developed by Isaac Asimov in the 1940s/1950s, researchers and humanity as a whole must conquer the “new” challenge of adjusting and expanding these laws to the way AI is used today as well as the way AI will be used tomorrow and in the near future (Asimov, 1950).

The combination of big data and AI, along with the subfield of machine learning, accompany each other in numerous ways. In the public health sector, this can consist of medical screening, vision augmentation, and epidemiological and psychological matters, amongst others (Benke & Benke, 2018). Nonetheless, the question of ethical issues in the context of health is a big topic for big data to tackle. Benke and Benke (2018), for example, view genetic privacy and the balance between public rights and law enforcement to be a controversial subject. Moreover, they believe that algorithmic transparency (e.g., free of bias; explanatory rather than predictive) should be part of an ethical standard. Brady and Neri (2020) agree, stating that the knowledge of deep learning and the way algorithms are developed and used can lead to ethical concerns or abuses. This is especially significant when algorithms are applied to risk detection and prediction. Therefore, algorithms should be tested intensively in order to be able to withstand potential biases and false negatives (Linthicum et al., 2018). Yet, the question then remains: who is responsible for the future lives that are lost?

The large number of publications on the situation of COVID-19 in the field of tourism is also currently bringing more health-specific issues into perspective and giving them new relevance. In this regard, “ethical issues arise in terms of ownership of data, how data are used, and how the privacy of those from whom the data is derived is protected” (Brady & Neri, 2020, p. 232). Here, the importance of anonymizing data arises as patient re-identification can bring about unwanted advertising or even lead to medical records being brought forward to the public (Brady & Neri, 2020). Other ethical issues include topics such as resource inequality, liability, conflicts of interest, and workforce disruption. Overall, a framework for AI, not only within the health and medical sectors, should be deemed necessary in order to protect human rights, foster user safety, discuss the roles of future diagnosticians and medical specialists, and raise awareness of the risks of AI tools (Benke & Benke, 2018; Brady & Neri, 2020).

In addition, AI faces two principal challenges: technical and normative. The technical challenge focuses on ensuring the reliability of artificial agents to fulfill the tasks they are expected to achieve (Gabriel, 2020). Normative challenges, on the other hand, concentrate on the alignment of human values and principles to avoid unsafe and unreliable outcomes (Gabriel, 2020). Nevertheless, the possibility of imitation-learning of an AI system from a moral expert reveals the deeper underlying problem concerning ethics and brings up questions if moral experts even exist in this framework as well as who can be called a moral expert and by whom (Gabriel, 2020; MacIntyre, 2013; McDowell, 1979; Vallor, 2016). In addition, Gabriel (2020, p. 6) raises the following questions: “From what data should AI extract its conception of values, and how should this be decided? Should this data include everyone’s behaviour, or should it exclude the behaviour of those who are manifestly unethical

(sociopathic) or unreasonable (fundamentalists)? Finally, what criteria should be used for determining which agent is the ‘most moral’, and is it possible to rank entities in this way?” Different frameworks have already been developed to define such principles and rules for a more responsible and ethically designed AI (see Future of Life Institute, 2017; Montreal Declaration, 2017; IEEE, 2017; European Group on Ethics in Science and New Technologies, 2018; Floridi et al., 2018; Partnership on AI, 2018). However, more work must be done in order to establish a universal consensus regarding a consistent framework of the ethical development and usage of AI.

As society becomes more and more dependent on technology, and when it comes to AI and machine learning in other domains, the biggest concerns remain in regard to data privacy and data security. For instance, Mageswaran et al. (2018) emphasize the need to scrutinize data that derive from personal apps and highlight the establishment of ethical algorithms in the business field. The use of AI tools could not only lead to technologies deliberately intruding on people’s lives but also to revealing or preventing a crime; therefore, coinciding with law and ethics policies is of utmost importance (Soroka & Kurkova, 2019). Durante (2019), in addition, says that society needs “to deal with the social (ethical, legal, economic, and political) impact of the delegation of decisions [in regard] to automated systems and autonomous artificial agents” (p. 372). Based on what is already known, the best way to face ethical dilemmas in light of data science is to review every situation thoroughly and question whether or not the data and ethics have been used appropriately within the established context (Saltz & Dewar, 2019).

## 6 Conclusion

In order to gain knowledge and subsequently derive recommendations for actions and decisions, the use of structured and unstructured data with the aid of data science methods and procedures has increased significantly both in academic research and in business-related situations. Tourism can be seen as a social phenomenon, which is why the analysis of people and their behavior is often put in the foreground. Thus, tourism-specific research and data are usually embedded in a social context and therefore linked to complex and sensitive ethical issues. However, such issues are much broader than the often superficially discussed topics of privacy, security, identity, trust, responsibility, and ownership (Longbing, 2019).

In addition to the data itself, the processing and analysis, along with the use of suitable algorithms and the development of models, right up to the interpretation and utilization of the results can all give rise to massive ethical pitfalls, especially since the application of data science approaches has only recently started to gain importance in the tourism industry. While certain sectors, such as online travel agencies (OTAs), are strategically aligning themselves with data science, destinations and the hospitality sector are only slowly starting to address this topic. As in any other industry, there is a sense of optimism and excitement about the new opportunities to

make more and more data-driven decisions. However, at the beginning, the understanding as well as the correct assessment of the relevance of ethical aspects often remain on the back burner. Only when new achievements have been consolidated and established is there time to optimize processes and take issues such as ethics into account. Theoretically, though, this process should be the other way around, and this applies to academic research as well. It seems as if the new possibilities of gaining knowledge often lead to a rash decision of incorporating methods without thinking twice about the ethical requirements, applications, and/or consequences beforehand. Therefore, one can only hope that a solution to this problem will be recognized and enforced at all levels and that individuals become aware of the fact that ignoring ethical aspects corresponds to a short-sighted view of endangering not only others but also oneself.

## References

- Ananny, M. (2016). Toward an ethics of algorithms: Convening, observation, probability, and timeliness. *Science, Technology, and Human Values*, 41(1), 93–117.
- Asadi-Someh, I., Breidbach, C. F., Davern, M. J., & Shanks, G. G. (Eds.). (2016). *Ethical implications of big data analytics*. In 24th European Conference on Information Systems, ECIS 2016. Association for Information Systems.
- Asimov, I. (1950). *I, Robot*. Gnome Press.
- Balas, E. A., Vernon, M., Magrabi, F., Gordon, L. T., & Sexton, J. (2015). Big data clinical research: Validity, ethics, and regulation. *Studies in Health Technology and Informatics*, 216, 448–452.
- Barocas, S., & Selbst, A. (2016). Big data's disparate impact. *California Law Review*, 104, 671.
- Benke, K., & Benke, G. (2018). Artificial intelligence and big data in public health. *International Journal of Environmental Research and Public Health*, 15(12), 2796.
- Birrer, F. A. (2005). Data mining to combat terrorism and the roots of privacy concerns. *Ethics and Information Technology*, 7(4), 211–220.
- Brady, A. P., & Neri, E. (2020). Artificial intelligence in radiology—Ethical considerations. *Diagnostics*, 10(4), 231.
- Brey, P., & Soraker, J. (2009). Philosophy of computing and information technology. In D. M. Gabbay, A. W. M. Meijers, J. Woods, & P. Thagard (Eds.), *Philosophy of technology and engineering sciences* (pp. 1341–1408). Elsevier.
- Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70–79.
- Choudhury, S., Fishman, J. R., McGowan, M. L., & Juengst, E. T. (2014). Big data, open science and the brain: Lessons learned from genomics. *Frontiers in Human Neuroscience*, 8, 239.
- Crawford, K. (2013, May 10). Think again: Big data. Why the rise of machines isn't all it's cracked up to be. *Foreign Policy*. <https://foreignpolicy.com/2013/05/10/think-again-big-data/>
- Custers, B. (2013). Data dilemmas in the information society: Introduction and overview. In B. Custers, T. Calders, B. Schermer, & T. Zarsky (Eds.), *Discrimination and privacy in the information society* (pp. 3–26). Springer.
- Daniel, B. K. (2019). Big data and data science: A critical review of issues for educational research. *British Journal of Educational Technology*, 50(1), 101–113.
- Delpierre, C., & Kelly-Irving, M. (2018). Big data and the study of social inequalities in health: Expectations and Issues. *Frontiers in Public Health*, 6(312), 1–5.

- Dergiades, T., Mavragani, E., & Pan, B. (2018). Google Trends and tourists' arrivals: Emerging biases and proposed corrections. *Tourism Management*, *66*, 108–120.
- Dindar, M., & Yaman, N. D. (2018). #UseTwitterBecause: Content analytic study of a trending topic in Twitter. *Information Technology and People*, *31*(1), 256–277.
- Durante, M. (2019). Safety and security in the digital age. Trust, algorithms, standards, and risks. In D. Berkich & M. V. d'Alfonso (Eds.), *On the cognitive, ethical, and scientific dimensions of artificial intelligence* (pp. 371–383). Springer.
- English, L. P. (2009). *Information quality applied: Best practices for improving business information, processes and systems*. Wiley.
- European Group on Ethics in Science and New Technologies. (2018). *Statement on artificial intelligence, robotics and 'autonomous' systems*. [https://ec.europa.eu/info/news/ethics-artificial-intelligence-statement-ege-released-2018-apr-24\\_en](https://ec.europa.eu/info/news/ethics-artificial-intelligence-statement-ege-released-2018-apr-24_en)
- Fairfield, J., & Shtein, H. (2014). Big data, big problems: Emerging issues in the ethics of data science and journalism. *Journal of Mass Media Ethics*, *29*, 38–51.
- Florida, L., Cows, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F., Schafer, B., Valcke, P., & Vayena, E. (2018). AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. *Minds and Machines*, *28*(4), 689–707.
- Future of Life Institute. (2017). *Asilomar AI Principles*. <https://futureoflife.org/ai-principles>
- Gabriel, I. (2020). Artificial intelligence, values, and alignment. *Minds and Machines*, *30*, 411–437.
- Gandy, O. H. (2010). Engaging rational discrimination: Exploring reasons for placing regulatory constraints on decision support systems. *Ethics and Information Technology*, *12*(1), 29–42.
- Gao, J., Xie, C., & Tao, C. (Eds.). (2016). *Big data validation and quality assurance—Issues, Challenges, and Needs*. In 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE). IEEE.
- Hildebrandt, M., & Koops, B. J. (2010). The challenges of ambient law and legal protection in the profiling era. *The Modern Law Review*, *73*(3), 428–460.
- Hursthouse, R. (1999). *On virtue ethics*. Oxford University Press.
- IEEE Advancing Technology for Humanity. (2017). *Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems*. [https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead\\_v2.pdf](https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead_v2.pdf)
- Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data and Society*, *1*(1), 2053951714528481.
- Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(15), 5802–5805.
- Kraemer, F., van Overveld, K., & Peterson, M. (2011). Is there an ethics of algorithms? *Ethics and Information Technology*, *13*(3), 251–260.
- Kramer, A. D., Guillory, J. E., & Hancock, J. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences of the United States of America*, *111*(24), 8788–8790.
- Kwon, O., Lee, N., & Shin, B. (2014). Data quality management, data usage experience and acquisition intention of big data analytics. *International Journal of Information Management*, *34*(3), 387–394.
- LaFollette, H. (2007). *Ethics in practice*. Blackwell.
- Leonelli, S. (2016). Locating ethics in data science: Responsibility and accountability in global and distributed knowledge production systems. *Philosophical Transactions A*, *374*, 2083.
- Lerman, J. (2013). Big data and its exclusions. *Stanford Law Review*, *66*, 55–57.
- Lever, J., Krzywinski, M., & Altman, N. (2016). Points of significance: Model selection and overfitting. *Nature Methods*, *13*(9), 703–704.
- Linthicum, K. P., Schafer, K. M., & Ribeiro, J. D. (2018). Machine learning in suicide science: Applications and ethics. *Behavior Science Law*, *37*, 214–222.

- Longbing, C. (2019). *Data science thinking: The next scientific, technological and economic revolution*. Springer.
- Louden, R. B. (1986). Kant's virtue ethics. *Philosophy*, 61(238), 473–489.
- Lyon, D. (Ed.). (2003). *Surveillance as social sorting: Privacy, risk, and digital discrimination*. Psychology Press.
- MacIntyre, A. (2013). *After virtue*. A&C Black.
- Mageswaran, G., Nagappan, S. D., Hamzah, N., & Brohi, S. N. (Eds.). (2018). *Machine learning: An ethical, social & political perspective*. In 2018 Fourth International Conference on Advances in Computing, Communication and Automation (ICACCA). IEEE.
- Mantelero, A. (2016). Personal data for decisional purposes in the age of analytics: From an individual to a collective dimension of data protection. *Computer Law and Security Review*, 32(2), 238–255.
- Mateosian, R. (2013). Ethics of big data. *IEEE Micro*, 33(2), 60–61.
- Mayer, J., & Mutchler, P. (2014, March 12). MetaPhone: The sensitivity of telephone metadata. *Web Policy*. <http://webpolicy.org/2014/03/12/metaphone-the-sensitivity-of-telephone-metadata/>
- McDowell, J. (1979). Virtue and reason. *The Monist*, 62(3), 331–350.
- Merrill, J. C. (2011). Theoretical foundations for media ethics. In A. D. Gordon, J. M. Kittross, J. C. C. Merrill, W. Babcock, & M. Dorsher (Eds.), *Controversies in media ethics* (pp. 3–32). Routledge.
- Metcalf, J., & Crawford, K. (2016). Where are human subjects in Big Data research? The emerging ethics divide. *Big Data and Society*, 3(1), 1–14.
- Mingers, J., & Walsham, G. (2010). Toward ethical information systems: The contribution of discourse ethics. *Mis Quarterly*, 34(4), 833–854.
- Mittelstadt, B. D., & Floridi, L. (2016). The ethics of big data: Current and foreseeable issues in biomedical contexts. In B. D. Mittelstadt & L. Floridi (Eds.), *The ethics of biomedical big data* (pp. 445–480). Springer.
- Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data and Society*, 3(2), 2053951716679679.
- Montreal Declaration for a Responsible Development of Artificial Intelligence. (2017). *The declaration*. <https://www.montrealdeclaration-responsibleai.com/the-declaration>
- Nakamura, L. (2013). *Cybertypes: Race, ethnicity, and identity on the internet*. Routledge.
- Newell, S., & Marabelli, M. (2015). Strategic opportunities (and challenges) of algorithmic decision-making: A call for action on the long-term societal effects of 'datification'. *The Journal of Strategic Information Systems*, 24(1), 3–14.
- Ngiam, K. Y., & Khor, W. (2019). Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology*, 20(5), e262–e273.
- O'Leary, D. (2016). Ethics for big data and analytics. *IEEE Intelligent Systems*, 31(4), 81–84.
- Partnership on AI. (2018). *Tenets*. <https://www.partnershiponai.org/tenets/>
- Potamias, R. A., Siolas, G., & Stafylopatis, A. G. (2020). A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32, 17309–17320.
- Pratama, I., Permasari, A. E., Ardiyanto, I., & Indrayani, R. (Eds.). (2016). *A review of missing values handling methods on time-series data*. In 2016 International Conference on Information Technology Systems and Innovation (ICITSI). IEEE.
- Raschka, S. (2018, November 13). *Model evaluation, model selection, and algorithm selection in machine learning*. Cornell University. <https://arxiv.org/abs/1811.12808>
- Saltz, J. S., & Dewar, N. (2019). Data science ethical considerations: A systematic literature review and proposed project framework. *Ethics and Information Technology*, 21, 197–208.
- Schermer, B. (2013). Risks of profiling and the limits of data protection law. In B. Custers, T. Calders, B. Schermer, & T. Zarsky (Eds.), *Discrimination and privacy in the information society* (pp. 137–152). Springer.
- Seely-Gant, K., & Frehill, L. M. (2015). Exploring bias and error in big data research. *Journal of the Washington Academy of Sciences*, 101(3), 29–38.

- Shaw, W. (1999). *Contemporary ethics: Taking account of utilitarianism*. Blackwell.
- Slote, M. (1992). *From morality to virtue*. Oxford University Press.
- Soroka, L., & Kurkova, K. (2019). Artificial intelligence and space technologies: Legal, ethical and technological issues. *Advanced Space Law*, 3, 131–139.
- Stahl, B. C., Timmermans, J., & Mittelstadt, B. D. (2016). The ethics of computing: A survey of the computing-oriented literature. *ACM Computing Surveys (CSUR)*, 48(4), 1–38.
- Steinmann, M., Matei, S. A., & Collmann, J. (2016). A theoretical framework for ethical reflection in big data research. In J. Collmann & S. A. Matei (Eds.), *Ethical reasoning in big data: An exploratory analysis* (pp. 11–27). Springer.
- Sweeney, L. (2013). Discrimination in online ad delivery: Google ads, black names and white names, racial discrimination, and click advertising. *Queue*, 11(3), 10–29.
- Tufekci, Z. (Ed.). (2014). *Big questions for social media big data: Representativeness, validity and other methodological pitfalls*. In ICWSM '14: Proceedings of the 8th International AAAI Conference on Weblogs and Social Media. arXiv.
- Ulrich, P. (2008). *Integrative economic ethics: Foundations of a civilized market economy*. Cambridge University Press.
- Vallor, S. (2016). *Technology and the virtues: A Philosophical guide to a future worth wanting*. Oxford University Press.
- van den Hoven, J. (2008). Information technology, privacy, and the protection of personal data. In J. van den Hoven & J. Weckert (Eds.), *Information technology and moral philosophy* (pp. 301–321). Cambridge University Press.
- Vincent, J. (2018, October 10). Amazon reportedly scraps internal AI recruiting tool that was biased against women. *The Verge*. <https://www.theverge.com/2018/10/10/17958784/ai-recruiting-tool-bias-amazon-report>
- Vincent, J. (2019, November 11). Apple's credit card is being investigated for discriminating against women. *The Verge*. <https://www.theverge.com/2019/11/11/20958953/apple-credit-card-gender-discrimination-algorithms-black-box-investigation>
- Wang, R. Y., & Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4), 5–33.



# Web Scraping



## Collecting and Retrieving Data from the Web

Roman Egger, Markus Kroner, and Andreas Stöckl

### Learning Objectives

- Discuss types of data in tourism
- Show what tools and packages are available for web scraping
- Explain the legal aspects of web crawling
- Illustrate how to crawl and parse a website with Python

## 1 Introduction and Theoretical Foundations

Arguably, the most far-reaching changes in recent decades are owed to the digitalization of our society, which has also massively altered the tourism industry. As early as 1955, the German philosopher Karl Jaspers stated that “it is impossible to overestimate the impact of modern technology and its consequences on all aspects of life” (Jaspers, 1955). While Marshall (1962) already saw the harbingers of a global information society in the 1960s, Nora and Minc (1980) also spoke of the “computerization of society” in 1978 (Egger, 2015). At the moment, the popularity of social

---

R. Egger (✉)

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

e-mail: [roman.egger@fh-salzburg.ac.at](mailto:roman.egger@fh-salzburg.ac.at)

M. Kroner

Kroner Rechtsanwalts GmbH, Salzburg, Austria

e-mail: [office@legalcounsel.at](mailto:office@legalcounsel.at)

A. Stöckl

School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Wels, Austria

e-mail: [andreas.stoeckl@fh-hagenberg.at](mailto:andreas.stoeckl@fh-hagenberg.at)

media platforms, in particular, is leading to changes in social structures (Fuchs, 2021), and, as a result, rapidly growing datasets require new methodological approaches to investigate them.

“Big Data is the new oil” is a frequently quoted slogan, and indeed this analogy seems to be confirmed on several levels. Although data is renewable, in contrast to fossil fuels, Thorp (2012) mentions the danger of “data pollution” in addition to data’s enormous economic and commercial importance. Corbett (2018) argues that, in the case of fossil fuels, the long-term effects, the costs incurred, and the dependency and associated problems of steering away from them have not been considered, and the same also seems to hold true for data. Thus, caution is advised as to how we generate, process, and use data (for further information, see chapter “Data Science & Ethics”). In line with this, new methods and approaches of data analysis are viewed as disruptive innovations creating epistemological challenges. In the course of the USA’s fight against terrorism in 2005, the then new director of the National Security Agency (NSA), General Keith Alexander, once said, “collect it all.” What he meant by this corresponded to a paradigm shift; it was no longer a matter of looking for a needle in a haystack but of collecting the whole haystack as a first step. As such, he demanded, “collect it all, tag it, store it. . . . And whatever it is you want, you go searching for it.” Currently, there are huge discussions about a new form of empiricism that is reconfiguring the way we do research (Egger & Yu, 2022). Yet, before some prematurely proclaim the “end of theory,” a wide-ranging critical reflection on the epistemological implications of the data revolution is needed (Kitchin, 2014) (see chapter “Epistemological Challenges”).

Our society is dependent on the generation, collection, processing, and use of data, and, for each of us, data has become a central element in our everyday lives (Johnson, 2014). While we, as recipients, try to at least select data deliberately, be it by researching on the web, browsing social media channels, or obtaining multimedia content, we are, at the same time, unconsciously producing a multitude of data. We leave digital footprints based on our online research, generating metadata, such as GPS coordinates and dates, when we upload photos or draw conclusions about our preferences by liking and commenting on social media posts. Besides that, sensors, wearables, and the internet of things (IoT) are also collecting and sharing data automatically (Gretzel et al., 2015a, 2015b; Li et al., 2018; Xiang & Fesenmaier, 2017a).

Tourism has always been an information-intensive industry and is currently transforming into a data-driven industry. The amount and variety of available data nowadays allow for unimaginable possibilities of blending and combining data, resulting in new extracted features that can be fed into machine learning models to reveal patterns that were previously unknown. Therefore, data is collected, combined, and analyzed to better understand customers, competitors, and other stakeholders as well as develop products, analyze and optimize processes, forecast market developments, and much more (Xiang & Fesenmaier, 2017b).

Fundamentally, a distinction between static and dynamic data can be made. In tourism, static data includes editorial texts such as descriptions of destinations and attractions or any marketing media from tourist providers. Usually, such data can be retrieved from its respective websites or the platforms of intermediaries. This data rarely changes and is available in all media formats, ranging from textual descriptions to photos and images and audio and video files, usually existing in a very unstructured and fragmented way. In most cases, in order to obtain this data for further processing and analysis, it must be crawled first. Dynamic data, on the other hand, changes frequently and includes such data as prices and availability, which, as a rule of thumb, can only be obtained by means of web scraping. Another type of dynamic data involves data that has already been pre-processed and prepared for further use, for example, information on tourism markets or statistical indicators (e.g., the number of arrivals or overnight stays) made available by organizations or government institutions either in the form of open data or via application programming interfaces (APIs). With the popularity of social media channels, user-generated content (UGC) has emerged as a distinct data type that can be both static or dynamic (Pantano et al., 2017). Depending on the social media channel, the content tends to be text-, image-, or video-heavy, and the platforms regulate access to this data. While Twitter data can be accessed relatively easily via APIs, Facebook, Instagram, or TripAdvisor data is hardly accessible via APIs so as to ensure the protection of users' privacy.

Regardless of the data type, metadata, describing the original data in more detail, is almost always present. For example, a hotel can be described in more detail based on its geo-coordinates and addresses as well as opening hours and star rating, while price information can be valid for certain time periods or only valid for certain customer groups. When it comes to blog posts, these are tagged with meta tags such as the author's name, the date of creation, and, possibly, an assigned category. Social media posts, on the other hand, are often geotagged, and one can see when the post was created, who the author is, and who commented or liked it. This metadata often allows datasets to be linked and, thus, new, interesting, and information-rich features can be extracted. Since metadata describes other data further, it is almost always extracted alongside the base data. Figure 1 shows a general overview of the different types of data used in tourism cases.

At one point, Li et al. (2018) analyzed 165 articles that used big data (although it is questionable whether all the data was derived from big data or simply from larger datasets) in the context of tourism research and also examined where this data came from. As can be seen in Fig. 2, the results show that UGC (text and photos) accounts for almost half of the data used in tourism big data analysis. However, data collected from end devices, especially GPS data, also makes up a significant portion (36%). Only 17% of the data can be attributed to transaction data.

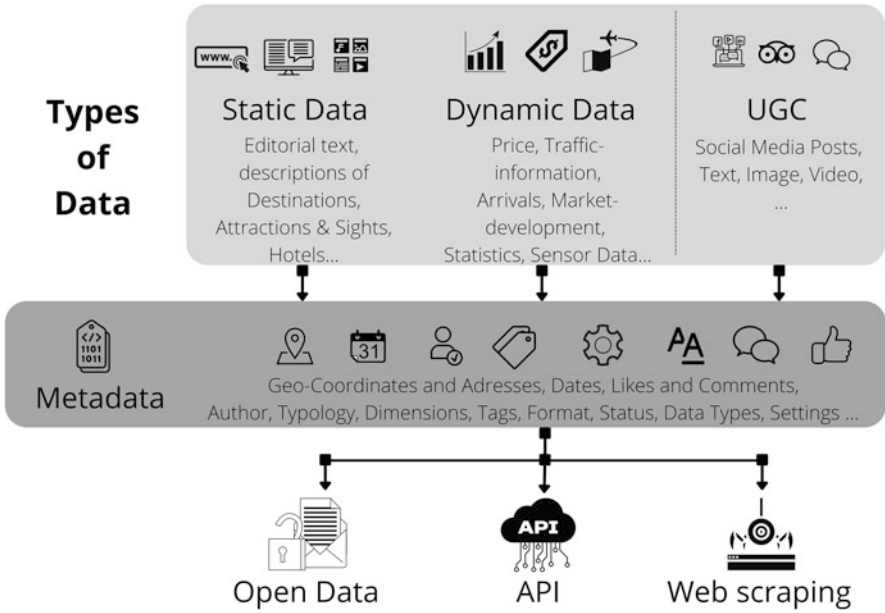


Fig. 1 Types of data in tourism

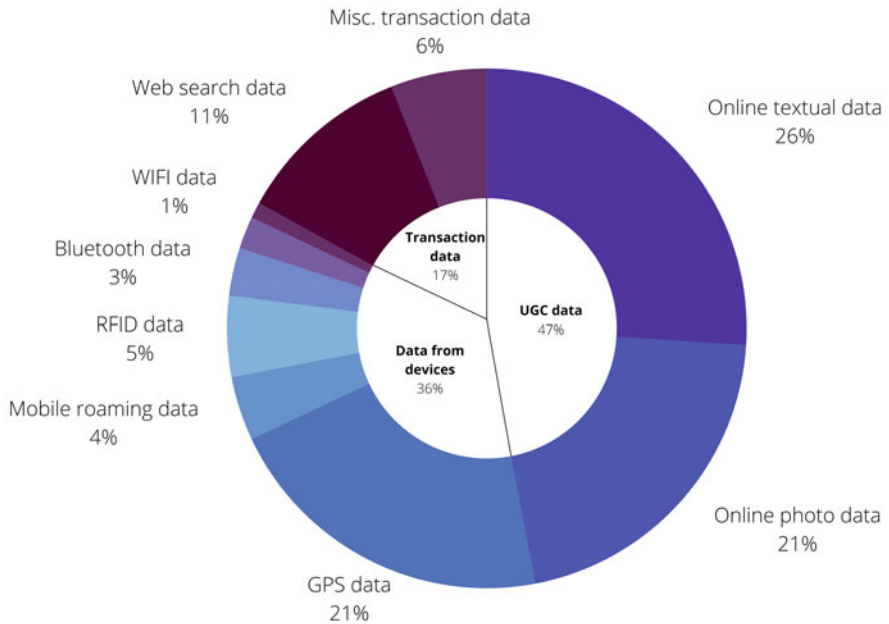


Fig. 2 Percentages of big data usage in tourism based on Li et al. (2018)

## 1.1 Open Data

Today, we have more data than ever at our fingertips. The trend towards open data continues to accelerate this notion, and in tourism, as well, more and more organizations, authorities, and governments are making data publicly available (Pesonen & Lampi, 2016). Longhi, Titz, and Viallis (2014) along with Pesonen and Lampi (2016) note that the tourism industry has taken on a pioneering role in providing and using open data, which includes traffic information, spatial data, event data, statistics and key figures, and information about culture and points of interest, to name but a few. Open data is freely available to everyone and made accessible via the Internet, without being subject to restrictive conditions of use (Fermoso et al., 2015; Longhi et al., 2014). Thus, its overall intention is to drive innovation (Xiang & Fesenmaier, 2017b), improve public welfare, and use public resources more efficiently (Johnson et al., 2012). As tourism is a very complex and interconnected industry, which places high demands on communication and coordination between all stakeholders, open data can contribute to this by making data available for tourism and hospitality research, e.g., to understand tourists needs and optimize the quality of their experiences (Pantano et al., 2019). As such, the availability and use of open data have been intensively discussed and advocated, particularly in the context of smart cities or smart destinations (Gretzel et al., 2015a, 2015b).

To cater to this, platforms operated by governments, destinations, or authorities are starting to emerge around the world in order to make tourism data publicly available for use. Examples in Europe include <https://data.europa.eu/en/highlights/open-data-tourism>, <https://info.datatourisme.gouv.fr/> from the French government or <https://opendata.swiss/de/group/tourism> from Switzerland. Another example is the Austria Experience Data Hub, which, together with the tourism industry, is trying to define data standards and make relevant data from third-party providers, such as mobility, weather, or telecom data, available in a bundled and uncomplicated way. In this way, data can be integrated into the company's own ecosystem, and innovative services can be developed, for instance, via startups. However, tourism studies also use datasets provided on platforms, such as Kaggle, data provided by companies for further analysis, such as the academic Yelp dataset (Guerreiro & Rita, 2020), or initiatives like Inside Airbnb, an independent and non-commercial website that crawls Airbnb data and makes it publicly available. For example, Yu et al. (2020) used data from Inside Airbnb to compare Airbnb listing's amenities with hotels, and Tussyadiah and Park (2018) analyzed hosts' description and trust on Airbnb, while Güçlü, Roche, and Marimon (2020) investigated the characteristics of cities in Europe using this data. At the same time, however, Alsudais (2021), by investigating the accuracy of Inside Airbnb datasets, cautions against the use of such datasets. Additionally, Marsden and Pingry (2018) warn of major numerical data quality problems in the unreflective use of publicly provided data.

## 1.2 APIs

Another way to retrieve data is through the use of Application Programming Interfaces (APIs), which many companies provide in order to connect to their system. By doing so, the shared data can also be used by other developers to design additional services, ultimately increasing the attractiveness of the original system. For many APIs, you need to submit a request, along with a description, of which service you need the data for, and the companies then decide individually whether or not an approval will be granted. For research projects, it is therefore quite challenging to obtain access, especially when it comes to social media channels. Although they offer their data via APIs, recently, massive restrictions have been applied to protect the rights of their users. For example, while it is still relatively easy to obtain tweets from Twitter or photos from Flickr via an API (Ainin et al., 2020), Instagram and Facebook no longer permit the collection of user data (Zhang et al., 2021). Alternatively, on Github, for instance, unofficial APIs can sometimes be programmed and made available (Golenvaux et al., 2020).

## 1.3 Scraping Data

Finally, there is also data that is publicly available on the Internet but has not been formatted as a download or cannot be obtained via an API. Typically, this content needs to be scraped from websites or social media channels, and, because they are available for display in browsers, this involves software to retrieve and store the web pages. As a result of such a page retrieval, the text from the HTML source is converted into a plain text file. However, a problem with obtaining data in this way is that, in addition to the data that is actually required, the file often contains code used for the display or navigation of the page. The data must, therefore, still be cleaned and cleared of the HTML code.

A closer look at the literature on tourism reveals that data from TripAdvisor, especially, is crawled quite frequently in order to analyze tourism products or tourist behavior. For example, Taecharungroj and Mathayomchan (2019) used TripAdvisor data to analyze tourist attractions in Phuket, while Khorsand, Rafiee, and Kayvanfar (2020) provided insights into Tehran's hotels, and Banerjee and Chua (2016) analyzed travelers' rating patterns in online hotel reviews. Thereby, different scraping approaches are used to obtain the required data. Chang, Ku, and Chen (2020), for example, programmed a crawler in C# based on Selenium to investigate deep learning and visual analytics of hotel reviews and responses. Furthermore, An, Ma, Du, Xiang, and Fan (2020) developed a Python crawler to analyze user-generated photos in online hotel reviews, and Ganzaroli, Noni, and van Baalen (2017) used import.io to investigate the influence of TripAdvisor on the quality of restaurants in Venice.

As such, one can see that there are a multitude of ways to crawl data from websites. On the one hand, various programming or scripting languages can be used, with Python and R considered the most relevant, while, at the same time, numerous tools and software solutions that provide a graphical user interface to develop an individual web crawler also exist. Usually, a workflow in which one defines how the crawler should navigate through the page and which data should be stored is created, and, thereafter, a CSV or Excel file with the crawled data can be downloaded. *Octoparse*, as used in Yu and Egger's (2021) study, *Prowebscraper*, or *Scrapestorm* are just a few of the many available solutions.

## 1.4 Legal Perspectives of Text and Data Mining

The legal perspective of crawling data on the Internet is a frequent subject of debate as it is all too easy to fall into a tricky legal situation, especially considering that international legal circumstances vary widely. In the following section, we will discuss the legal situation for the European Union in more detail.

The European Union (EU) introduced a legal framework for Text and Data Mining (TDM) in the Directive EU 2019/790 of April 17, 2019, on copyright and related rights in the Digital Single Market (DSM Directive), which had to be transposed into national law by June 7, 2021, by the Member States. Moreover, the European Commission had stated in the Commission Staff Working Document of September 14, 2016, that the fragmentation of the Single Market is likely to increase as a result of Member States adopting TDM exceptions at the national level (e.g., Section 60d of the German Act on Copyright and Related Rights). This could be based on different conditions, likely to happen in the absence of intervention at EU level. Generally speaking, in Article 2 (2), the DSM Directive provides a definition of TDM as any automated analytical technique aimed at analyzing text and data in digital form in order to generate information that includes, but is not limited to, patterns, trends, and correlations.

Up until the new legal framework of the DSM Directive was introduced, TDM had been complying with the regulations laid down in the InfoSoc Directive (Directive 2001/29/EC of May 22, 2001, on the harmonization of certain aspects of copyright and related rights in the information society) and the Database Directive (Directive 96/9/EC of March 11, 1996, on the legal protection of databases). Under these Directives, the right of reproduction, with the exception of temporary reduction, as well as the extraction of all or substantial parts of a database (the so-called "sui generis right") had to be authorized by the right holders. Since, from a technical point of view, TDM can involve both a reproduction of protected materials and the extraction of substantial parts of a database, the DSM Directive thus aims to grant TDM exceptions for reasons of legal certainty. For the sake of clarity, Recital 9 and 18 point out that the mandatory exception for temporary acts of reproduction in the InfoSoc Directive should continue to apply to TDM techniques as long as any process of copying/duplicating for reasons beyond the scope of the granted

exception does not take place. As such, the DSM Directive grants TDM an exception in regard to reproductions and extractions made by research organizations and cultural heritage institutions in order to carry out, for the purposes of scientific research, TDM on works or other subject matter to which lawful access is acknowledged (Article 3). According to Article 2 (1), research organization means, i.a., a university, a research institute, or any other entity with its primary goal being to conduct scientific research or be part of an educational activity, organized as either a non-profit entity or an entity with a public interest mission recognized by a Member State. As a result, commercial research institutes do not fall within the scope of this exception, and any contractual provisions relating to the exceptions provided in this Article are unenforceable (Article 7 (1)).

Outside the scope of this mandatory exception and the existing exception for temporary acts of reproduction written in the InfoSoc Directive, right holders should still be able to continue licensing the uses of their works (Recital 18). Therefore, in Article 4 (1), the DSM Directive permits, in general, the reproduction and extraction of lawfully accessible works for the purpose of TDM, although the terms of the right holders' licenses can exclude TDM under appropriate circumstances, such as machine-readable content being made publicly available online (Article 4 (3)). According to Recital 18, it can also be appropriate to reserve such rights through other means such as contractual agreements or a unilateral declaration. As a result, pointed out correctly by Hugenholtz in the Kluwer Copyright Blog ([The New Copyright Directive: Text and Data Mining \(Articles 3 and 4\)—Kluwer Copyright Blog \(kluweriplaw.com\)](#)), the DSM Directive effectively creates and legitimizes a derivative market for text and data mining in which right holders can control, license, or even entirely prohibit the process.

Since personal data, e.g., name, pictures, IP addresses, e-mail addresses, etc., may be wholly or partly processed by automated means during the course of TDM, the provisions of the General Data Protection Regulation (Regulation (EU) 2016/679 of April 27, 2016, on the protection of natural persons with regard to the processing of personal data and on the free movement of such data) also apply. Regarding the legality of processing personal data, public research organizations recognized by a Member State can rely on Article 6 (1.e) of the GDPR, which allows the processing of personal data for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller. On the other hand, privately organized non-commercial research organizations can justify this processing by means of other legitimate interests, which, in the case of publicly available data, override the interests, fundamental rights, and freedoms of the data subject (s) (Article 6 (1.f) GDPR).

A frequently cited example from outside Europe is that of HiQ vs. LinkedIn. The data science company HiQ scraped data from public profiles of LinkedIn users without logging in, leading them straight to court (Woollacott, 2019); however, there are also contrasting examples. As the international legal situation can vary greatly across countries, Ng (2019) recommends to carefully review and respect the Terms of Service, contact the site owner, and check the Robots.txt before starting to scrape any data. She also exhorts all researchers to carry out the scraping process



conservatively and to not stress or crash servers so as to ensure that that the scraped data is kept safe, is not unmistakably and indiscriminately given to third parties, and/or is not duplicated.

## 1.5 *Typical Use Cases of Web Scraping in Tourism*

A very typical application of web scraping in the tourism sector is to fetch data from portals for booking and rating holiday offers that are normally not accessible via APIs, such as the TripAdvisor portal mentioned above. These usually involve descriptions of offers and price information provided by the portal operators as well as user-generated content, such as reviews. As such, the data can be used not only to analyze and compare the quality of holiday offers but, naturally, also to monitor prices, which, in turn, allow general trends in the industry to be observed and compared across different regions. One special use case, in particular, includes the collection of airfare data from airline websites and online flight booking portals. Here, a unique challenge lies in the dynamic pricing of this segment, therefore rendering crawling a necessity not only at frequent times but also with different end devices and from different geographical locations. However, in addition to applications based on global data, other use cases that operate in a more localized area, such as the evaluation of restaurant menus, also exist and are equally important.

Before data can be used and analyzed by means of an appropriate method, for example, regression or classification (see the corresponding chapters) for predictions, they usually have to undergo certain pre-processing steps. Besides numerical data (e.g., prices), the data at hand could also potentially be in the form of texts, which, however, often contain parts that are not needed for the analysis. Searching for and extracting the desired text passages using text analysis functions, such as regular expressions, is usually a complex and time-consuming task that is also prone to errors. Therefore, specialized packages including “BeautifulSoup,”<sup>1</sup> “Selenium,”<sup>2</sup> or “Scrapy”<sup>3</sup> provide a remedy here.

## 1.6 *BeautifulSoup*

BeautifulSoup is a Python library that reads data from HTML and XML files. It offers countless possibilities for navigating, searching, and changing parse trees, thus usually saving hours, or even days, of work. By addressing a specific HTML element (using the CSS class name), the program package offers functions that allow

---

<sup>1</sup>BeautifulSoup documentation: <https://www.crummy.com/software/BeautifulSoup/bs4/doc>

<sup>2</sup><https://www.selenium.dev/>

<sup>3</sup><https://scrapy.org/>

```

<html lang="en">
<head>...</head>
<body class="rebrand_2017 js_logging desktop_web Hotels" id="BODY_BLOCK_QUERY_REFLOW" data-new-gr-c-s-check-loaded="14.1036.0" data-gr-ext-installed">
  <div class="header global-header">
    <div class="masthead">
      <!-- PLACEMENT horizon_ad -->
      <div id="tapic_horizon_ad_0" class="ppr_rup ppr_priv_horizon_ad" data-placement-name="horizon_ad">...</div>
      <!-- PLACEMENT global_nav -->
      <div id="tapic_global_nav_0" class="ppr_rup ppr_priv_global_nav" data-placement-name="global_nav">...</div>
      <!-- PLACEMENT masthead_search_empty -->
      <div id="tapic_masthead_search_empty_0" class="ppr_rup ppr_priv_masthead_search" data-placement-name="masthead_search:empty">...</div>
      <!-- PLACEMENT boost_native_ads -->
      <div id="tapic_boost_native_ads_0" class="ppr_rup ppr_priv_boost_native_ads" data-placement-name="boost_native_ads">...</div>
    </div>
  </div>
  <div class="page">...</div>
  <div id="singletonRoot">...</div>
  <div id="mcrx-pinner-root">...</div>
  <div id="mcrx-drawer-root">...</div>
  <div class="cBtam Za f e" style="position: absolute; inset: 266px 0px auto; z-index: 9991;">...</div>
  <form method="post" action="//www.tangrt.com/RT?target=ta80372890036321183" style="display: none;">...</form>
  <iframe id="Nonikes" style="border:none; position:absolute; bottom:0; right:0" width="1" height="1" src="//PageMoniker?pixel=facebook_dat_mv_img_n_mv_pixel_facebook_dat_mv_img_n-92086_3398927?boisellType=PAGEVIEW&oid=470522c8-adde-d4c3-884d-cde83fe86655" scrolling="no">...</iframe>
  
  <iframe src="https://www.google.com/recaptcha/api2/iframe" width="0" height="0" style="display: none;">...</iframe>
  <iframe id="Nonikes_HPL" style="border:none; position:absolute; bottom:0; right:0" width="1" height="1" src="//PageMoniker?kserv=let=Hotels&userInQue=75a006d-nt=0&boisellType=HAC_PRICES_LOAD&oid=470522c8-adde-d4c3-884d-cde83fe86655" scrolling="no">...</iframe>
  <div id="fb-root" class="fb_reset">...</div>
</body>
<!-- grammarly-desktop-integration data-grammarly-shadow-root="true">...</grammarly-desktop-integration>
</html>

```

Fig. 3 HTML page tree from [Tripadvisor.com](https://www.tripadvisor.com)

parts of the text to be found and also has functions for extracting the content or attributes of the HTML element. It can be used to find either individual elements or also passages that meet certain criteria. To be able to move around the document, one not only has the possibility to traverse it linearly but can also navigate along the edges of the tree that is spanned by the HTML page elements. For example, all hyperlinks that are present in a particular paragraph of the document can be extracted, first by navigating to the desired paragraph and then by searching for the attributes of the `<a>`-tags that occur there.

The HTML or XML document’s page tree can, however, not only be used for navigation, but it can also be changed itself (Fig. 3). For instance, parts of the document can be removed in a structured way by deleting a subtree and/or nodes can be inserted at a very specific point in the document.

Although there are a variety of other tools available (see below), as part of the practical demonstration later, we will use BeautifulSoup to extract TripAdvisor reviews.

## 1.7 Selenium

Selenium is a web browser automation software that is often used for automated testing but can also be applied for web scraping. Generally speaking, it is always used when calling up and evaluating pages by means of requests is not sufficient. If actions in the browser, such as mouse clicks, are necessary to access the information, then controlling the browser with tools such as Selenium is useful. Furthermore, an additional plugin in Chrome and Firefox can record and playback user interactions with the browser, which, in turn, can be used to create simple scripts. In addition to

the actual scraping tool, Selenium also offers a development environment for creating crawling applications as well as a platform that allows the crawlers to run on a network of servers in order to meet even high-performance requirements.

## 1.8 Scrapy

Scrapy is a Python program package that can be easily incorporated with other software developed with Python. In addition to this download, extensive documentation is also offered. The Scrapy library not only allows you to crawl HTML pages and extract parts of them but, as a framework, additionally offers the possibility to write complete web crawling applications that read the contents and follow the links contained in them in order to make their contents available. Moreover, it also takes care of the processing and storing of the data in databases.

## 2 Practical Demonstration

Let us assume that you want to analyze online hotel reviews and, since no API is provided, the data needs to be crawled from an online rating platform. In this example, the user feedback should be extracted from [www.TripAdvisor.com](http://www.TripAdvisor.com); the full Jupyter notebook, including markdowns and all calculations and graphics, is available on the book’s Github profile.

Figure 4 shows a typical review written by a customer, and, as such, we want to extract the text of the review and store it locally. For this demonstration, Python is

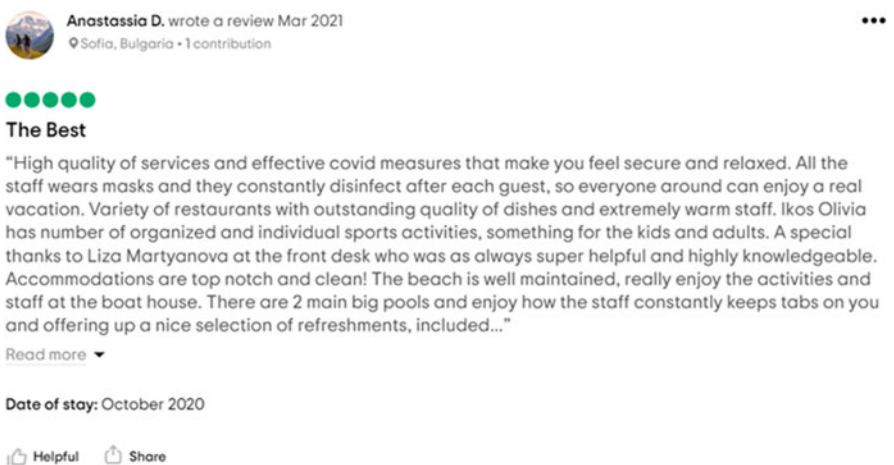


Fig. 4 A typical review from [www.TripAdvisor.com](http://www.TripAdvisor.com)

```

b'<DOCTYPE html><html lang="de-AT" xmlns:og="http://opengraphprotocol.org/schema/"><head><meta http-equiv="content-type" content="text/html; charset=utf-8"/><link rel="icon" id="favicon" href="https://static.tacdn.com/favicon.ico?v2" type="image/x-icon"/><link rel="mask-icon" sizes="any" href="https://static.tacdn.com/img2/brand/refresh/application_icons/mask-icon.svg" color="#000000" /><meta name="theme-color" content="#34e0a1" /><meta name="format-detection" content="telephone=no" /><script type="text/javascript">window.taRollupsAreAsync = true;</script><link rel="stylesheet" href="https://static.tacdn.com/css2/webfonts/tripsans/tripsans.css?v1.002" crossorigin><title>IKOS OLIVIA: Bewertungen, Fotos & Preisvergleich (Gerakini, Griechenland) - Tripadvisor</title><meta property="al:ios:app_name" content="TripAdvisor"><meta property="al:ios:app_store_id" content="284876795"><meta property="twitter:app:id:ipad" name="twitter:app:id:ipad" content="284876795"><meta property="twitter:app:id:iphone" name="twitter:app:id:iphone" content="284876795"><meta property="al:ios:url" content="tripadvisor://www.tripadvisor.at/Hotel_Review-g651973-d6978275-Reviews-Ikos_Olivia-Gerakini_Halkidiki_Region_Central_Macedonia.html?m=33762"><meta property="twitter:app:url:ipad" name="twitter:app:url:ipad" content="tripadvisor://www.tripadvisor.at/Hotel_Review-g651973-d6978275-Reviews-Ikos_Olivia-Gerakini_Halkidiki_Region_Central_Macedonia.html?m=33762"><meta property="twitter:app:url:iphone" name="twitter:app:url:iphone" content="tripadvisor://www.tripadvisor.at/Hotel_Review-g651973-d6978275-Reviews-Ikos_Olivia-Gerakini_Halkidiki_Region_Central_Macedonia.html?m=33762"><meta name="description" content="Hotel Ikos Olivia, Gerakini: 3.413 Bewertungen, 4.204 authentische Reisefotos und g\xc3\xbcnstige Angebote f\xc3\xbc Hotel Ikos Olivia. Bei Tripadvisor auf Platz 1 von 12 Hotels in Gerakini mit 5/5 von Reisenden bewertet."/><meta property="og:title" content="IKOS OLIVIA: Bewertungen, Fotos & Preisvergleich (Gerakini, Griechenland) - Tripadvisor"/><meta property="og:description" content="Hotel Ikos Olivia, Gerakini: 3.413 Bewertungen, 4.204 authentische Reisefotos und g\xc3\xbcnstige Angebote f\xc3\xbc Hotel Ikos Olivia. Bei Tripadvisor auf Platz 1 von 12 Hotels in Gerakini mit 5/5 von Reisenden bewertet."/><meta property="og:image" content="https://

```

Fig. 5 HTML code of the webpage

used as the scripting language, and the package BeautifulSoup is the module used to perform this task.

As a first step, we import BeautifulSoup “bs4” using pip and then import the module together with a package to be able to make calls from web pages. When calling up the corresponding page with the ratings, we receive a so-called “request object.” With this object, we can then try to read the page. As a result, either the content of the page is delivered and can be displayed or a suitable error message appears.

The following code snippet below shows this process, while Fig. 5 exemplifies a successful call of a page from TripAdvisor.<sup>4</sup>

```

url = https://www.TripAdvisor.at/...
req = urllib.request.Request(url, None)
try:
    html = urllib.request.urlopen(req).read()
    print(html)
except urllib.error.URLError as e:
    print(e.reason)

```

As one can observe, the result is a very confusing text file with special characters and numerous HTML tags and scripts. It is difficult to find the ratings in this file, and it is also time-consuming to extract them; this is where BeautifulSoup comes in, turning the text file into a structured data object that is easy to search. The following code shows this call and prints the result.

```

soup = BeautifulSoup(html, 'html.parser')
print(soup)

```

Although the output of the object does not appear very different yet, internally, everything is ready to work with. Next, in the source code of the page, we look for the HTML tag and the CSS class name that enclose the text of the ratings. In this

<sup>4</sup>More about requesting URLs at: <https://docs.python.org/3/howto/urllib2.html>

```
[<div class="cPQsENeY" style="max-height:242px;line-break:normal;cursor:auto"><div><p>Das Ikos Olivia ist eine ausgezeichnete Wahl, wenn Sie Gerakini besuchen möchten. Die Unterkunft bietet ein familienfreundliches Umfeld mit vielen Annehmlichkeiten für Reisende und überzeugt außerdem durch die ideale Kombination aus Preis-Leistung, Komfort und Bequemlichkeit.</p><p>Zimmer im Ikos Olivia bieten Flachbildfernseher, Klimaanlage und Minibar und Gäste können mit dem kostenlosen WLAN in Kontakt bleiben.</p><p>Darüber hinaus können die Gäste während ihres Aufenthalts im Ikos Olivia den Concierge und den Zimmerservice in Anspruch nehmen. Während Ihres Aufenthalts im Ikos Olivia können Sie gerne den Pool und das Frühstück besuchen. Benötigen Sie einen Parkplatz? Am Ikos Olivia ist ein kostenloser Parkplatz verfügbar.</p><p>Wenn Sie nach einem griechischen Restaurant suchen, probieren Sie doch Anemilos Restaurant, Elia oder 4 Epohes, die alle nicht weit vom Ikos Olivia entfernt sind.</p><p>Komfort und Zufriedenheit der Gäste stehen im Ikos Olivia an erster Stelle und die Unterkunft freut sich, Sie in Gerakini begrüßen zu dürfen.</p></div></div>, <div class="cPQsENeY" style="max-height:160px;line-break:normal;cursor:auto"><q class="IrsGk0Pm"><span>Das Ikos Olivia würde ich immer wieder gerne weiter empfehlen.Trotz momentaner angespannter Coronasituation fühlten wir uns sehr sicher und gut aufgehoben.Das Personal von der Rezeption über alle Sparten war top.Zu erwähnen ist auch eine super Küche in allen Restaurants.Das einzige was uns fehlte war ein Buffet, was aber den Umständen entsprechend verständlich war.Sollten wir wieder einmal in Griechenland unsere Ferien verbringen ist das Olivia sicher eines der ersten Adressen.</span></q></div>, <div class="cPQsENeY" style="max-height:160px;line-break:normal;cursor:auto"><q class="IrsGk0Pm"><span>Das Ikos Olivia würde ich immer wieder gerne weiter empfehlen.Trotz momentaner angespannter Coronasituation fühlten wir uns sehr sicher und gut aufgehoben.Das Personal von der Rezeption über alle Sparten war top.Zu erwähnen ist auch eine super Küche in allen Restaurants.Das einzige was uns fehlte war ein Buffet, was aber den Umständen entsprechend verständlich war.Sollten wir wieder einmal in Griechenland unsere Ferien verbringen ist das Olivia sicher eines der ersten Adressen.</span></q></div>
```

Fig. 6 List of reviews from the website

```
Das Ikos Olivia ist eine ausgezeichnete Wahl, wenn Sie Gerakini besuchen möchten. Die Unterkunft bietet ein familienfreundliches Umfeld mit vielen Annehmlichkeiten für Reisende und überzeugt außerdem durch die ideale Kombination aus Preis-Leistung, Komfort und Bequemlichkeit.Zimmer im Ikos Olivia bieten Flachbildfernseher, Klimaanlage und Minibar und Gäste können mit dem kostenlosen WLAN in Kontakt bleiben.Darüber hinaus können die Gäste während ihres Aufenthalts im Ikos Olivia den Concierge und den Zimmerservice in Anspruch nehmen. Während Ihres Aufenthalts im Ikos Olivia können Sie gerne den Pool und das Frühstück besuchen. Benötigen Sie einen Parkplatz? Am Ikos Olivia ist ein kostenloser Parkplatz verfügbar.Wenn Sie nach einem griechischen Restaurant suchen, probieren Sie doch Anemilos Restaurant, Elia oder 4 Epohes, die alle nicht weit vom Ikos Olivia entfernt sind.Komfort und Zufriedenheit der Gäste stehen im Ikos Olivia an erster Stelle und die Unterkunft freut sich, Sie in Gerakini begrüßen zu dürfen.
```

Fig. 7 Cleaned review

example, it is a div tag called “cPQsENeY.” This manual search for the correct tags may be quite a nuisance when confronted with large pages such as this one. Moreover, one should also keep in mind that this query is a fragile construct, and any structural change to the page, such as a change to the page template, will result in a script that no longer works/runs. Unfortunately, these two disadvantages are part of the process and have to be accepted when working with web scraping.

With the help of the function “find\_all” for the “Soup” object in the code below, we can now obtain a list of all these elements, as shown in Fig. 6.

```
reviews = soup.find_all("div", class_="cPQsENeY")
print(reviews)
```

The individual elements from the list each contain a rating text for the hotel, but other HTML tags also seem to be included. In order to get solely the pure text, we apply the “get\_text” method to each element to receive the rating texts as output (Fig. 7).

```
for i in reviews:
    print(i.get_text())
```

The rating texts for the hotel are now available and can be stored, for example, in a database and used for analyses. For such purposes, sentiment analysis is particularly useful to determine the tone of the texts.

If you wish to read several pages or read all pages with a certain URL structure, you call them up one after the other and extract the results. With multiple pages, however, this can become confusing, which is where other tools such as “Scrapy” can come into play.

### Service Section

**Main Application Fields:** Any area where data that is not provided in a structured form as downloads or APIs is collected and extracted from websites.

Applications include, for example:

- Analysis of customer ratings
- Comparing price information
- Collecting hotel or restaurant descriptions
- Preparing flight times or airfare information in a structured form

**Limitations and Pitfalls:** Legal framework must be taken into consideration as not everything that is technically possible is also legally covered.

Programs developed using web crawler software typically only work in the context for which they were developed and are hardly transferable to other situations. For example, special elements from HTML pages that are only present on the one page or the underlying page template are often addressed. This means that the crawling process is also very sensitive to changes in the pages involved. For instance, changing a CSS class name can cause the data extraction process to no longer work. It is therefore crucial to monitor and control the processes throughout.

**Similar Methods and Methods to Combine with:** Crawling methods usually involve data retrieval and pre-processing in order to prep the data for various analysis methods (presented in the other chapters) and can therefore be combined with (more or less) all of them, depending on the application situation.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-5-Web-Mining-Data-Crawling>

## Further Readings and Other Sources

### *Blogposts*

BeautifulSoup—<https://codeburst.io/web-scraping-101-with-python-beautiful-soup-bb617be1f486>

Scrapy—<https://towardsdatascience.com/using-scrapy-to-build-your-own-dataset-64ea2d7d4673>

Selenium—<https://medium.com/free-code-camp/better-web-scraping-in-python-with-selenium-beautiful-soup-and-pandas-d6390592e251>

## References

- Ainin, S., Feizollah, A., Anuar, N. B., & Abdullah, N. A. (2020). Sentiment analyses of multilingual tweets on halal tourism. *Tourism Management Perspectives*, 34, 100658. <https://doi.org/10.1016/j.tmp.2020.100658>
- Alsudais, A. (2021). Incorrect data in the widely used Inside Airbnb dataset. *Decision Support Systems*, 141, 113,453. <https://doi.org/10.1016/j.dss.2020.113453>
- An, Q., Ma, Y., Du, Q., Xiang, Z., & Fan, W. (2020). Role of user-generated photos in online hotel reviews: An analytical approach. *Journal of Hospitality and Tourism Management*, 45, 633–640. <https://doi.org/10.1016/j.jhtm.2020.11.002>
- Banerjee, S., & Chua, A. Y. (2016). In search of patterns among travellers' hotel ratings in TripAdvisor. *Tourism Management*, 53, 125–131. <https://doi.org/10.1016/j.tourman.2015.09.020>
- Chang, Y.-C., Ku, C.-H., & Chen, C.-H. (2020). Using deep learning and visual analytics to explore hotel reviews and responses. *Tourism Management*, 80, 104129. <https://doi.org/10.1016/j.tourman.2020.104129>
- Corbett, C. J. (2018). How sustainable is big data? *Production and Operations Management*, 27(9), 1685–1695. <https://doi.org/10.1111/poms.12837>
- Egger, R. (2015). Die Welt wird phygital: Metamorphosen touristischer Räume. In R. Egger & K. Luger (Eds.), *Tourismus und mobile Freizeit: Lebensformen, Trends, Herausforderungen* (pp. 27–46). Books on Demand.
- Egger, R., & Yu, C.-E. (2022). Epistemological challenges. In R. Egger (Ed.), *Tourism on the verge. Applied data science in Tourism: Interdisciplinary approaches, methodologies and applications* (pp. 17–34). Springer.
- Fermoso, A. M., Mateos, M., Beato, M. E., & Berjón, R. (2015). Open linked data and mobile devices as e-tourism tools. A practical approach to collaborative e-learning. *Computers in Human Behavior*, 51, 618–626. <https://doi.org/10.1016/j.chb.2015.02.032>
- Fuchs, C. (2021). *Social media: A critical introduction* (3rd ed.). Sage.
- Ganzaroli, A., de Noni, I., & van Baalen, P. (2017). Vicious advice: Analyzing the impact of TripAdvisor on the quality of restaurants as part of the cultural heritage of Venice. *Tourism Management*, 61, 501–510. <https://doi.org/10.1016/j.tourman.2017.03.019>
- Golenvaux, N., Alvarez, P. G., Kioussou, H. S., & Schaus, P. (2020). *An LSTM approach to Forecast Migration using Google Trends*.
- Gretzel, U., Reino, S., Kopera, S., & Koo, C. (2015a). Smart tourism challenges. *Journal of Tourism*, 41–47.
- Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015b). Smart tourism: Foundations and developments. *Electronic Markets*, 25(3), 179–188. <https://doi.org/10.1007/s12525-015-0196-8>
- Güçlü, B., Roche, D., & Marimon, F. (2020). City characteristics that attract Airbnb Travellers: Evidence from Europe. 1800–6450. *International Journal for Quality Research*, 14(1), 271–290. <https://doi.org/10.24874/IJQR14.01-17>
- Guerreiro, J., & Rita, P. (2020). How to predict explicit recommendations in online reviews using text mining and sentiment analysis. *Journal of Hospitality and Tourism Management*, 43, 269–272. <https://doi.org/10.1016/j.jhtm.2019.07.001>
- Jaspers, K. (1955). *Vom Ursprung und Ziel der Geschichte*. Kösel.
- Johnson, J. A. (2014). From open data to information justice. *Ethics and Information Technology*, 16(4), 263–274.
- Johnson, P. A., Sieber, R. E., Magnien, N., & Ariwi, J. (2012). Automated web harvesting to collect and analyse user-generated content for tourism. *Current Issues in Tourism*, 15(3), 293–299. <https://doi.org/10.1080/13683500.2011.555528>
- Khorsand, R., Rafiee, M., & Kayvanfar, V. (2020). Insights into TripAdvisor's online reviews: The case of Tehran's hotels. *Tourism Management Perspectives*, 34, 100673. <https://doi.org/10.1016/j.tmp.2020.100673>

- Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. *Big Data and Society*, 1(1), 205395171452848. <https://doi.org/10.1177/2053951714528481>
- Li, J., Xu, L., Tang, L., Wang, S., & Li, L. (2018). Big data in tourism research: A literature review. *Tourism Management*, 68, 301–323. <https://doi.org/10.1016/j.tourman.2018.03.009>
- Longhi, C., Titz, J. B., & Viallis, L. (2014). Open data: Challenges and opportunities for the tourism industry. In E. Kitanovska-Ristoska (Ed.), *Tourism management, marketing, and development* (pp. 15–40).
- Marsden, J. R., & Pingry, D. E. (2018). Numerical data quality in IS research and the implications for replication. *Decision Support Systems*, 115, A1–A7. <https://doi.org/10.1016/j.dss.2018.10.007>
- Marshall, M. (1962). *The Gutenberg Galaxy: The making of typographic man*. University of Toronto Press. <https://doi.org/10.4324/9780203992968-14>
- Ng, A. (2019, July 18). Is web crawling legal? Towards Data Science. *Towards Data Science*. Retrieved from <https://towardsdatascience.com/is-web-crawling-legal-a758c8fcacde>
- Nora, S., & Minc, A. (1980). Computerizing society. *Society*, 17(2), 25–30. <https://doi.org/10.1007/bf02700056>
- Pantano, E., Priporas, C.-V., & Stylos, N. (2017). ‘You will like it!’ using open data to predict tourists’ response to a tourist attraction. *Tourism Management*, 60, 430–438. <https://doi.org/10.1016/j.tourman.2016.12.020>
- Pantano, E., Priporas, C.-V., Stylos, N., & Dennis, C. (2019). Facilitating tourists’ decision making through open data analyses: A novel recommender system. *Tourism Management Perspectives*, 31, 323–331. <https://doi.org/10.1016/j.tmp.2019.06.003>
- Pesonen, J., & Lampi, M. (2016). Utilizing open data in tourism. In A. Inversini & R. Schegg (Chairs), *ENTER 2016 conference on information and communication technologies in tourism*.
- Taecharungroj, V., & Mathayomchan, B. (2019). Analysing TripAdvisor reviews of tourist attractions in Phuket, Thailand. *Tourism Management*, 75, 550–568. <https://doi.org/10.1016/j.tourman.2019.06.020>
- Thorp, J. (2012). *Big data is not the new oil*. Retrieved from <https://hbr.org/2012/11/data-humans-and-the-new-oil>
- Tussyadiah, I. P., & Park, S. (2018). When guests trust hosts for their words: Host description and trust in sharing economy. *Tourism Management*, 67, 261–272. <https://doi.org/10.1016/j.tourman.2018.02.002>
- Woollacott, E. (2019, October 9). LinkedIn data scraping ruled legal. *Forbes*. Retrieved from <https://www.forbes.com/sites/emmawoollacott/2019/09/10/linkedin-data-scraping-ruled-legal/?sh=3e9917a31b54>
- Xiang, Z., & Fesenmaier, D. R. (2017a). Analytics in tourism design. In Z. Xiang & D. R. Fesenmaier (Eds.), *Analytics in smart tourism design* (pp. 1–12). Springer International.
- Xiang, Z., & Fesenmaier, D. R. (2017b). Big data analytics, tourism design and smart tourism. In *Analytics in smart tourism design* (pp. 299–307). Springer. [https://doi.org/10.1007/978-3-319-44263-1\\_17](https://doi.org/10.1007/978-3-319-44263-1_17)
- Yu, J., & Egger, R. (2021). Color and engagement in touristic Instagram pictures: A machine learning approach. *Annals of Tourism Research*, 103, 204. <https://doi.org/10.1016/j.annals.2021.103204>
- Yu, M., Cheng, M., Yu, Z., Tan, J., & Li, Z. (2020). Investigating Airbnb listings’ amenities relative to hotels. *Current Issues in Tourism*, 1–18. <https://doi.org/10.1080/13683500.2020.1733497>
- Zhang, H., van Berkel, D., Howe, P. D., Miller, Z. D., & Smith, J. W. (2021). Using social media to measure and map visitation to public lands in Utah. *Applied Geography*, 128, 102389. <https://doi.org/10.1016/j.apgeog.2021.102389>



# **Part II**

# **Machine Learning**

# Machine Learning in Tourism: A Brief Overview



## Generation of Knowledge from Experience

Roman Egger

### Learning Objectives

- Illustrate the intuition behind machine learning
- Explain the different ML paradigms
- Provide an overview of ML algorithms
- Discuss areas of application in the field of tourism
- Understand the limitations and challenges of ML

## 1 Introduction and Theoretical Foundations

Machine Learning (ML) is undoubtedly one of the most significant and far-reaching technological developments to currently shape our times (Jamal et al., 2018). These technologies can be found in all areas of our lives, providing us with information and knowledge derived from data, albeit in an inconspicuous way. They serve as the backbone of voice assistants, such as Siri, Cortana, Bixby, or Alexa, are the cornerstones of chatbots, support personalized marketing, and predict customer behavior. They optimize processes of all kinds, filter and classify spam from our emails, form the basis of fraud prevention, and also provide the foundation for plagiarism checks. Additionally, ML technologies are constantly being employed in the use of social media platforms without users even noticing or being aware of it. For example, ML is used when photos are uploaded to Facebook and faces are automatically recognized and suggested for tagging or when sentiments behind

---

R. Egger (✉)

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

e-mail: [roman.egger@fh-salzburg.ac.at](mailto:roman.egger@fh-salzburg.ac.at)

emojis are recognized on Instagram and auto-hashtags are suggested. ML provides the central infrastructure for Artificial Intelligence (AI) and often lays the foundation for data science projects, linking computer science and statistics in order to develop algorithms and statistical model-based theories (Althbiti & Ma, 2020) with the objective of high predictive performance and generalizability (Jordan & Mitchell, 2015).

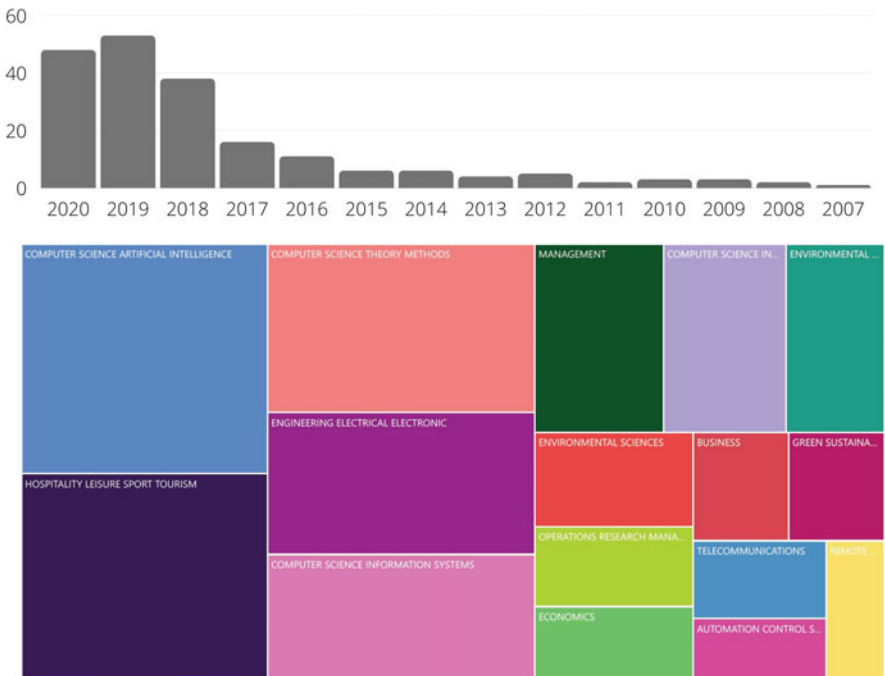
Although the beginnings of machine learning date back to the 1950s, its potential has only recently become apparent as the amount of data available today has reached enormous proportions. ML is often mentioned in the context of “Big Data” (Kelm et al., 2020), where it is true that its influence increases in parallel with the amount of data available. At the same time, however, it must be mentioned—and this is particularly relevant for scientific research projects—that one does not always have to be in possession of several gigabytes of data in order to be able to successfully apply ML. As will be exemplified later on, even smaller datasets containing several hundreds or thousands of instances might be sufficient enough to use ML approaches to automatically identify patterns in complex data. The knowledge of these patterns can then help to predict future events and make complex decisions with confidence.

There have been numerous attempts to define the term *machine learning*, like the one from Arthur Samuel, who first coined the term “Machine Learning” in 1959, describing it as “a field of study that gives computers the ability to learn without being explicitly programmed” (the quote is often cited, but cannot be found in his papers. It can therefore be seen as a gist of Arthur Samuel’s 1959 paper). Similarly, Akerkar (2019b) defines ML as “computational methods using the experience to improve the performance or to make accurate predictions. [...] It is the study of algorithms that learn from examples and experience instead of hardcoded rules” (p. 19). The term “experience” is used here to refer to existing databases and their properties (the training data), which are used to learn and train a model (Mohri et al., 2018). The aim is to identify patterns in the data that allow to either better describe the data, increase performance, or perform the most accurate possible prediction (Jamal et al., 2018). Let us first take a closer look at some general terms before delving into the different parts of ML.

*Datasets* are a set of examples that contain *features* to solve a problem. If we think of data in a tabular form, each row is an *instance*, and each column is a feature. Features are measurable pieces of data that are fed into a machine learning algorithm and help to understand the problem. The result is a model, which is to be understood as the trained representation of what the algorithm has learned. For example, a random forest algorithm can be trained with training data, and the output is a random forest model. New, unknown data can now be fed into the model in order to obtain predictions, classify the data, and much more, depending on the algorithm used. Thus, a predictive algorithm creates a predictive model that, when fed with new data, produces a prediction based on the data it was trained on (Kelm et al., 2020).

Artificial intelligence, big data, and machine learning are frequently mentioned in connection with the currently popular and (semi-)overused terms “smart tourism” and “smart destinations.” According to Gretzel, Sigala, Xiang, and Koo (2015), what supposedly makes destinations, cities, and tourism “smart,” generally speaking, includes the various information and communication technologies (ICTs) integrated into the physical infrastructure, smart experiences that attempt to optimize travel experiences through personalization, contextualization, and real-time analysis (Buhalis & Amaranggana, 2015) as well as the business ecosystem geared toward smartness. Smartness thus requires the processing of big data, available as transactional data, user-generated content, data provided by integrated devices and measured by sensors, etc. (Gretzel et al., 2015; Koo et al., 2016). All data types, i.e., image, audio, video, text, and metadata such as date and time values, geospatial data, tags, and more, are of great relevance. For detailed information on analytics in smart tourism design, it is recommended to read Xiang and Fesenmaier (2017).

In order to process data accordingly, recognize structures and patterns within a dataset, extract new informative features, perform far-reaching analyses and forecasts, and personalize recommendations, among other tasks, various machine learning approaches can be implemented. An analysis of multiple publications containing the search query “Tourism” AND “Machine Learning” in the article title, abstract, or keywords revealed 390 papers in Scopus and 216 papers in Web of Science. As Fig. 1 shows, increased ML use in tourism research can be observed, especially from



**Fig. 1** ML-methods in tourism literature. Source: Author’s depiction based on Scopus and Web of Science

2018 onwards. Moreover, the treemap depicts the subject areas in which papers have been published, with the field of “Computer Science Artificial Intelligence” still containing the most published articles, followed by the categories “Hospitality Leisure Sport Tourism” and “Computer Science Theory Methods.”

Among tourism-specific journals, the most noteworthy (as of June 2021) is *Tourism Management* with 81 ML publications, followed by *Annals of Tourism Research* with 30 articles and *Tourism Management Perspectives* with 26 papers.

## 1.1 The Machine Learning Process

Although the step-by-step procedure for ML projects can vary depending on the chosen algorithm, the ideal-typical process looks more or less like the stages presented in Fig. 2, starting with the collection of training data and concluding with the interpretation of the data.

Since many people associate big data with machine learning, there is often an attempt to use all the available data. However, Awad and Khanna (2015) point out that this is sometimes counterproductive and unnecessary as it seems to be more efficient to select only a subset of the data (features) that is useful for solving the problem. This data can be either in structured or unstructured form and must be prepared accordingly before being processed via an algorithm (Egger et al., 2022). For example, the data must be presented in a uniform format, and incorrect and missing data must be removed. In addition, it may be necessary to normalize, discretize, average, smooth, etc. the data so as to be able to process it further (Awad & Khanna, 2015).

Feature engineering and feature selection appear next and are of particular importance in ML since the statement “garbage in—garbage out” holds especially true; in other words, good features are the backbone of any machine learning model. It is understandable that a model can only be as good as the data with which it was trained on, and that fatal errors can occur if a model trained on bad data has been used (Sanchez, 2003). Optimally, only those features that have an influence on the

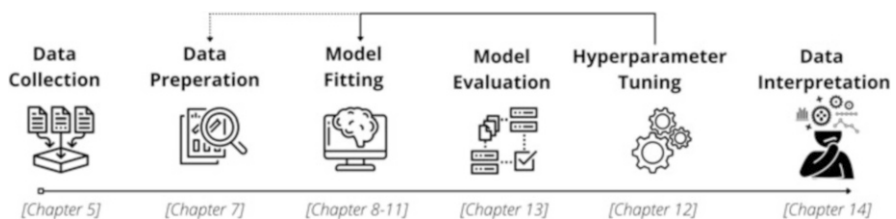


Fig. 2 ML process

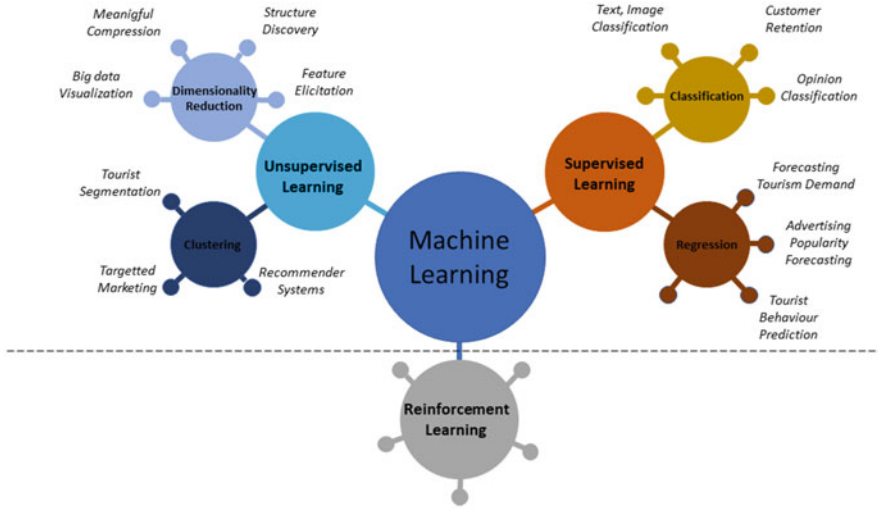
quality of the model should be selected (Wennker, 2020). (See also chapter “Feature Engineering”). The subsequent step is to train the algorithm. For this purpose, the data is split into training and testing data, where the training data is used to train the algorithm, and the testing data is used to measure its performance. In contrary to supervised learning, unsupervised learning does not require the data to be split into training and testing and, thus, does not call for cross-validation either.

Once the model has been trained, it must be evaluated. As we will show below, this is not possible for unsupervised tasks in terms of calculated key metrics because there is no ground truth label. In supervised tasks, the effectiveness and performance of the algorithm can be evaluated, and hints for optimizing the data processing as well as changeable hyperparameters are obtained. Each ML system contains hyperparameters with settings that can be changed and, in turn, affect the algorithm’s performance (Feurer & Hutter, 2019). Hyperparameter tuning is a sensitive process and requires comprehensive knowledge of the effects of such a change. Thus, to identify the settings that produce the best results, an iterative process between data preparation, model fitting, hyperparameter tuning, and model evaluation takes place. As a final step, the validated model should be applied to an actual task, for example, performing a prediction, and the results then need to be interpreted and put into its subject-specific context.

The “no free lunch theorem” states that, averaged across all optimization problems, each algorithm performs equally well when no resampling is performed (Adam et al., 2019). In other words, no algorithm works optimally for all tasks; each task has its own peculiarities and requires the correct choice of the appropriate algorithm (Egger, 2022). Therefore, numerous approaches have been developed to cater to specific tasks, and new types and forms of ML and their algorithms are constantly being developed and improved (Edwards, 2018).

There are three main types of ML algorithms, with the first, unsupervised learning algorithms, being covered in detail in chapters “Clustering” and “Dimensionality Reduction”, and the second, supervised algorithms, being discussed in chapters “Classification” and “Regression”. The third, reinforcement learning, has not been added as a separate chapter due to its comparatively low relevance for tourism cases (Jamal et al., 2018) (Fig. 3). Natural Language Processing (NLP), with algorithms for text classification, topic modeling, or sentiment analysis, is an additional, special ML case (Egger & Gokce, 2022) and will be discussed in detail in chapters “Natural Language Processing (NLP): An Introduction” to “Knowledge Graphs”; therefore, it will not be covered at this point in time.

ML approaches can be distinguished according to data type and availability of the dependent variable’s label (Edwards, 2018). Thus, either continuous or discrete dependent variables are given, which, if they contain a label, can be processed with supervised algorithms, or, if no label is given, unsupervised methods are applied (Table 1).



**Fig. 3** Machine learning paradigms and application examples

**Table 1** Learning problems and algorithms

	Supervised	Unsupervised
Discrete	Classification/categorization	Clustering
Continuous	Regression	Dimensionality reduction

Source: Skilton and Hovsepián (2018a)

In recent years, numerous ML paradigms have evolved (Fig. 4), and in addition to the supervised-, unsupervised-, and reinforcement-learning types highlighted in Fig. 3, model-based, memory-based, and deep-learning are also worth mentioning (Skilton & Hovsepián, 2018b). In particular, deep learning with neural networks can be viewed as a driver of the current ML renaissance, representing an entirely independent subfield (Choo et al., 2020). Deep learning approaches can scale the amount of data better and, subsequently, often deliver better results (Papp et al., 2019). However, it is a misconception to assume that neural networks are always superior to classical machine learning approaches (Choo et al., 2020). For a more detailed discussion on neural networks and deep learning, refer to further literature, e.g., Aggarwal (2018b) or Ekman (2021).



**Fig. 4** Machine learning paradigms and algorithms. (An interactive version of this figure is available at: <http://www.datascience-in-tourism.com/?p=132>). Source: Dobilas (2021)

## 1.2 Unsupervised Learning

Unsupervised learning algorithms attempt to identify common elements (Mich, 2020) and recognize useful structures and patterns from input data without requiring the data to be labeled. Jordan and Mitchell (2015) mention two main problems for which unsupervised methods can be helpful: (1) when missing data leads to *data sparsity*, affecting the model’s performance and accuracy, and (2) when data is presented in high-dimensional spaces, resulting in the *curse of dimensionality* phenomenon (Bernstein & Kuleshov, 2014). Overall, unsupervised algorithms take a set of predictors and analyze the relationships between them (Ozdemir, 2016). On the one hand, this can lead to identifying groups of observations that behave



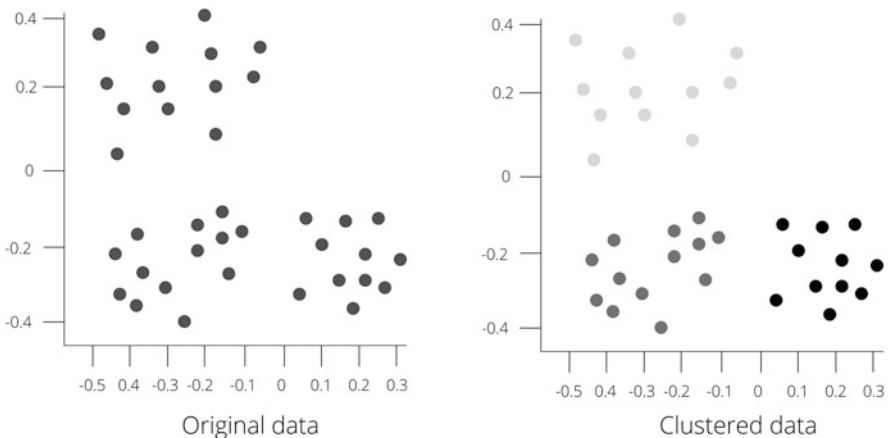
similarly due to their features, which corresponds to *clustering*, or, on the other hand, to grouping features together in order to achieve *dimensionality reduction*.

A big advantage of unsupervised learning is that the data do not require labels, which, in practice, makes it much easier to find suitable data material (Provost & Fawcett, 2013b). On the other hand, however, the predictive power is lost since only the response variable contains the information necessary for a prediction. Another major disadvantage is that it is difficult to measure how well the model works or to what extent it performs successfully (Dy & Brodley, 2004). Since there is no response variable, the model cannot be evaluated with respect to its performance, and the result ends in differences and similarities that require subjective human judgment (Ozdemir, 2016).

### 1.2.1 Clustering

A very common method of unsupervised learning is clustering, to which the aim is to identify distinct groups in the data. The data should be homogeneous within a group and show similar characteristics, and the individual groups should differ distinctly from one another, i.e., they should be heterogeneous. Thus, the goal, in this case, is not to make a prediction but, rather, to learn something about the structure and patterns inherent in the data (Arefieva et al., 2021) (Fig. 5).

Skilton and Hovsepian (2018a) list the following as information that a clustering algorithm tries to determine when searching for subsets: What is the number of subsets, and what is their size? Since there is no labeled data in unsupervised learning to specify the number of groups, this is often a nontrivial task. While there are indeed metrics to help the researcher determine the number of clusters, it ultimately boils down to a decision that must be made individually by evaluating the data. Furthermore, the question arises as to what common characteristics and



**Fig. 5** Clustering

properties the members of a group have and whether these subsets themselves exhibit structures and patterns. Typical questions could be, for example, “can tourists be divided into natural groups in terms of perceived risk with regard to COVID-19? (Neuburger & Egger, 2021)” or “into which spatial units can a city be divided in terms of its tourist use?” As shown in Fig. 4, there are many different clustering algorithms available, each developed for specific problems, which should be chosen carefully based on the intended application.

A special form of clustering is known as *community detection*. These are graph-based approaches in which similarities are not identified based on the data’s features but, instead, on the relationships between the data. In contrast to clustering methods that process tabular data, community detection (Ghosh et al., 2018) algorithms (such as Louvain, Leiden, or Markov clustering) use networks as a data source. They process a matrix of edges and nodes and detect commonalities based on membership features in networks (Fortunato & Hric, 2016). The question, therefore, is how to represent the relationships in a graph in a compact way. According to McAuley (2017), a community can be defined in cases where the members are connected to each other and where there are few edges between the communities, a high density inside and few corners outside, and a sense of “cliquishness.”

Clustering methods have always played an essential role in the context of tourism, especially to typologize tourists and their behavior, but also to group photos, reviews, destinations and their characteristics, etc. into homogeneous groups. Up until now, hierarchical clustering and k-means methods have primarily been applied. Hierarchical clusterings have been used, for example, by Neuburger and Egger (2021) to group travelers according to their perceived COVID-19 risk, Derek, Woźniak, and Kulczyk (2019), who created a typology of outdoor tourists, Batista e Silva, Barranco, Proietti, Pigaiani, and Lavalle (2020), who developed a new systematic classification of EU regions based on the predominant location of hotels, or by Del Chiappa, Atzeni, and Ghasemi (2018) to analyze residents’ perceptions and attitudes towards tourism development in Costa Smeralda, Italy. Less frequently, one may also come across studies that have used k-means clustering to investigate a tourism context. For instance, Srihadi, Hartoyo, Sukandar, and Soehadi (2016) segmented the Jakarta travel market by creating a typology of tourists based on their lifestyles, and Chua, Meng, Ryu, and Han (2021) used k-means clustering to group tourists’ volunteering in terms of their life satisfaction and attitudes toward volunteering.

The use of density-based clustering algorithms is still quite rare in tourism, although these methods are particularly suitable for spatial analysis. Park, Xu, Jiang, Chen, and Huang (2020) studied spatial structures of tourism destinations by applying a trajectory data mining approach using mobile big data. The goal was to better understand the spatial structures and interactions of tourist attractions, for which they used density-based spatial clustering of applications with noise (DBSCAN). This algorithm was also applied by Ma, Kirilenko, and Stepchenkova (2020) to cluster Instagram photos of the solar eclipse. Community detection algorithms, such as Louvain or Leiden, have also hardly been used in tourism thus far. One example is Yu and Egger’s (2021) study, in which they analyzed the

relationship between colors and user engagement of Instagram images. The Louvain algorithm was used to categorize annotated images using Google Cloud Vision on the basis of their labels. In this case, the network-based algorithm was applied to cluster Instagram photos according to their interconnected nodes, where the labels were considered as nodes.

See chapter “Clustering” for a more explicit discussion on the topic of clustering and its different approaches.

## 1.2.2 Dimensionality Reduction

Oftentimes, high-dimensional data must be processed, and the so-called “curse of dimensionality,” seen as an obstacle when using many methods, requires the complexity of the data to be reduced through the act of decreasing the dimensions. This can be achieved by identifying features that best represent the data, which, in turn, results in a smaller, more efficient dataset. Obtained in this way, the smaller size of the data is then easier to process and should help render significant information more recognizable (Bernstein & Kuleshov, 2014). It is important, however, that the loss of information associated with the reduction of data (in favor of gaining better insights into the data) remains within a reasonable range (Provost & Fawcett, 2013a).

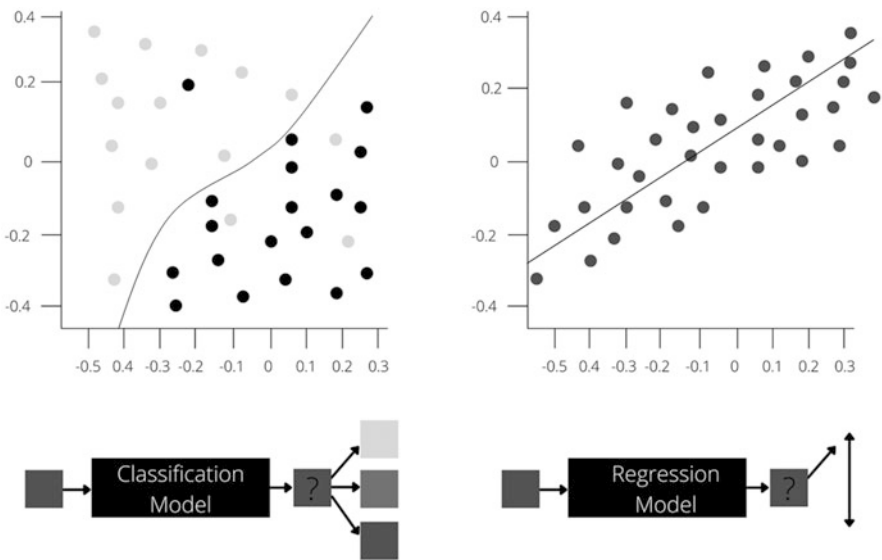
The best-known way of reducing dimensions in data is principal component analysis (PCA). In recent years, however, algorithms that are better able to preserve the local and global structure of the data and are used to reduce high-dimensional data, such as t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP), have emerged. In the context of tourism, usually the classical methods of dimension reduction are preferred; in most cases, PCA is used, while t-SNE and UMAP still rarely make their presence known. This is most likely due to the fact that most studies work with survey data, which hardly produce high-dimensional vectors unless they are generated during the process of feature engineering. These algorithms are also often used for the reduction of high-dimensional data in order to visualize them in 3D or 2D.

For example, Li, Li, Hu, Zhang, and Hu (2018) performed sentiment classification paired with a topic model approach using a bidirectional recurrent neural network. By applying `lda2vec` for topic identification, multidimensional word vector representations were created. In order to observe and visualize the process of sentiment classification in more detail, the data had to be reduced to a 2D projection, for which they used t-SNE as a dimension reduction algorithm. Another example is a study from Payntar, Hsiao, Covey, and Grauman (2021) in which high-dimensional features of geotagged internet photos were generated by applying ResNet50 and ImageNet, which are convolutional neural networks. In order to be able to map the photos onto a map for interpretation purposes, a visual embedding with t-SNE was performed afterward.

Chapter “Dimensionality Reduction” deals with the different methods of dimension reduction in more detail, with a special focus on “rarely used” methods such as t-SNE and UMAP.

### 1.3 Supervised Learning

In contrast to unsupervised learning, the target variable of a training dataset has a label, i.e., a more detailed description that can be used to train the algorithm. This makes supervised learning, especially suitable for prediction tasks. Depending on whether the target variable is continuous or discrete, it becomes either a regression or a classification problem (Fig. 6). In both cases, the attempt is made, by means of a trained prediction model, to predict an output variable  $y$  by approximating a function  $f(x)$  (Hastie et al., 2009). In this sense, the labeled training data is used to synthesize the model function by trying to generalize the relationship between the input data and the output data (Awad & Khanna, 2015). The choice of features is of particular importance here because having too many features can confuse the learning algorithm (Jamal et al., 2018). The goal is to train the algorithm, with the help of a labeled training set, so well that it is able to predict the correct class labels for a new, unseen dataset as accurately as possible (Awad & Khanna, 2015). Thus, the quality of a model is highly influenced by the training data, especially when it comes to supervised learning approaches. In practice, one may often be confronted with the



**Fig. 6** Classification vs. Regression. Source: Adapted and expanded upon from Langs and Wazir (2019)

issue that such required labels are unavailable in the training dataset and have to be created first. This can be solved partially with unsupervised approaches (semi-supervised, self-supervised tasks) (Saeed et al., 2019), but if one wants to seriously ensure data quality, then one also has to rely on human labeling, which can quickly become a complex and expensive (time-consuming) task. On the other hand, one is then able to measure the quality of the results accurately, provided that the aspect of overfitting (see chapter “Model Evaluation”), where models fit too precisely to the training data at the expense of generalization, has been taken care of as well (Provost & Fawcett, 2013a).

Oftentimes in practice, it is not so straightforward or easy to decide whether a situation is a classification or a regression problem, especially when the features are taken from a typical Likert scale and, as in most cases, can be interpreted as both ordinal and interval scaled data.

### 1.3.1 Classification

Classification is one of the most commonly used ML applications, most likely due to the fact that a myriad of different use-cases containing classification problems exist. The overall goal of classification is to divide similar data points into different classes. The most commonly used approaches include decision trees, rule-based methods, probabilistic methods, support vector machines (SVM) methods, instance-based methods, and neural networks (Aggarwal, 2015). By using different classification algorithms, various rules, depending on which of these classifications have been applied, can be identified (Cleve & Lämmel, 2020). For example, customers can be divided into classes according to their creditworthiness. For this particular case, the basis for rule generation would involve a dataset consisting of customer data where, in addition to numerous other features, the characteristic of creditworthiness would be recognized as a label. The machine would then learn from the training data and find the rules and patterns that expel the most errorless classification of the new and previously unknown data. At first glance, classification and clustering methods seem to appear very similar as they both try to segment different groups based on characteristics. Yet, the vital difference is that, for classification, the structure of the groups is determined by the given labels, while in clustering, the segmentation is done on the basis of feature similarities, with the main objective being to reveal the hidden structures in the dataset (Arabie et al., 1996).

Apart from regression cases, classification tasks can be seen as the most widely used ML approach. In their study, Ramos-Henríquez, Gutiérrez-Taño, and Díaz-Armas (2021) aimed to operationalize the value proposition of hosts on Airbnb. From more than 250 variables, they first identified those that contribute most to being classified as a “superhost.” The authors then used a SVM classifier for binary classification and naïve Bayes and Logit models as baseline models when comparing analyses. Moreover, Deng and Li (2018) presented an approach with regard to the correct selection of photos for destination image communication based on classifying them into affection categories. For this purpose, the authors trained a naïve Bayes

model with numerous “content-emotion” pairs using Flickr images to predict an emotion classification for new photos based on their content. In another study, Martinez- Torres and Toral (2019) attempted to classify reviews using a text-based ML approach by examining the content of online reviews from the hospitality sector. The goal of their paper was to identify those features in order to successfully classify reviews into either deceptive or nondeceptive reviews. For this purpose, the authors determined the TF-IDF value from the words of the reviews and used them as input data for the classification process. Keeping the “no-free lunch theorem” in mind, they applied six different classifiers and compared their results. They trained k-NN, logistic regression, SVM, random forest, gradient boosting, and multilayer perceptron (MLP) classifiers for this task.

For more information, chapter “Classification” goes into detail on the topic of classification and describes, together with an example, the different approaches.

### 1.3.2 Regression

Regression is commonly known as a set of statistical tools that models the relationship between explanatory variables and a target variable, thus describing the average relationship between numerical attributes (Shalev-Shwartz & Ben-David, 2014). There are many different regression methods in machine learning, and, in contrast to classification, continuous numerical values, rather than discrete features in the form of classes, are predicted. If one divides the predicted numerical feature into a defined number of intervals, any regression algorithm can also be used as a classification one (Akerkar, 2019b; Althbiti & Ma, 2020). Thus, regression analyses can be used to solve both prediction problems and classification problems (Skilton & Hovsepian, 2018b). An extension of linear regression is logistic regression; here, the dependent variable has a categorical characteristic, typically with binary values of 0 and 1.

In tourism literature, regression is mostly used to explain one variable with the help of a dependent variable. In contrast to this frequently used explanatory function, regression in ML mainly involves prediction. An excellent overview of tourism forecasting research using internet data is provided by Li, Law, Xie, and Wang (2021), and chapter “Time Series Analysis” also deals with the topic of time series analysis in order to forecast tourism demand.

Prediction models based on regression approaches are used in a wide variety of areas. For example, in terms of sustainability, models for predicting the impact of tourism on nature are highly relevant but widely missing. In this sense, Jahani, Goshtasb, and Saffariha (2020) used different regression models to investigate which ecological factors are associated with vegetation density regeneration and how these can be predicted. For this purpose, they compared SVM, radial basis, and multilayer perceptron models. Similarly, Han (2020) also investigated the relationship between local tourism development and its impact on environmental resources so as to develop a quantitative SVM prediction model for turnover development in Chengdu, China.

Some additional tourism examples in which different regression approaches have been used come from, for example, Guerreiro and Rita (2020), who explored the connotation of explicit recommendations based on Yelp reviews by applying logistic regression, as well as Martin-Fuentes, Fernandez, Mateu, and Marine-Roig (2018), who strengthened the efficacy of using SVM for multiclass classification. By taking properties listed on Airbnb as the case context, the accuracy of SVM proved to be higher than the logistic model in general. Moreover, to group a large number of travel blog entries, Shibata, Shinoda, Nanba, Ishino, and Takezawa (2020) applied ensemble learning using SVM with RBF kernel.

Chapter “Regression” provides further information on various regression approaches.

## ***1.4 Reinforcement Learning***

The third approach, which will not be discussed further in this book, is reinforcement learning, a form of automated, goal-directed learning and decision making (Akerkar, 2019a). Here, the focus is on the characteristics of the learning problem and not on the learning algorithm itself. In this way, any algorithm can be selected to solve the problem as long as it is suitable for solving that particular problem (Skilton & Hovsepian, 2018a). The system is given a task, and it is supposed to learn and evolve on its own based on positive or negative feedback in an attempt to maximize a numerical reward value (Ngyuen & Zeigermann, 2021).

One of the very scarce examples of applying reinforcement learning in a tourism context comes from Lu, Meng, Timmermans, and Zhang (2021). Using a deep reinforcement learning model, random forest, and a multinomial logit model, the authors investigated the hesitation of passengers to choose a connecting airport when they have a large number of online sales options via several channels at their disposal. Their results yielded that the reinforcement learning model achieved the highest accuracy, followed by the random forest algorithm and that the multinomial logit model performed much worse than the alternative approaches.

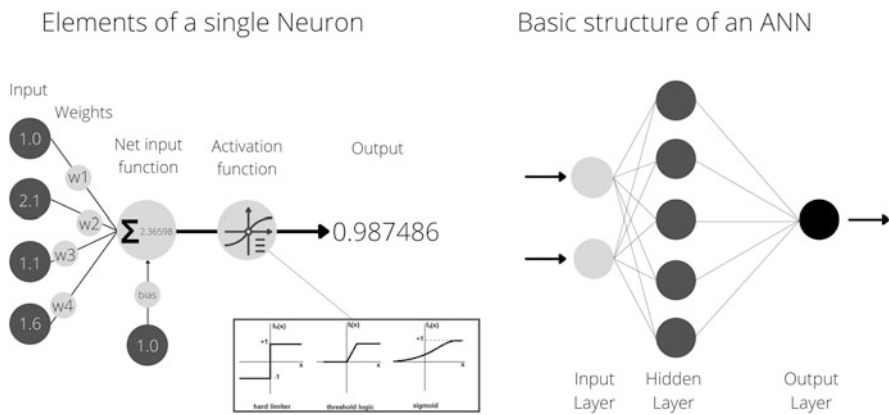
## ***1.5 Neural Networks***

Artificial neural networks (ANNs) take on a special role in ML in part because, while other algorithms may have each been developed with a specific type of task in mind, ANNs are highly adaptable and can, theoretically, learn any mathematical function (Aggarwal, 2018a). Thus, almost all networks developed for ML are “Turing complete,” meaning that they can simulate any learning algorithm, given they have enough training data (Pérez et al., 2021). Generally speaking, neural networks attempt to simulate the decision process of the neuron networks in the human brain (Graupe, 2013) so as to produce artificial intelligence. While other ML approaches

use highly complex mathematical functions designed to solve a very specific problem, ANNs work with numerous, yet elementary, computational operations, possess self-organizing properties and solve complex, mathematically ambiguous, and nonlinear or stochastic problems (Graupe, 2013). Additionally, ANNs are characterized by high parallelism, a distributed memory, and adaptability (Adeli & Hung, 1994). Compared to other ML approaches, this renders ANNs particularly robust and tolerant to error and noise (Palmer et al., 2006).

There are a variety of different network architectures that differ mainly due to their net topologies and connection types (e.g., single-layer, multi-layer, feedforward, feedbackward networks). For instance, since feedforward neural networks only pass signals from input to output, there are no cycles or loops in this network. In contrast, recurrent neural networks (RNNs) have additional connections between neurons in the same layer and previous layers. RNNs can, therefore, not only learn from the input and weights but also from the previously learned so-called hidden states.

As shown in Fig. 7, the basic structure of an ANN provides a primary input layer in which neurons receive and process the weighted input signals, where the weights reflect the strength of the connection between neurons. Each neuron has an activation function whose output is computed by functions, such as hard limit, threshold logic, and sigmoid (Ebrahimpour-Komleh & Afsharizadeh, 2015). Suppose a certain output value is reached, then the neuron “fires” and forwards the output to the next neuron. The higher the forwarded output value is, the more significant its input dimension was (Raschka & Mirjalili, 2018). The output dimensions are then combined in the next layer (hidden layer) with new further dimensions, but this step is hardly comprehensible and leads to black box models. The continuation of these processes results in a complex network with multiple connections. As mentioned before, ANNs are highly adaptable, and this adaptability threshold is reached by providing the network with feedback based on its output. As such, the ANN can



**Fig. 7** Elements and architecture of a simple ANN. Source: Adapted and extended from Ebrahimpour-Komleh and Afsharizadeh (2015)



make better predictions with each next step as it updates and adjusts the weights of the connections. By repeating this “backpropagation” several times, with larger and larger amounts of data, the system learns based on the rules it creates itself (Moolayil, 2019).

More information on the role of the bias neuron, the different activation functions, the error measurement of the forward pass, and details regarding backpropagation are beyond the scope of this chapter. For the interested reader, Graupe (2013) or Aggarwal (2018b) are recommended at this stage.

As with other ML approaches, overfitting or underfitting can also occur in neural networks. In overfitting, the algorithm learns and memorizes the training data by heart, so to speak, and processes new, unknown data inadequately. To solve this issue, Wennker (2020) recommends reducing the complexity of a network by removing neurons. However, the problem here is that one cannot know in advance how many neurons need to be removed from the network. Mathematically, an L1 and L2 regularization can help. On the other hand, to counteract underfitting, new neurons and weights should be included.

Deep learning is a subfield of machine learning, and deep artificial neural network uses. The success of deep learning is based on the advancements of network architectures such as RNNs and CNNs (Aggarwal, 2018b). As the name suggests, deep learning uses multilayer neural networks in which more layers mean a larger parameter space to be used for the learning process (Ba & Caruana, 2013). In particular, deep learning is used in image recognition and natural language processing.

Neural networks are also becoming increasingly important in tourism since, as mentioned above, they can basically be applied to any task. Phillips, Zigan, Santos Silva, and Schegg (2015), for example, used neural network analysis to examine the determinants of hotel performance by investigating the relationships of user-generated online reviews, hotel characteristics, and Revpar. Additionally, Palmer et al. (2006) suggested an ANN for tourism time series forecasting, Bloom (2005) applied a neural network to segment the international tourism market in Cape Town, South Africa, and, similarly, Kim, Wei, and Ruys (2003) used an ANN to segment the West Australian senior tourist market. Furthermore, Tsaur, Chiu, and Huang (2002) analyzed the determinants of guest loyalty to international tourist hotels, and Abubakar, Namin, Harazneh, Arasli, and Tunç (2017) used a structural equation model and an ANN to examine the influence of favoritism/nepotism and supervisor incivility on employee cynicism and job disengagement as well as the moderating role of gender. These diverse examples show that neural networks offer a vast field for application and can be used extensively in tourism contexts in order to perform a wide variety of ML tasks.

## ***1.6 Machine Learning Limitations and Challenges***

Nonetheless, despite ML's advantages, its uses and applications also come with risks and challenges. Similar to classical statistics, there are numerous points that need to be taken into account throughout the duration of the ML process in order to achieve valid and satisfying results. This starts with the quantity and quality of the data. Most of the time, data is unstructured and messy, and appropriate preprocessing is required. In addition to the quality of the data, a corresponding quantity thereof is also necessary for numerous procedures. Depending on whether the process requires an unsupervised or supervised task, this can quickly become a significant challenge. In the case of unsupervised procedures, their lack of evaluation capabilities calls for high-quality data, whereas when it comes to supervised procedures, high-quality labeled data is often not available in sufficient quantities.

In most cases, features that seem particularly promising for use as input data must first be generated and then selected (see chapter "Feature Engineering"). Moreover, during the actual ML process, the appropriate ML model must be selected, and the correct hyperparameters have to be chosen (see chapter "Hyperparameter Tuning"), which requires a good understanding of the setting values and their potential impacts on the results. Another main problem is the overfitting or underfitting issue as well as the evaluation of models (see chapter "Model Evaluation"). Finally, and importantly, the results have to be interpreted correctly; since many ML algorithms have a black box characteristic, making interpretation difficult, this is yet an additional hurdle that must be overcome (see chapter "Interpretability of Machine Learning Models").

## ***1.7 Auto-ML***

To make the benefits of machine learning available to a wider range of users and shorten a human's working time during the data science process, companies such as Google, Amazon, Microsoft, and IBM are currently focusing heavily on developing auto-ML approaches. Another aim thereof is to attract larger groups of customers to their services. Although most of the steps in such a process are automated, a human being is still necessary to provide and define the required training data for the input. In most cases, only the dataset needs to be uploaded to the cloud, and the corresponding categories need to be labeled. The system then prepares the data accordingly, selects the right algorithm, and tunes the hyperparameters. Ultimately, the result is a REST endpoint for using predictions (Janakiram, 2018). All of this happens automatically "in the background," with the entirety of the decision-making being data driven and objective, and should, in turn, save the user both time and the necessary expertise (Gijsbers et al., 2019). In addition, Auto-ML approaches are mostly offered as a cloud solution, guaranteeing sufficient memory and computing power. Hutter, Kotthoff, and Vanschoren (2019) summarize this by saying, "this can

be seen as a democratization of machine learning: with AutoML, customized state-of-the-art machine learning is at everyone’s fingertips” (p.ix).

In recent years, numerous Auto-ML solutions have been launched on the market, with certain systems being named as leading solutions. For instance, H2O AutoML was founded as a platform for nonexperts to experiment with ML (H2O, 2017), and Amazon offers a comprehensive package with AWS Sagemaker Autopilot (Das et al., 2020). Other similar solutions include Microsoft’s Azure AutoML, Google Cloud AutoML, and IBM’s AutoAI. Auto-WEKA (Hutter et al., 2019; Thornton et al., 2013), Auto-sklearn (Feurer et al., 2019), MLBox Auto-ML (Romblay, 2017), and TPOT (Olson & Moore, 2019) also exist as Auto-ML solutions but can only be used locally with your own hardware and software environment since they are not offered as cloud solutions.

For ML projects, Python, with its countless available modules, is the typical go-to software, yet R or Julia (Bezanson et al., 2017) can also be considered alternatives. For those who have no prior experience with scripting or programming languages, numerous visual computing solutions, such as the Konstanz Information Miner (KNIME), Rapidminer, or Orange3, are available. With these applications, analysis pipelines are assembled in the sense of workflows via drag and drop, and these solutions have a molar structure and are continuously expanded upon with new components. In KNIME, for example, almost all WEKA methods are available (Cleve & Lämmel, 2020). For more information on software solutions, see chapter “Software and Tools”.

## Further Readings & Other Sources

- Dulhare, U. N., Ahmad, K., & Ahmad, K. A. B. (Eds.). (2020). *Machine learning and big data: Concepts, algorithms, tools and applications*. Wiley.
- Google’s Machine Learning Crash Course. <https://developers.google.com/machine-learning/crash-course>
- Ng, A. (2021). *Machine Learning—Coursea Course with more than 4 million participants*. <https://tinyurl.com/andrewng-ml>
- Nwanganga, F., & Chapple, M. (2020). *Practical machine learning in R*. Wiley.
- Raschka, S., & Mirjalili, V. (2017). *Python machine learning: Machine learning and deep learning with python* (2nd ed.). Scikit-Learn, and TensorFlow.
- Theobald, O. (2021). *Machine learning for absolute beginners: A plain English introduction* (3rd ed.). Scatterplot Press.
- Venturi, D. (2019). *Provides an extensive list of machine learning courses provided on the Internet*. <https://www.freecodecamp.org/news/every-single-machine-learning-course-on-the-internet-ranked-by-your-reviews-3c4a7b8026c0/>

## References

- Abubakar, A. M., Namin, B. H., Harazneh, I., Arasli, H., & Tunç, T. (2017). Does gender moderates the relationship between favoritism/nepotism, supervisor incivility, cynicism and workplace withdrawal: A neural network and SEM approach. *Tourism Management Perspectives*, 23, 129–139. <https://doi.org/10.1016/j.tmp.2017.06.001>
- Adam, S. P., Alexandropoulos, S.-A. N., Pardalos, P. M., & Vrahatis, M. N. (2019). No free lunch theorem: A review. In I. C. Demetriou & P. M. Pardalos (Eds.), *Springer optimization and its applications. Approximation and optimization* (Vol. 145, pp. 57–82). Springer International. [https://doi.org/10.1007/978-3-030-12767-1\\_5](https://doi.org/10.1007/978-3-030-12767-1_5)
- Adeli, H., & Hung, S.-L. (1994). *Machine learning: Neural networks, genetic algorithms, and fuzzy systems*. 97804710. Retrieved from <https://scholar.nctu.edu.tw/zh/publications/machine-learning-neural-networks-genetic-algorithms-and-fuzzy-sys>
- Aggarwal, C. C. (2015). *Data classification: Algorithms and applications*. Chapman & Hall/CRC data mining and knowledge discovery series. CRC Press. Retrieved from <http://proquest.tech.safaribooksonline.de/9781466586741>
- Aggarwal, C. C. (2018a). *Neural networks and deep learning*. Springer International. <https://doi.org/10.1007/978-3-319-94463-0>
- Aggarwal, C. C. (2018b). *Neural networks and deep learning: A textbook*. Springer International. Retrieved from <https://books.google.at/books?id=achQdWAAQBAJ>. <https://doi.org/10.1007/978-3-319-94463-0>
- Akerkar, R. (Ed.). (2019a). *SpringerBriefs in business. Artificial Intelligence for Business*. Springer International. <https://doi.org/10.1007/978-3-319-97436-1>
- Akerkar, R. (2019b). Machine learning. In R. Akerkar (Ed.), *SpringerBriefs in business. Artificial intelligence for business* (pp. 19–32). Springer International. [https://doi.org/10.1007/978-3-319-97436-1\\_2](https://doi.org/10.1007/978-3-319-97436-1_2)
- Althbiti, A., & Ma, X. (2020). Machine learning. In L. A. Schintler & C. L. McNeely (Eds.), *Encyclopedia of big data* (pp. 1–5). Springer International. [https://doi.org/10.1007/978-3-319-32001-4\\_539-1](https://doi.org/10.1007/978-3-319-32001-4_539-1)
- Arabie, P., Hubert, L., & de Soete, G. (1996). *Clustering and classification*. World Scientific.
- Arefieva, V., Egger, R., & Yu, J. (2021). A machine learning approach to cluster destination image on Instagram. *Tourism Management*, 85, 104318. <https://doi.org/10.1016/j.tourman.2021.104318>
- Awad, M., & Khanna, R. (Eds.). (2015). *Books for professionals by professionals. Efficient learning machines: Theories, concepts, and applications for engineers and system designers*. Apress Springer Science+Business Media. <https://doi.org/10.1007/978-1-4302-5990-9>
- Ba, L. J., & Caruana, R. (2013, December 21). *Do deep nets really need to be deep?* Retrieved from <http://arxiv.org/pdf/1312.6184v7>
- Batista e Silva, F., Barranco, R., Proietti, P., Pigaiani, C., & Lavallo, C. (2020). A new European regional tourism typology based on hotel location patterns and geographical criteria. *Annals of Tourism Research*, 103077. <https://doi.org/10.1016/j.annals.2020.103077>
- Bernstein, A., & Kuleshov, A. (2014). Dimensionality reduction in statistical learning. In *ICMLA '14: Proceedings of the 2014 13th international conference on machine learning and applications* (pp. 330–335). IEEE Computer Society. <https://doi.org/10.1109/ICMLA.2014.59>
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98. <https://doi.org/10.1137/141000671>
- Bloom, J. Z. (2005). Market segmentation. *Annals of Tourism Research*, 32(1), 93–111. <https://doi.org/10.1016/j.annals.2004.05.001>
- Buhalis, D., & Amaranggana, A. (2015). Smart tourism destinations enhancing tourism experience through personalisation of services. In *Information and communication technologies in tourism 2015* (pp. 377–389). Springer. [https://doi.org/10.1007/978-3-319-14343-9\\_28](https://doi.org/10.1007/978-3-319-14343-9_28)
- Choo, K., Greplova, E., Fischer, M. H., & Neupert, T. (2020). *Machine learning kompakt*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-32268-7>

- Chua, B.-L., Meng, B., Ryu, H. B., & Han, H. (2021). Participate in volunteer tourism again? Effect of volunteering value on temporal re-participation intention. *Journal of Hospitality and Tourism Management*, 46, 193–204. <https://doi.org/10.1016/j.jhtm.2020.12.003>
- Cleve, J., & Lämmel, U. (2020). *Data mining* (3. Aufl.). De Gruyter Studium. De Gruyter Oldenbourg. Retrieved from <https://books.google.at/books?id=T0YCEAAAQBAJ>
- Das, P., Ivkin, N., Bansal, T., Rouesnel, L., Gautier, P., Karnin, Z., et al. (2020). Amazon SageMaker autopilot. In *Proceedings of the fourth international workshop on data management for end-to-end machine learning* (pp. 1–7). <https://doi.org/10.1145/3399579.3399870>
- de Romblay, A. A. (2017). *MLBox's official documentation*. Retrieved from <https://mlbox.readthedocs.io/en/latest/>
- Del Chiappa, G., Atzeni, M., & Ghasemi, V. (2018). Community-based collaborative tourism planning in islands: A cluster analysis in the context of Costa Smeralda. *Journal of Destination Marketing and Management*, 8, 41–48. <https://doi.org/10.1016/j.jdmm.2016.10.005>
- Deng, N., & Li, X. (2018). Feeling a destination through the “right” photos: A machine learning model for DMOs’ photo selection. *Tourism Management*, 65, 267–278. <https://doi.org/10.1016/j.tourman.2017.09.010>
- Derek, M., Woźniak, E., & Kulczyk, S. (2019). Clustering nature-based tourists by activity. Social, economic and spatial dimensions. *Tourism Management*, 75, 509–521. <https://doi.org/10.1016/j.tourman.2019.06.014>
- Dobilas, S. (2021, July 2). Random Forest models: Why are they better than single decision trees? *Towards Data Science*. Retrieved from <https://towardsdatascience.com/random-forest-models-why-are-they-better-than-single-decision-trees-70494c29ccd1>
- Dy, J., & Brodley, C. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5, 845–889. [https://doi.org/10.1007/SpringerReference\\_302701](https://doi.org/10.1007/SpringerReference_302701)
- Ebrahimpor-Komleh, H., & Afsharizadeh, M. (2015). Improving prediction based digital image reversible watermarking by neural networks. In *2015 international congress on technology, communication and knowledge (ICTCK)* (pp. 201–208). IEEE. <https://doi.org/10.1109/ICTCK.2015.7582671>.
- Edwards, G. (2018, November 18). Machine learning | An introduction—Towards data science. *Towards Data Science*. Retrieved from <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0#6246>
- Egger, R. (2022). Introduction: Data science in tourism. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. ix–xxiv). Springer.
- Egger, R., & Gokce, E. (2022). Natural language processing (NLP): An introduction. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 307–334). Springer.
- Egger, R., Kroner, M., & Stöckl, A. (2022). Web scraping. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 67–84). Springer.
- Ekman, M. (2021). *Learning deep learning: Theory and practice of neural networks, computer vision, Nlp, and transformers using Tensorflow*. [S.l.]: ADDISON-WESLEY. Retrieved from <https://books.google.at/books?id=W8dFzgEACAAJ>
- Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated machine learning* (pp. 3–34). Springer International.
- Feurer, M., Klein, A., Eggenberger, K., Springenberg, J. T., Blum, M., & Hutter, F. (2019). Auto-sklearn: Efficient and robust automated machine learning. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated machine learning* (pp. 113–134). Springer International.
- Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports*, 659, 1–44. <https://doi.org/10.1016/j.physrep.2016.09.002>
- Ghosh, S., Halappanavar, M., Tumeo, A., Kalyanaraman, A., Lu, H., Chavarria-Miranda, D., et al. (2018). Distributed Louvain algorithm for graph community detection. In *2018 IEEE International Parallel 2018* (pp. 885–895). <https://doi.org/10.1109/IPDPS.2018.00098>
- Gijsbers, P., LeDell, E., Thomas, J., Poirier, S., Bischl, B., & Vanschoren, J. (2019, July 1). *An open source AutoML benchmark*. Retrieved from <http://arxiv.org/pdf/1907.00909v1>

- Graupe, D. (2013). *Principles of artificial neural networks. Advanced series in Circuits & Systems: v.7*. World Scientific. Retrieved from <https://ebookcentral.proquest.com/lib/subhh/detail.action?docID=1336559>
- Gretzel, U., Sigala, M., Xiang, Z., & Koo, C. (2015). Smart tourism: Foundations and developments. *Electronic Markets*, 25(3), 179–188. <https://doi.org/10.1007/s12525-015-0196-8>
- Guerreiro, J., & Rita, P. (2020). How to predict explicit recommendations in online reviews using text mining and sentiment analysis. *Journal of Hospitality and Tourism Management*, 43, 269–272. <https://doi.org/10.1016/j.jhtm.2019.07.001>
- H2O. (2017). *AutoML: Automatic machine learning—H2O 3.32.1.3 documentation*. Retrieved from <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>
- Han, J. (2020). Application of SVM model to environmental resource analysis in tourism development. *Journal of Physics: Conference Series*, 1629, 12007. <https://doi.org/10.1088/1742-6596/1629/1/012007>
- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Springer eBook collection mathematics and statistics) (2nd ed.). Springer. <https://doi.org/10.1007/978-0-387-84858-7>
- Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated machine learning*. Springer International. <https://doi.org/10.1007/978-3-030-05318-5>
- Jahani, A., Goshtash, H., & Saffariha, M. (2020). Tourism impact assessment modeling of vegetation density for protected areas using data mining techniques. *Land Degradation and Development*, 31(12), 1502–1519. <https://doi.org/10.1002/ldr.3549>
- Jamal, S., Goyal, S., Grover, A., & Shanker, A. (2018). Machine learning: What, why, and how? In A. Shanker (Ed.), *Bioinformatics: Sequences, structures, phylogeny* (pp. 359–374). Springer Singapore. [https://doi.org/10.1007/978-981-13-1562-6\\_16](https://doi.org/10.1007/978-981-13-1562-6_16)
- Janakiram, M. (2018, April 15). *Why AutoML is set to become the future of artificial intelligence*. *Forbes*. Retrieved from <https://www.forbes.com/sites/janakirammsv/2018/04/15/why-automl-is-set-to-become-the-future-of-artificial-intelligence/?sh=bcd730780ae>
- Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science (New York, N.Y.)*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>
- Kelm, O., Gerl, K., & Meißner, F. (2020). Machine learning. In I. Borucki, K. Kleinen-von Königslöw, S. Marschall, & T. Zerback (Eds.), *Handbuch Politische Kommunikation* (pp. 1–9). Springer Fachmedien Wiesbaden. [https://doi.org/10.1007/978-3-658-26242-6\\_55-1](https://doi.org/10.1007/978-3-658-26242-6_55-1)
- Kim, J., Wei, S., & Ruys, H. (2003). Segmenting the market of West Australian senior tourists using an artificial neural network. *Tourism Management*, 24(1), 25–34. [https://doi.org/10.1016/S0261-5177\(02\)00050-X](https://doi.org/10.1016/S0261-5177(02)00050-X)
- Koo, C., Shin, S., Gretzel, U., Hunter, W. C., & Chung, N. (2016). Conceptualization of smart tourism destination competitiveness. *Asia Pacific Journal of Information Systems*, 26(4), 561–576. <https://doi.org/10.14329/apjis.2016.26.4.561>
- Langs, G., & Wazir, R. (2019). Machine learning. In S. Papp, W. Weidinger, M. Meir-Huber, B. Ortner, G. Langs, & R. Wazir (Eds.), *Handbuch Data Science: Mit Datenanalyse und Machine Learning Wert aus Daten generieren* (pp. 178–198). Hanser.
- Li, Q., Li, S., Hu, J., Zhang, S., & Hu, J. (2018). Tourism review sentiment classification using a bidirectional recurrent neural network with an attention mechanism and topic-enriched word vectors. *Sustainability*, 10(9), 3313. <https://doi.org/10.3390/su10093313>
- Li, X., Law, R., Xie, G., & Wang, S. (2021). Review of tourism forecasting research with internet data. *Tourism Management*, 83, 104245. <https://doi.org/10.1016/j.tourman.2020.104245>
- Lu, J., Meng, Y., Timmermans, H., & Zhang, A. (2021). Modeling hesitancy in airport choice: A comparison of discrete choice and machine learning methods. *Transportation Research Part A: Policy and Practice*, 147, 230–250. <https://doi.org/10.1016/j.tra.2021.03.006>
- Ma, S., Kirilenko, A. P., & Stepchenkova, S. (2020). Special interest tourism is not so special after all: Big data evidence from the 2017 Great American Solar Eclipse. *Tourism Management*, 77, 104021. <https://doi.org/10.1016/j.tourman.2019.104021>

- Martinez-Torres, M. R., & Toral, S. L. (2019). A machine learning approach for the identification of the deceptive reviews in the hospitality sector using unique attributes and sentiment orientation. *Tourism Management*, 75, 393–403. <https://doi.org/10.1016/j.tourman.2019.06.003>
- Martin-Fuentes, E., Fernandez, C., Mateu, C., & Marine-Roig, E. (2018). Modelling a grading scheme for peer-to-peer accommodation: Stars for Airbnb. *International Journal of Hospitality Management*, 69, 75–83. <https://doi.org/10.1016/j.ijhm.2017.10.016>
- McAuley, J. (2017). *CSE 158: Web mining and recommender systems*. Retrieved from <https://cseweb.ucsd.edu/classes/wi17/cse158-a/>
- Mich, L. (2020). Artificial intelligence and machine learning. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-tourism* (pp. 1–21). Springer International. [https://doi.org/10.1007/978-3-030-05324-6\\_25-1](https://doi.org/10.1007/978-3-030-05324-6_25-1)
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. In *Adaptive computation and machine learning*. Cambridge Massachusetts (2nd ed.). The MIT Press.
- Moolayil, J. (2019). *Learn Keras for deep neural networks*. Apress. <https://doi.org/10.1007/978-1-4842-4240-7>
- Neuburger, L., & Egger, R. (2021). Travel risk perception and travel behaviour during the COVID-19 pandemic 2020: A case study of the DACH region. *Current Issues in Tourism*, 24(7), 1003–1016. <https://doi.org/10.1080/13683500.2020.1803807>
- Ngyuen, C. N., & Zeigermann, O. (2021). *Machine learning: Kurz & gut* (2. Auflage). dpunkt.verlag, O'Reilly, Preselect.media GmbH.
- Olson, R. S., & Moore, J. H. (2019). TPOT: A tree-based pipeline optimization tool for automating machine learning. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *The springer series on challenges in machine learning. Automated machine learning* (pp. 151–160). Springer International. [https://doi.org/10.1007/978-3-030-05318-5\\_8](https://doi.org/10.1007/978-3-030-05318-5_8)
- Ozdemir, S. (2016). *Principles of data science: Learn the techniques and math you need to start making sense of your data*. Packt. Retrieved from <http://proquest.tech.safaribooksonline.de/9781785887918>
- Palmer, A., José Montaña, J., & Sesé, A. (2006). Designing an artificial neural network for forecasting tourism time series. *Tourism Management*, 27(5), 781–790. <https://doi.org/10.1016/j.tourman.2005.05.006>
- Papp, S., Weidinger, W., Meir-Huber, M., Ortner, B., Langs, G., & Wazir, R. (Eds.). (2019). *Handbuch Data Science: Mit Datenanalyse und Machine Learning Wert aus Daten generieren*. Hanser. <https://doi.org/10.3139/9783446459755>
- Park, S., Xu, Y., Jiang, L., Chen, Z., & Huang, S. (2020). Spatial structures of tourism destinations: A trajectory data mining approach leveraging mobile big data. *Annals of Tourism Research*, 84, 102973. <https://doi.org/10.1016/j.annals.2020.102973>
- Payntar, N. D., Hsiao, W.-L., Covey, R. A., & Grauman, K. (2021). Learning patterns of tourist movement and photography from geotagged photos at archaeological heritage sites in Cuzco, Peru. *Tourism Management*, 82, 104165. <https://doi.org/10.1016/j.tourman.2020.104165>
- Pérez, J., Barceló, P., & Marinkovic, J. (2021). Attention is Turing-complete. *Journal of Machine Learning Research*, 22, 1–35. Retrieved from <https://www.jmlr.org/papers/volume22/20-302/20-302.pdf>
- Phillips, P., Zigan, K., Santos Silva, M. M., & Schegg, R. (2015). The interactive effects of online reviews on the determinants of Swiss hotel performance: A neural network analysis. *Tourism Management*, 50, 130–141. <https://doi.org/10.1016/j.tourman.2015.01.028>
- Provost, F., & Fawcett, T. (2013a). *Data science for business: What you need to know about data mining and data-analytic thinking* (1st ed.). O'Reilly Media. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=619895>
- Provost, F., & Fawcett, T. (2013b). *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly.

- Ramos-Henríquez, J. M., Gutiérrez-Taño, D., & Díaz-Armas, R. J. (2021). Value proposition operationalization in peer-to-peer platforms using machine learning. *Tourism Management*, 84, 104288. <https://doi.org/10.1016/j.tourman.2021.104288>
- Raschka, S., & Mirjalili, V. (2018). *Machine Learning mit Python und Scikit-Learn und TensorFlow: Das umfassende Praxis-Handbuch für Data Science, Deep Learning und Predictive Analytics* (K. Lorenzen, Trans.) (2., aktualisierte und erweiterte Auflage). mitp.
- Saeed, A., Ozelebi, T., & Lukkien, J. (2019). Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2), 1–30. <https://doi.org/10.1145/3328932>
- Sanchez, J. S. (2003). Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7), 1015–1022. [https://doi.org/10.1016/S0167-8655\(02\)00225-8](https://doi.org/10.1016/S0167-8655(02)00225-8)
- Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.
- Shibata, N., Shinoda, H., Nanba, H., Ishino, A., & Takezawa, T. (2020). Classification and visualization of travel blog entries based on types of tourism. In J. Neidhardt & W. Würndl (Eds.), *Information and communication Technologies in Tourism 2020* (pp. 27–37). Springer International. [https://doi.org/10.1007/978-3-030-36737-4\\_3](https://doi.org/10.1007/978-3-030-36737-4_3)
- Skilton, M., & Hovsepian, F. (2018a). Machine learning. In M. Skilton & F. Hovsepian (Eds.), *The 4th industrial revolution* (pp. 121–157). Springer International. [https://doi.org/10.1007/978-3-319-62479-2\\_5](https://doi.org/10.1007/978-3-319-62479-2_5)
- Skilton, M., & Hovsepian, F. (Eds.). (2018b). *The 4th industrial revolution*. Springer International. <https://doi.org/10.1007/978-3-319-62479-2>
- Srihadi, T. F., Hartoyo, Sukandar, D., & Soehadi, A. W. (2016). Segmentation of the tourism market for Jakarta: Classification of foreign visitors' lifestyle typologies. *Tourism Management Perspectives*, 19, 32–39. <https://doi.org/10.1016/j.tmp.2016.03.005>
- Thornton, C., Hutter, F., Hoos, H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and Hyperparameter optimization of classification algorithms. In I. S. Dhillon & R. L. Grossman (Eds.), *Kdd 2013: The 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, August 11–14, 2013, Chicago, Illinois, USA ; [including] the Industry Practice Expo (IPE) [and] co-located workshops* (pp. 847–855). ACM.
- Tsaur, S.-H., Chiu, Y.-C., & Huang, C.-H. (2002). Determinants of guest loyalty to international tourist hotels—A neural network approach. *Tourism Management*, 23(4), 397–405. [https://doi.org/10.1016/S0261-5177\(01\)00097-8](https://doi.org/10.1016/S0261-5177(01)00097-8)
- Wennker, P. (2020). Machine learning. In P. Wennker (Ed.), *Künstliche Intelligenz in der Praxis* (pp. 9–37). Springer Fachmedien Wiesbaden. [https://doi.org/10.1007/978-3-658-30480-5\\_2](https://doi.org/10.1007/978-3-658-30480-5_2)
- Xiang, Z., & Fesenmaier, D. R. (2017). *Analytics in smart tourism design*. Springer International. <https://doi.org/10.1007/978-3-319-44263-1>
- Yu, J., & Egger, R. (2021). Color and engagement in touristic Instagram pictures: A machine learning approach. *Annals of Tourism Research*, 103204. <https://doi.org/10.1016/j.annals.2021.103204>



# Feature Engineering



## Human-in-the-Loop Machine Learning

Pablo Duboue

### Learning Objectives

- Expand the concept of ML as a black box to a partnership between the data scientist and the computer
- Illustrate the value of domain expertise for data science in a specific context/niche, such as tourism
- Explain the basics of feature engineering: selection, expansion, and homogenization
- Appreciate the impact of nuisance variations

## 1 Introduction

Feature Engineering (FE) is a human-centric process where humans and computers work together to solve a problem using data. In contrast, pure Machine Learning (ML) focuses on solving the problem solely from data, with as little human intervention as possible. FE involves modifying the data based on intuitions and experience to make problems easier to solve by computer. As Géron (2017) states in his book:

Researchers try to find algorithms that work well even when the agent initially knows nothing (...) unless you are writing a paper, you should inject as much prior knowledge as possible (...) as it will speed up training dramatically. (p. 591)

That idea applies to all industrial uses of ML. Data scientists in the tourism domain are very likely to have relevant, domain-specific experience or at least work in a team

---

P. Duboue (✉)  
Textualization Software Ltd., Vancouver, BC, Canada  
e-mail: [pablo@artoffeatureengineering.com](mailto:pablo@artoffeatureengineering.com)

with people who do (Egger & Yu, 2022). This chapter will help to leverage that experience to build better-performing models. The keyword here is “better” but not “quicker.” Sadly, FE is a slow process that can involve a fair amount of trial and error. Moreover, success is not guaranteed. As such, if enough data is available, Deep Learning (DL) can be tried first (which does complex feature synthesis automatically), or an AutoML framework (that might automatically search how to perform large numbers of FE operations under the hood) can be used. When neither of these approaches fits the problem or if there is a strong will to improve features before embarking on an AutoML or DL approach, the techniques sketched here can help.

FE is a very extensive topic, on which full books, such as Duboue (2020), have been written. The objective of this chapter is to inform the data scientist about FE concepts to help them decide whether FE is a viable tool for their problem at hand. If they decide to perform FE, Galli (2020) is a great place to start. FE is also one of the most programming-intensive parts of applied ML. The case study in this chapter is lengthy in terms of coding and shows a non-trivial FE example within the domain; studying it in detail can help the data scientist get a better sense of FE.

Brynjolfsson and McAfee (2012) put forth that the future of employment is hybrid, that tasks we once considered exclusive to humans are becoming a partnership between humans and computers. In data science, that means that if tasks seem automatable, they will be done so, and practitioners need to make the human side count. They can focus on improving algorithms (a task normally reserved for people with special ML expertise), improving the utility of the output of the model (and the business aspects surrounding it), or focus on improving the input (and the particular data domain). FE allows data scientists working in the tourism domain to achieve input data improvements beyond what a computer can do on its own.

The FE process proposed here hinges on two types of analyses to be run by the data scientist. The first one is Exploratory Data Analysis (EDA), a staple of statistical analysis. It involves getting acquainted with the variability, distribution, statistical behavior of the different input features, possibly some of their interactions, and their relation to the target class or value we are trying to predict or regress. After a successful EDA, we should have an idea of which features are favorable and which ones we might want to ignore. Moreover, we might be able to start envisioning some feature combinations that could make sense for the problem at hand (“is this feature greater than this other feature?” might be meaningful for a given problem).

ML models are designed to be constrained because such constraints are key to the generalization of training data to new, unseen data. Training data has a variation that carries information and variations that are unrelated to the problem at hand. Unrelated variations are referred to as “nuisance variations.” To diminish the impact of nuisance variations, we prefer very constrained ML models that cannot learn such variations. That also means that certain insights (like “feature A is greater than feature B when the target class is 0”) will either not be captured at all or will require massive amounts of data to be elicited. Providing extra features that have already

been computed directly (also called “computable features”) along with this information is a great way to help the ML model solve the problem.

Another technique includes feature selection, that is, dropping unhelpful features. For example, suppose we have the license plate number for a potential customer’s car. The value of the license plate has no information that can help us categorize this lead, but, for two otherwise identical customers, we will now have an extra variation based on their license plates. Dropping them will clearly reduce the number of nuisance variations. Similarly, it is possible to reduce the number of features and increase the signal in the data by automatically projecting the data into a smaller space. This is called “dimensionality reduction,” to which there are a number of techniques; the most common one is PCA, but others might work better depending on the nature of the available data. These are discussed further in chapter “Dimensionality Reduction.”

Finally, we will discuss expanding the input data with external, additional data. This “thinking-outside-the-data box” can be very powerful, particularly in the tourism domain, as many data sources are not publicly available and might need to be purchased or acquired in-house or through strategic partnerships.

FE rewards practitioners that are methodical and keep a well-written track of the different experiments they have done with the data, the hypotheses behind said experiments, and the outcomes for each of them. Jupyter Notebooks can be used as an experimental logbook, but it requires a great deal of copying and pasting to do so. It is recommended to keep a separate record in wiki or document format. While being methodical pays off in FE, being an automaton does not. Build intuitions about the data and the problem, and apply them judiciously. Many new practitioners expect simple rules of thumb that can be used for different FE techniques. When such rules exist, then they have already been implemented into ML and AutoML frameworks, and there is no need for a human-in-the-loop process.

## ***1.1 Definitions***

In this chapter, we will discuss FE for supervised learning, that is, for the construction of classifiers (categorical targets) and regressors (numeric targets). For readers new to these concepts, they are explained in chapters “Classification” and “Regression” in more detail. We start with raw data plus target values and undergo a featurization process in which the raw data is transformed into a fixed-size feature vector that will be fed to the ML trainer. We consider the following feature types: binary (true and false), categorical (one among a small set of values), discrete (integer), continuous (floating point), and complex (sets, lists, strings, etc.; most systems profit from decomposing those).

Our objective in FE is to obtain good features, that is, features that are informative (they describe something that makes sense to a human), useful (they are defined for as many instances as possible), and discriminant (they divide instances according to

the different target values). See Baker (2015) for more details on quality feature creation.

FE operations with parameters tunable on the data itself are called trainable FE operations. If the target value (either class or floating point number) is used, they are called supervised FE operations; otherwise, they are unsupervised.

## 1.2 *Feature Engineering Cycle*

The basic ML cycle involves splitting the available data into train and test sets, training a model based on the training set and evaluating it in the testing set. The FE cycle contains the basic ML cycle and iterates it with a human-in-the-loop approach. In each iteration, different FE operations are chosen by the data scientist based on two types of analyses: EDA and EA.

EDA is used to analyze datasets and summarize their characteristics, usually through visual means. It helps to formulate hypotheses about the data, pick an ML model, and initiate featurization. It should be performed every time new raw data comes in. There is always the temptation to jump into model building right away, but this should be resisted. EDA is very data-dependent and is usually taught in statistics courses. It is possible to gain some insights by running a battery of standard statistical tests and characterizing the distribution of the data. See Pyle (2002) for a discussion thereof within data science and the case study at the end of this chapter for an example.

Once a model has been trained, error analysis (EA) can be performed. Models are usually evaluated through aggregate metrics (error rate, etc.). These metrics are a good starting point, but, for FE, it is useful to have a detailed look in order to identify individual erroneous instances or classes of instances that contribute substantially to the error. EA is also a great moment to engage future users of the trained model as their feedback might go beyond errors found in the model itself. Communication is much easier with real output from a trained model. We will now discuss FE operations: combining, selecting, and expanding features.

## 2 **Combining Features**

A great way to separate nuisance variations from meaning-carrying variations is to analyze the types of variations we encounter across the whole dataset. Therefore, if a given variation is not related to a variation in the target value, we can conclude that we are better off removing it. We will discuss normalizing features, discretization, handling missing data, and descriptive features.

## 2.1 Normalization

Feature normalization is a great way to reduce variations present in the feature values. It also assigns feature values within a specific range, for example, floating point between 0 and 1 (or  $-1$  and  $1$ ). This is crucial for SVMs and NNs that need input data scaled to specific values. For a detailed discussion, see Zheng and Casari (2018), chapter “Epistemological Challenges.”

The most common approach is scaling: given a training set, we compute, for a given column, the maximum observed value ( $M$ ) and the minimum observed value ( $m$ ), and then use, as a feature for a value  $v$  in that column, the result of

$$f_v = \frac{v - m}{M - m}$$

This scales the original values to the interval  $[0, 1]$ . This is an unsupervised FE operation. Note that  $M$  and  $m$  ought to be computed on the training set; computing  $M$  and  $m$  for the full dataset is a common mistake. The trained FE operation should behave appropriately for unseen values smaller than  $m$  or larger than  $M$  as they might appear in production use. If  $M$  and  $m$  are computed throughout the entire dataset, the impact of values outside the range will be unevaluated.

Other normalization operations include centering (moving from  $[0, 1]$  to  $[-1, 1]$ , for example) and standardization (dividing by the standard deviation).

## 2.2 Discretization

Discretization is the process of transforming a continuous signal into discrete values and can help reduce the number of parameters from the machine learning model. Discretization can be supervised (using the target class) or unsupervised (truncation, rounding, and binning, when we divide the values into segments of either equal frequency or equal width).

## 2.3 Missing Data

Many times, the data has instances where a particular feature value is unknown due to a variety of reasons. Ignoring rows in the raw data that have missing columns is poor practice if such missing data can also appear in production. Instead, it is better to perform some imputation on them, that is, to fill in the missing value with a value as incongruous as possible so that the ML model will not zoom in on that value while predicting. In a simpler case, the median value for the feature can be used. In the case study, we will impute using the mode, the most frequent value.

## 2.4 *Descriptive Features*

For raw data with a large number of similar features (e.g., word counts or pixel colors), it is possible to produce features describing the raw data rather than all the actual values in the raw data themselves. For example, we can have a feature that records the maximum value observed in many columns of the raw data, and, likewise, the minimum, the average, or the number of non-zero values.

This can be generalized/visualized via histograms, a simple representation of the distribution of the values in the raw data. For each particular value that can appear in the raw data, the histogram records the number of times the value was observed. For continuous values, a discretization of them can be used instead. They are very popular in image processing and in natural language processing (NLP).

## 3 Reducing Features

Spurious features result in a large number of nuisance variations; as such, it turns out that, for many, “feature selection” is a synonym of FE. Besides feature selection, another technique to reduce the number of features is dimensionality reduction, which will be discussed in chapter “Dimensionality Reduction”. We will start by looking at the key concept of feature importance.

### 3.1 *Feature Importance*

If we want to filter features, we need to measure how well a particular subset of features behaves. In a simpler case, we can define a metric for each individual feature, which is fast but does not take feature interaction into account. For example, in the case study, neither latitude nor longitude is a good feature on its own, but, when combined, they do help. It is common for combinations of features to be more informative than the features themselves, as the other feature values provide each other with meaningful context. For individual feature metrics, we can use a maximum likelihood estimator, that is, how well the feature (alone) predicts the target class, or the chi-square correlation between the feature and the target class.

One feature utility that works well in practice is mutual information, the amount of bits of information about a second random variable that is obtained through knowing the value of a first random variable:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

As a feature utility metric, we compute the mutual information between the feature being evaluated and the target class, with the probabilities in the formula being estimated by using counts on the training data.

In the case study, we look into individual feature importance by plotting curves against the target values during EDA. Once we have a trained Random Forest (RF) model, we will use the RF's feature importance to perform the feature selection. The RF's feature importance includes feature interactions and is of better quality than any individual feature importance metric, including the aforementioned maximum likelihood, mutual information, and chi-square.

While feature importance metrics seek to elucidate the importance of different features for the overall behavior of the ML model, it is also possible to find the importance of different feature values for an *individual prediction*. This is important for EA and model interpretability (a topic discussed in chapter “Interpretability of Machine Learning Models”). For this purpose, in the case study, SHAP values will be used (Lundberg & Lee, 2017).

### 3.2 Feature Selection

A common mistake is to use every column in the raw data as a feature and, thus, avoid the need for EDA. However, raw data likely contains columns unrelated to the target class (i.e., “id” in the case study). Removing unhelpful features might help strengthen the signal in the data. If quality EDA has already been performed, robust ML algorithms will benefit from less helpful features, as we will see in the case study, where a filtered dataset underperformed.

Given a feature importance metric, we can take the top N features of the metric, or plot their scores, and choose a suitable cut-off point (as done in the case study), in what is known as feature filtering. Alternatively, we can use wrapper methods and retrain a full system on different feature subsets. The total number of possible subsets is exponential in the number of features; therefore, this approach is usually tackled through approximate, greedy methods. We can either start from an empty feature set and add one feature at a time (forward feature selection or “incremental construction of the feature set”) or start from the entire dataset and remove one feature at a time (backward feature selection or “ablation study”).

In forward feature selection, we train a new system for each potential new feature and stop adding features when the error rate stops decreasing. For example, given feature A, B, C, D, we will train the ML on  $\{A\}$ ,  $\{B\}$ ,  $\{C\}$ ,  $\{D\}$  and evaluate all four models. If the model on  $\{C\}$  presents a lower error, we will then train on  $\{C, A\}$ ,  $\{C, B\}$ ,  $\{C, D\}$ . If the  $\{C, A\}$  presents a lower error, we will then train on  $\{C, A, B\}$ ,  $\{C, A, D\}$ . If no error reduction is achieved with either of these two models, we will use  $\{C, A\}$  as the final feature set.

In backward feature selection, we train on the full feature set and remove one feature at a time, again retraining the system on the reduced feature set and evaluating them. We maintain the feature set that keeps the error stable and stops

removing features once the error increases. It tends to capture feature interaction better than forward feature selection, but it involves training larger models and, as such, is more time-consuming.

## 4 Expanding Features

After a discussion regarding the reduction of features, it might seem counterintuitive to talk about expanding features, but, for FE in the tourism domain, there might be plenty of value to gain in doing so. We will discuss using functions in relation to the raw data columns or using external data to provide more contextual information to the original information.

### 4.1 *Computable Features*

The more straightforward way of expanding features is by operating over them by computing new features using the existing ones as input. For example, in the case study, we believed that the amenities string, which is long and has many substrings in it, only has value if the substrings “TV” or “Washer” appear on it. Writing suitable functions allowed this insight to be executed.

By far a more simple type of computable feature is a feature drill down, where we find a good feature and make changes to it. There are a number of variations to be considered; for example, if the feature is binary, we can transform it to a probability-based one, depending on how well it correlates with target classes, or if it is categorical, we can reduce the number of categories by adding an “other” category, as done in the case study. Alternatively, if it is discrete, we can threshold it or split it into bins. A feature drill down is a good place to start FE work.

The most common computable features are single feature transformations, which work on a single feature using different operations, including  $e^x$ ,  $\log x$ ,  $x^2$ ,  $x^3$ ,  $\tanh x$ ,  $\sqrt{x}$ , and sigmoid. For example, if the variable seems to have boundary issues (it grows too much or reduces too much by a few instances), it can be suppressed by taking the log, as done with the “price” variable in the case study. One-hot encoding, discussed next, also falls in this category.

### 4.2 *One-Hot Encoding*

Not all ML models can use categorical features directly. In that case, we can take a categorical feature with N categories and expand it into N boolean features, one for each category, such that only one of them will be true (“hot”) for a given instance. For example, if we have three categories (“cable,” “dsl,” “fibre”), we can change the



**Table 1** One-hot encoding example

Internet		has_cable	has_dsl	has_fibre
dsl	→	F	T	F
fibre		F	F	T

feature “Internet” to three features “has\_cable,” “has\_dsl,” “has\_fibre,” as shown in Table 1.

This representation accommodates set-valued columns in the raw data naturally and is used in the case study to encode the value of the “amenities” column (where a property can have both a “TV” and “Wifi” so the boolean features are no longer exclusive).

### 4.3 Decomposing Complex Features

ML algorithms expect their features to be simple values. Many times, raw data contains fields where multiple pieces of information have been aggregated. This step involves decomposing such columns. By far the most common case is decomposing date values. In the case study, the column “host\_since” contains a date that has been processed to only keep its year.

### 4.4 External Data

A travel agent might mention that the shoes potential customers wear are a great predictor of the type of destinations they might be interested in visiting. Now, how might it be possible to expand a dataset with the types of shoes potential customers wear? This might be harder than the original problem. Luckily, in this age of continuous data mining, it might be possible to either purchase such signals themselves or purchase proxy signals (such as “does this person like Crocs® on Facebook?”) from a variety of vendors, or there might be additional sources of information within the organization building the model.

However, linking existing entries to the new received information is not trivial, and data cleaning and preparation efforts will increase. For instance, there might be many more cases of missing data in the expanded dataset, and there will be a need to spend time imputing the data appropriately. For example, if the problem involves geographical data, adding distance to major points-of-interest might help (this will be discussed further in chapter “Geographic Information System (GIS)”). The case study below expands the dataset with distances to SkyTrain stations (Vancouver’s subway-like system).

## 5 Practical Demonstration: Airbnb Pricing

Airbnb is an online platform that allows everyday people to offer hospitality services. It is very popular in cities with significant tourism influx and expensive hotels, such as Vancouver, Canada. In this case study, we will use scraped data to predict pricing for new properties, that is, we use information available to a new host to try to elicit the price per night (a regression task).

This case study is built on the author’s knowledge of the city (as he is Vancouver-based) and Airbnb insights from the book “Start Your Own Travel Hosting Business” from Rich and Entrepreneur Media, Inc. (2017). According to the book, for a new host putting a property on the market, pricing is the most crucial decision, and it is much harder than in other hospitality contexts since Airbnb is driven by reviews, and high prices drive high expectations. *Mispricing generates bad reviews even if everything else was perfect.*

The code for this case study is available together with the code for this book, but it can also be accessed directly on Google Collab (just follow the link and click Runtime > Run all):



The code is written using pyspark, the Python interface for the Spark-distributed execution system. It benefits from the FE functionality available in Spark’s MLlib. FE in Python is usually presented using scikit-learn and Pandas (see Galli (2020) for such an approach), but the author’s opinion is that pyspark is better suited for the types of large datasets available in the tourism industry. Spark also keeps track of the type of columns, something very useful when doing complex FE operations. Note that Airbnb has a proprietary in-house FE framework written using Spark, as presented by Simha (2021).

### 5.1 Dataset and EDA

The dataset we are using has been graciously provided by Murray Cox as part of their [InsideAirbnb.com](https://insideairbnb.com) project. It contains close to 5000 listings in the Vancouver metro area. Originally, it was in a format that pyspark could not load directly (comma-delimited file with text fields spanning multiple lines). Moreover, it was much larger than necessary for our purposes as it contains all the text blobs from the property descriptions; however, NLP is not discussed until later in this book (chapter “Natural Language Processing (NLP): An Introduction”). It also contains review

**Table 2** Inferred types from the source data (partial)

Column	Type
id	Integer
host_since	String
host_location	String
host_response_time	String
host_response_rate	String
host_acceptance_rate	String
host_is_superhost	String
host_neighbourhood	String
host_listings_count	Integer
host_total_listings_count	Integer
host_verifications	String
host_has_profile_pic	String
host_identity_verified	String
neighbourhood	String
neighbourhood_cleansed	String
neighbourhood_group_cleansed	String

information that will not be available to our target application. Using a spreadsheet program, 15 columns were filtered to obtain the initial file for analysis.

From the cleaned csv file, pyspark can infer a schema, that is, it can guess an initial type for each feature by setting the option “inferSchema” to “True” when calling the csv reader. There are many columns with usable types (Table 2). Others seem to be assigned strings when a more suitable type might work (e.g., date for “host\_since”), but we need to do some EDA on them, and, for that, we need to keep some withheld data to avoid basing our intuitions on the whole dataset (which would not leave additional data over to test whether the intuitions are correct).

## 5.2 Data Split

We are using 20% of the data for final testing and 40% of the training data as the development test. Changing this data split might arrive at different conclusions (see, for example, the discussion in Silberzahn et al. (2018)). In general, the conclusions obtained from data should be double checked on a new data segment.

Let us now take a peek at the training data to gain some insights. The total number of columns we have is 45, but two columns have data that will need to be expanded (e.g., “amenities” with values such as “Oven,” “Gym,” “Hangers,” etc. and “host\_verification” with values such as “email” and “phone”). Using the show() method from the data frame, we can have an idea of what the different columns mean. Table 3 shows some highlights. Our target variable for regression (“price”) is also shown in the table.

**Table 3** Selected columns from the 2021-02-09-vancouver-listings.csv dataset

Id	Price	Amenities (partial)	Host Since	Latitude	Neighbourhood
Int	Int	String	String	Float	String
10080	150	["kitchen," "oven," "gym"]	2009/08/10	49.287	Coal Harbour
16254	195	["wifi," "smoke alarm"]	2009/12/15	49.277	Hastings-Sunrise
248014	100	["bbq grill," "tv," "dryer"]	2011/10/16	49.291	Grandview-Woodland

The string fields will need special processing before we can make sense of them. Some of them, such as “property\_type,” seem to have categorical data (specific strings that belong to a small set), and we will have to transform them to indices (integers indicating the category). Others are duplicates of the same information (e.g., there are three “neighbourhood” columns). For those, we will select the one with the highest quality to add to the feature vector. Finally, “amenities” and “host\_verifications” are set-based features, represented as strings, requiring quite a bit of processing.

### 5.3 Feature Transformers

Before analyzing how these columns behave with respect to our target value (“price”), let us handle the categorical data by encoding it to indices. This is accomplished in pyspark by using special feature transformers, aptly named *indexers*. In pyspark, we can define different feature transformers and apply them all together as part of a pipeline. We will be recording the different transformers that are used in a list, allowing us to execute the transformers on new data and experiment with variations of the pipeline later on.

### 5.4 Indexing Categorical Features

For this case study, three key features will be considered: which neighborhood the property is in, the type of property, and the room type. Spark has `StringIndexer`s that are very strict and make an exception if a new string is found. We will relax them with an “other” category and leave unseen categories and rare categories as “other.”

For the case of “neighbourhood,” we query the data to find out that there are 23 neighborhoods in the dataset; 564 properties appear to be located downtown, 191 in Kitsilano, and it continues to decrease quickly thereafter. We will group all neighborhoods that appear less than 50 times under “OTHER,” lowering the number of neighborhood categories to 14 categories (including “OTHER”). The same approach can be done to “property\_type” and “room\_type.” Once the strings have been reduced with the “OTHER” category, we can feed them into the Spark’s `Indexer`.

### 5.5 Set Features: Amenities and Host Verifications

Once we have dealt with the categorical features, we can address the set-based columns. We want to transform them into multiple columns reusing the machinery for NLP in MLlib. We first define a unary transformer that splits the string at the commas and removes quotes and brackets. Then, we can use a vectorizer to transform the arrays of strings to arrays of indices. We keep the number of amenities and verifications under control by setting the vocabulary size of the vectorizer and keeping only the most frequent items.

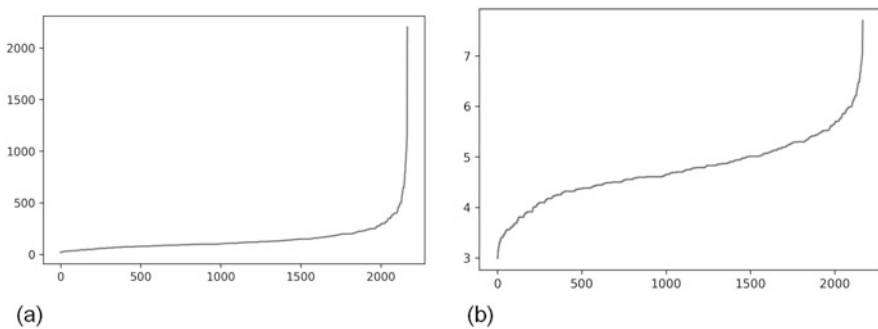
### 5.6 Decomposing Complex Features: host\_since

Finally, we will deal with the date in “host\_since” by changing the string to an integer via keeping only the year component of the date.

### 5.7 EDA

We will now perform some visualizations (this topic is discussed in detail in chapter “Visual Data Analysis”). We start by plotting our target value for regression, “price”, in Fig. 1a. This curve can profit from flattening by using a log to better understand it, as shown in Fig. 1b. This approach is then captured by a suitable feature transformer.

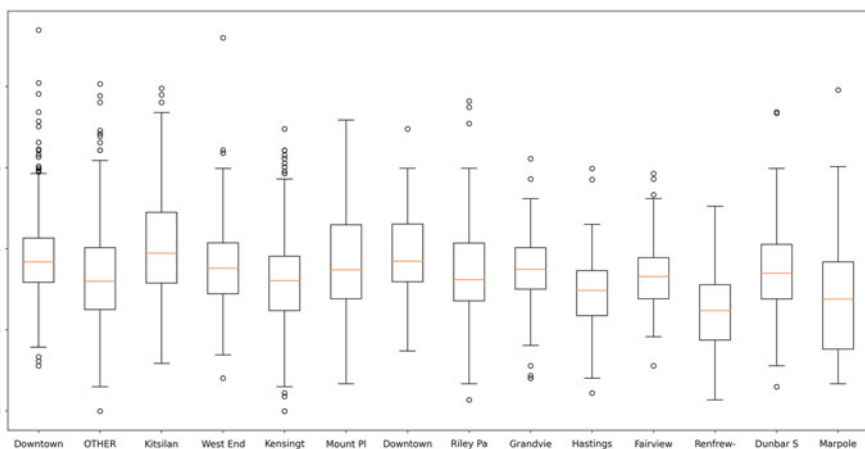
We can now plot each of the numeric columns against the target (log\_price). This produces a number of plots. Figure 2 shows some highlights thereof. First, in the figure, the column “id” is uncorrelated. That column is expected to be uninformative. The next column shown, “latitude,” also seems to be uncorrelated, but we expect the actual location (latitude, longitude) to be meaningful. Finally, the number of people



**Fig. 1** Price in training dataset. Base values vs. after applying a log function. Source: Companion Jupyter Notebook



**Fig. 2** EDA: Columns “id,” “latitude,” and “accommodates” plotted against the logarithm of the column “price.” Source: Companion Jupyter Notebook



**Fig. 3** Neighborhood vs. log prices. Source: Companion Jupyter Notebook

that can dwell in a given property (column “accommodates”) shows a rather marked correlation.

For categorical data, we use box plots, which show the median, the second and third quartiles, and the extremes of the distribution of values. The box plot for neighborhood (Fig. 3) shows it has the potential to be a very informative feature. The one for amenities (not pictured) shows a slight improvement for “TV,” which means that offering a TV pays slightly better on average. That might be par for a place where it tends to rain for 6 months straight. The box plot for host verifications (not pictured) contains no noticeable difference.

From this analysis, about 20 columns look good for a starting set, but “availability” and “host\_since” will not be computable for new properties. This is a common situation when using historical data for ML, where information that is available during training is not available during execution. We can save ourselves some trouble and avoid such features.

## 5.8 Imputation

Before we can feed this data to the ML trainer, we need to deal with undefined values in each of the selected columns. We will replace them with the *mode*, the most common value for that column.

While plotting the number of rows that are in need of imputation, we can see that the number of bedrooms has more than a hundred rows that need imputing. That is the case because the original scraped data is a string (“2 bedrooms and a den”), and the integer is obtained by processing that string beforehand. Such cases will be imputed with the most common value for “bedrooms,” which in this dataset is 1.

## 5.9 Base Model

To obtain a feature vector from the dataset, we use a vector assembler identifying the columns that make up the features. The “amenities\_vector” column is concatenated into the output vector. The total number of features is 36, with 16 base features plus 20 amenities. From this feature vector, we can train a random forest (RF).

To test the trained model, we reuse all the feature transformations that we applied to the development test in a pipeline. We evaluate the model using Spark’s “RegressionMetrics” functionality. It computes a number of metrics, but we will focus on the mean absolute error as it is easier to understand for practical applications. Now, in our case, the absolute error might not be as important as the positive errors (the times when the model said a property was worth more than it actually was). An improved metric is left as an exercise for the reader. The absolute error is 56.83 Canadian dollars. To put the figure into context, we can compute the average price for the training set, which is 150 dollars. Now, remembering that the dataset has some properties that command very high prices (Fig. 1), it is possible that these skew the average. Instead, we can also compute the median (the value that appears in the middle of the sorted list of prices). That is 115 dollars. Therefore, 56 dollars seems quite erroneous, and this model is most likely not ready to be used in production.

## 5.10 First Iteration

Let us see which features are better based on the RF feature importance (Table 4). We can see that “bedrooms” and “accommodates” are clear winners. The individual amenities come all at the end (the table is truncated to the top 16 features). Almost all amenities but “TV” and “Washer” come after the base features. We can see that many amenities are not crucial for price prediction, which might be because they are expected amenities (like “Dishes and silverware”) or not important for the type of tourism received by Vancouver (e.g., “Dedicated workspace”).

**Table 4** RF feature importance for the base system

Rank	Feature	Importance
1	<b>bedrooms</b>	0.2976
2	<b>accommodates</b>	0.1427
3	room_indexed	0.1127
4	calculated_host_listings_count_private_rooms	0.1073
5	calculated_host_listings_count_entire_homes	0.0829
6	beds	0.0597
7	property_indexed	0.0471
8	<b>neighbourhood_indexed</b>	0.0285
9	minimum_nights	0.0212
10	minimum_minimum_nights	0.0159
11	calculated_host_listings_count	0.0139
12	host_listings_count	0.0102
13	maximum_minimum_nights	0.0100
14	<b>amenity: TV</b>	0.0070
15	minimum_nights_avg_ntm	0.0060
16	<b>amenity: Washer</b>	0.0044

### 5.11 Feature Selection: Dropping Amenities

We can keep the only two amenities that seem to matter, “TV” and “Washer,” by dropping the processing of amenities through Splitter and CountVectorizer and, instead, use a spotter to check whether the original amenities string contains “TV” and “Washer” to produce two binary features. The new feature set, therefore, has 18 features. Smaller feature sets are also easier for humans to understand. The new dataset produces an error of 56.64 on the development test set, a slight decrease but too small to be trusted.

### 5.12 Second Iteration

To try to get better improvements, we can look at the feature importance scores again and, instead of removing features, try to seek out features that are not as important as we believe they are. In Table 4, the neighborhood feature is somewhat in the middle. However, intuition indicates that the neighborhood should be a great feature (Location! Location! Location!). We know most of our training properties are downtown; thus, it is possible the neighborhood is not as valuable as we would like it to be because the value of a location is related to its distance, for example, to SkyTrain stations (Vancouver’s elevated subway alternative). We can try to expand the dataset based on this insight.



### 5.13 Expanding Using External Data: SkyTrain Stations

In order to do that, we will have to go back to the latitude and longitude columns in the raw data and compute a rough approximation of the distance to the closest SkyTrain station using a list of stations obtained from [OpenStreet map](#)<sup>1</sup> (available under the [Open Database License](#)<sup>2</sup>).

To calculate accurate distance results, the curvature of the earth needs to be taken into account using a GIS library, but we will ignore this issue in this example. Since Vancouver is rather north, that means that differences in longitude will be stronger than differences in latitude. For detailed information about working with GIS data, see chapter “Geographic Information System (GIS)”.

The process reduced the error by almost a dollar (\$55.96), which seems like a step in the right direction. The new feature is less relevant than neighborhood but seems to contribute nonetheless, most likely complementing less desirable neighborhoods with the information that the property is close to a SkyTrain station.

### 5.14 Case Study Wrap-Up

To conclude, we can do a drill down on the instances with the highest absolute error using SHAP values. We first initialize a SHAP explainer (for more details, see chapter “Interpretability of Machine Learning Models”) using our model and identify the top rows that contribute the most to the absolute error. Looking at the first two rows, we realize that these are actual errors on the Airbnb website: new hosts that set a monthly price rather than a daily one. From this understanding, during a second try, such outliers can be cleaned from the data. The third row shows a useful SHAP plot (Fig. 4). This is an entire house that accommodates six people. Going back to the source data, it is described as a “luxury condo with an outdoor hot tub.” We can see that the neighborhood and distance to SkyTrain went against the pricing, but the description says it is very close to Granville Island and its shops and restaurants. Adding more POIs (points-of-interest) and, of course, applying NLP to the descriptions should help here.



**Fig. 4** SHAP values for an error instance. Red features contributed to a higher value, while blue ones decreased value. Source: Companion Jupyter Notebook

<sup>1</sup> [https://wiki.openstreetmap.org/wiki/Vancouver\\_SkyTrain](https://wiki.openstreetmap.org/wiki/Vancouver_SkyTrain)

<sup>2</sup> <https://opendatacommons.org/licenses/odbl/>

## 6 Conclusions

FE is a set of techniques that allow human knowledge and intuitions to be added to an ML solution. There are a number of well-understood methods and transformations that can be applied to the features. This process is better done iteratively, starting from EDA and performing EA after each iteration. As it is time-consuming in general, if there is enough data, good results could be obtained by using DL without FE. If DL does not work out of the box, a pre-trained DL model may be available, and Transfer Learning is worth trying. If the data is too small for DL and the problem is too rare for pre-trained models to be available, AutoML can be applied. Finally, if the solutions put forth by AutoML are not good enough, it is possible to study them and improve them using FE, tapping into domain experts to go forward faster and achieve better results.

For the sake of presentation, teaching material in FE typically shows techniques that have been applied to problems that end up working well. However, FE in practice involves much trial and error, and as little as 10–20% of the FE operations attempted on the data might result in an actual improvement of the final ML results. As FE is so labor intensive, there are efforts to capture quality features across organizations in the form of Feature Stores, where data engineers can add documentation and versioning information to different feature sets.

### Service Section

**Main Application Fields:** FE is normally the preferred technique for working with small datasets as they do not have enough data to automatically build features from them. When there is enough data available, the data itself behaves as context to give meaning to the data. Alternatively, this type of context for the data can be obtained from human experts if they are available and have strong intuitions about the data meaning and behavior.

**Limitations and Pitfalls:** The main limitation of FE is that it is very labor intensive. A practitioner can expect to go through many, many cycles of FE refinement. Each FE cycle involves analysis of the errors produced by the current feature set and the construction of an improved feature set based on hypotheses generated during the analysis. This takes many hours of analyst time. Moreover, it does not guarantee success, which, in turn, means not all the feature refinement cycles will succeed in improving results (10% of the attempts resulting in improvement is typical). Managing expectations when applying FE is therefore complicated. Finally, FE has a strong danger of overfitting; it is possible to overengineer the features and obtain a model that will perform poorly in production if the data available had a different bias than the production data.

(continued)

**Similar Methods and Methods to Combine with:** Many feature transformations can be attempted automatically, in what it is known as AutoML. Therefore, it is recommended to apply AutoML first, especially software tools like Featuretools that will provide feature transformations in a format that can be edited further. Finally, hyper-parameter tuning has to be executed in each FE cycle as the optimal parameters will change when the features are changed. Failing to do so will result in suboptimal features and poor error analysis.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-7-Feature-Engineering>

## Further Readings and Other Sources

- Duboue, P. (2020). *The art of feature engineering: Essentials for machine learning*. Cambridge University Press.
- Galli, S. (2020). *Python feature engineering cookbook*. Packt Publishing.
- Holbrook, R. (2021). *Kaggle tutorials: Learn feature engineering* (online resource). Retrieved from <https://www.kaggle.com/learn/feature-engineering>
- Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.
- Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: Principles and techniques for data scientists*. O'Reilly Media.

## References

- Baker, R. (2015). *Course webpage HUDK5053: Feature engineering studio* (online resource). Retrieved from <http://www.columbia.edu/~rsb2162/FES2015/>
- Brynjolfsson, E., & McAfee, A. (2012). *Race against the machine: How the digital revolution is accelerating innovation, driving productivity, and irreversibly transforming employment and the economy*. Digital Frontier Press.
- Egger, R., & Yu, C.-E. (2022). Data science and Interdisciplinarity. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications* (pp. 35–49). Springer.
- Géron, A. (2017). *Hands-on machine learning with scikit-learn and tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.
- Lundberg, S., & Lee, S.-I. (2017). *A unified approach to interpreting model predictions*. Advances in Neural Information Processing Systems.
- Pyle, D. (2002, January 28). *Data preparation for data mining*. Morgan Kaufmann.
- Rich, J., & Entrepreneur Media, Inc. (2017). *Start your own travel hosting business*. Entrepreneur Press.
- Silberzahn, R., Uhlmann, E. L., Martin, D. P., Anselmi, P., Aust, F., Awtrey, E., et al. (2018). Many analysts, one data set: Making transparent how variations in analytic choices affect results. *Advances in Methods and Practices in Psychological Science*, 1(3), 337–356.
- Simha, N. (2021, July 21). *Zipline—A declarative feature engineering framework* (Video file). Retrieved from [https://www.youtube.com/watch?v=LjcKCM0G\\_OY](https://www.youtube.com/watch?v=LjcKCM0G_OY)

# Clustering



## Hierarchical, $k$ -Means, DBSCAN

Matthias Fuchs and Wolfram Höpken

### Learning Objectives

- Learn typical applications of clustering within the tourism domain
- Explain the conceptual foundations of widely used clustering approaches
- Illustrate a step-by-step application of major clustering approaches on real tourism data using the data science platform *RapidMiner*<sup>®</sup>
- Demonstrate a tourism case that applies clustering approaches to identify points of interest based on uploaded photo data from the platform Flickr

## 1 Introduction and Theoretical Foundations

*Clustering* represents one of the most commonly used quantitative analysis techniques in tourism, typically applied to the task of market segmentation (Baggio & Klobas, 2017; Dolnicar, 2021). Cluster analysis aims to identify classes, also labeled as clusters, of the most similar cases within a dataset. Clusters may represent any type of object, which is represented as a statistical *case*, such as individuals (e.g., travelers or tourism entrepreneurs) but also tourism products, firms, etc. More formally speaking, a cluster analysis means grouping cases based on their similarity as given by the multivariate characteristics representing the cases of a particular

---

M. Fuchs (✉)

Department of Economics, Geography, Law and Tourism, Mid Sweden University, Östersund, Sweden

e-mail: [matthias.fuchs@miun.se](mailto:matthias.fuchs@miun.se)

W. Höpken

Institute for Digital Transformation, University of Applied Sciences, Weingarten, Germany

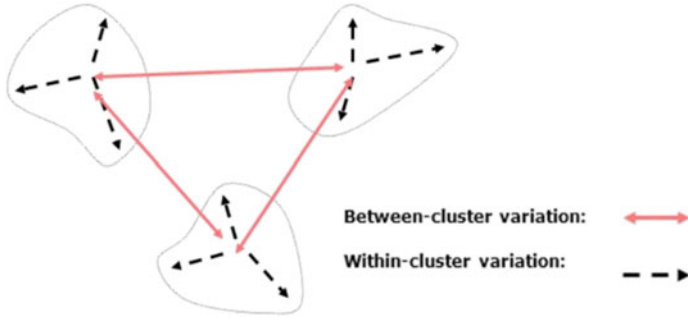
e-mail: [wolfram.hoepken@rwu.de](mailto:wolfram.hoepken@rwu.de)

sample or the population itself (Baggio & Klobas, 2017). Before describing the theoretical foundations of the most widely applied clustering approaches in the tourism domain, the typical application areas of cluster analysis in contemporary tourism science and generally in practice will be touched upon.

As mentioned, market segmentation represents the classical application domain of cluster analysis in tourism. A good example is the segmentation study by Hudson and Ritchie (2002), who used cluster analysis to identify domestic tourist segments in the Alberta region of Canada. Firstly, 13 influential factors driving tourism decision-making, such as the quality of accommodation, the variety of tourism activities offered, holiday periods, and weather conditions, were identified through qualitative research. As a second step, 3000+ residents were interviewed by telephone to assess the importance of these influential factors. On the basis of these answers, and by additionally considering demographic characteristics, the cases of this representative sample were clustered into five market segments: the young urban outdoor market, the indoor leisure traveler market, the children-first market, the fair-weather friends-visiting market, and the older cost-conscious market. More recently, Neuburger and Egger (2020) employed cluster analysis to identify segments of travelers at two different points in time based on their perceived risk of COVID-19, perceived risk of traveling during the pandemic, and travel behavior regarding a change, cancellation, or avoidance of travel (plans). The study identified three clusters (i.e., the anxious, the nervous, and the reserved) with distinctive characteristics. In addition, results revealed a significant increase in risk perception of COVID-19, travel risk perception, and travel behavior over a short period of time.

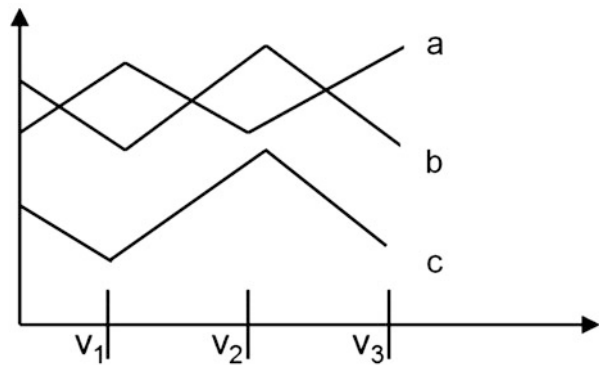
As suggested by Dolnicar (2021), in the future, market segmentation in tourism will harvest its primary strength by using web-based data, such as online search data (Fuchs et al., 2014; Höpken et al., 2015), web navigation data (Pitman et al., 2010), or online feedback data (Dietz et al., 2020) as opposed to relying on surveys or interview data. In line with this claim, Höpken et al. (2020) recently examined the suitability of different clustering techniques to identify points of interest based on uploaded photo data extracted from the photo-sharing platform Flickr. We will discuss the details of this work in more depth below. As the latter example shows, tourism studies can apply clustering to means beyond solely market segmentation, for instance, to meaningfully group tourism suppliers, such as lifestyle entrepreneurs in nature-based tourism (Fuchs et al., 2021). In this vein of analysis, Scholochow et al. (2010) employed cluster analysis to group 700 Austrian hotel managers based on their behavioral pattern of having adopted e-Business technologies to improve their companies' efficiency and effectiveness.

As a vast array of cluster analysis techniques exists (Everitt et al., 2011; Liu, 2011), our discussion will be restricted to the three most widely applied clustering approaches within the tourism domain, namely, *hierarchical* clustering, *k*-means, and DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) (Tan et al., 2018). In terms of an optimal mathematical solution, clusters exhibit high internal homogeneity (i.e., minimum *within-cluster* variation) and high external heterogeneity (i.e., maximum *between-cluster* variation), as shown in the cluster diagram in Fig. 1 (Hair et al., 2014).



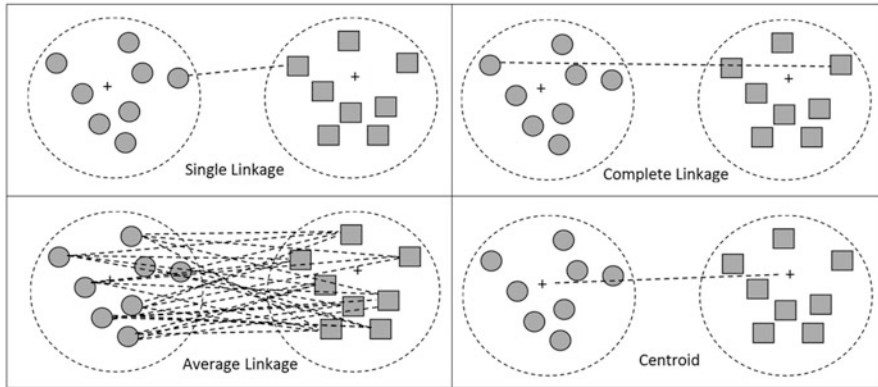
**Fig. 1** Between- and within-cluster variation (see: Hair et al., 2014, p. 439)

**Fig. 2** Inter-object similarity as difference or correlation (see: Hair et al., 2014, p. 430)



In fact, cluster analysis tries to minimize the differences between cases within each cluster by simultaneously maximizing the differences between clusters. Thus, one issue that is common to all clustering techniques is the representation of similarity (or difference) between pairs of cases or pairs of clusters, respectively (Baggio & Klobas, 2017). Differences are typically derived by a *similarity* measure (Everitt et al., 2011). Yet, the choice of a particular similarity measure does not only depend on the scale level of the cluster variables (i.e., binary, categorical, or metric), but, rather, it is mainly influenced by the fact that inter-object similarity (i.e., case/case, case/cluster, or cluster/cluster) can either be detected by *correlation* or *distance*-based measures of similarity. More precisely, by focusing on cluster variables' patterns, correlation measures (e.g., *Tanimoto*) can interpret the correlation of patterns as a similarity. By contrast, distance measures regard the magnitude of the distance. Accordingly, distance measures ponder an object-pair as similar if its variables show a low difference in magnitude (Hair et al., 2014). Looking at Fig. 2, object-pair a–b is viewed as similar based on a *distance*-based measure, while object pair b–c is viewed as similar by means of a *correlation*-based measure.

Notably, in clustering practice, distance-based measures of similarity tend to dominate. A prominent measure is the *Mahalanobis distance*, a standardized form



**Fig. 3** Bases for calculating differences between cases (see: Baggio & Klobas, 2017, p. 78)

of the *Squared Euclidean distance*, which takes the co-variances between cluster variables into account (Baggio & Klobas, 2017).

Clustering methods typically allow, or require, the specification of the *points* within clusters between which distances are calculated, or, in other words, on which base clusters are subsequently *formed* (ibid, 2017). In the case of *nearest neighbor*, or *single linkage*, distances are defined as the distance between the closest elements in the clusters. The approach is most convenient if clusters are poorly delineated and tend to build long and slender chains. By contrast, *farthest neighbor*, or *complete linkage*, calculates distances between the farthest elements in the clusters, which is useful if clusters are compact and have consistent diameters (Hair et al., 2014). *Average linkage* computes distances as the average of all pairwise distances, thereby tending to combine clusters with similar variance, and, finally, *centroid distance* calculates distances between the geometric centers (*centroids*) of the clusters (see Fig. 3).

## 1.1 Hierarchical Cluster Analysis

Cluster analysis techniques are typically divided into *hierarchical* and *partitioning* categories (Everitt et al., 2011; Tan et al., 2018). *Hierarchical cluster analysis* is subdivided even further into *divisive* and *agglomerative*. The former starts by placing all objects into one single large cluster and progressively subdividing the one cluster into two, thereby maximizing the differences between the clusters obtained from each division (Baggio & Klobas, 2017). In contrast to this *top-down* approach, *agglomerative* techniques operate in the opposite *bottom-up* direction; they begin by defining each case as a single cluster and then by continuously combining the pairs that are most similar until all cases and clusters are conjoined in one cluster. A popular hierarchical *agglomeration* algorithm, in the case that all

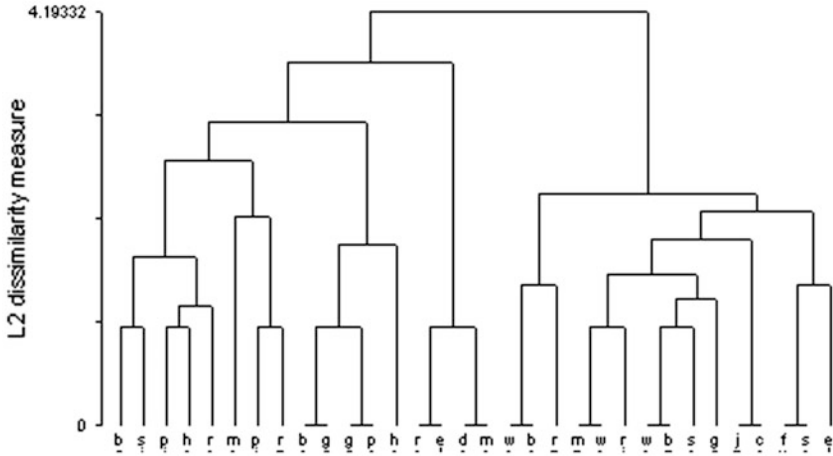


Fig. 4 Dendrogram (Source: Authors' illustration)

clustering variables are metric, is *Ward's linkage* (Ward, 1963). Considered a particular case of the single linkage algorithm, it combines clusters that minimize the within-cluster sum of squares across the complete set of clusters (Hair et al., 2014). Thus, the combined clusters are those that minimize the increase of the total sum of squares across all cluster variables. This popular algorithm tends to generate clusters of similar sizes. An alternative to *Euclidean distance*, typically employed with *Ward's linkage*, is *cophenetic distance* (Tan et al., 2018).

As highlighted, the process of joining clusters continues until the most distant (i.e., different) clusters are united. Notably, the further apart clusters join in later agglomeration stages, the more likely they are to form meaningfully distinct clusters (Baggio & Klobas, 2017). Therefore, when identifying an optimal number of the most meaningful clusters, an analyst normally refers to both the *agglomeration schedule* as well as a particular plot, the *dendrogram*. The former shows at which (typically late) stages the relative increase of the *agglomeration coefficient* appears to be particularly large, thus pointing at a merge of quite distinct, characteristic, and meaningful clusters. The latter provides a *rescaled* graphical illustration of the distance between clusters joined at each stage (Hair et al., 2014; see Fig. 4). Additional aid in identifying the optimal number of clusters can be provided by coefficients like *Silhouette scores*, *Calinski–Harabasz index*, and *Davies–Bouldin* (Tan et al., 2018).

## 1.2 Partitioning

In contrast to hierarchical clustering methods, non-hierarchical procedures do not involve the treelike construction process of clusters (Everitt et al., 2011; Baggio &

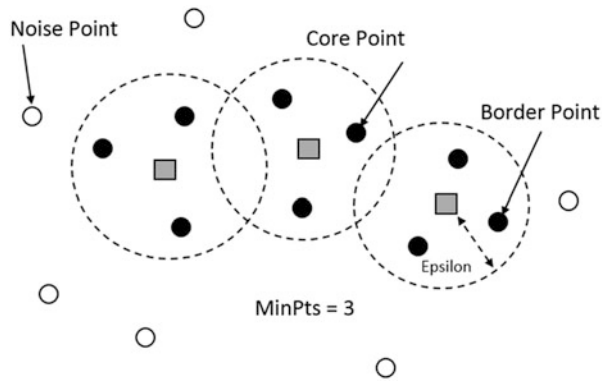


Klobas, 2017). Instead, they assign objects to clusters once the number of clusters is pre-specified (Hair et al., 2014). *Partitioning* procedures work based on a simple principle. As shown in Fig. 1, they seek to simultaneously maximize the distance between clusters and minimize the differences between in-group objects (ibid, 2014). In order to identify this optimum, partitioning procedures typically apply a series of *iterative* computations. The most widely used partitioning technique is known as *k*-means clustering (Kanungo et al., 2002). The *k*-means clustering algorithm partitions a dataset into a predefined number of clusters in which each data point belongs to the cluster with the nearest (i.e., most similar) cluster mean, or *centroid* (Lloyd, 1982). As finding the optimal clustering solution is computationally difficult, *k*-means is a heuristic algorithm, starting with a randomly chosen partition and then iteratively optimizing the solution by re-calculating the means (or centroids) and re-assigning data points to clusters accordingly (Larose & Larose, 2014; Tan et al., 2018). Cases continue to be moved until the sum of within-group variances is minimized (Baggio & Klobas, 2017). Through this iterative procedure, a potentially optimal solution can be identified. More concretely, in contrast to hierarchical cluster analysis, a 3-cluster solution is not merely a combination of 2 clusters from a 4-cluster solution; rather, it can be considered the “best possible” 3-cluster solution. Despite this advantage, one limitation of *k*-means involves that it requires the number of clusters to be explicitly specified and it can only partition a dataset into hyper-spherical or hyper-ellipsoid clusters, which, in turn, tend to be of similar size (Liu, 2011). Moreover, as *k*-means is sensitive to outliers, a distance-based outlier detection is also typically needed and is, therefore, recommended as a data preparation step (Pyle, 1999).

### 1.3 *Density-Based Spatial Clustering of Applications with Noise*

Developed by Ester et al. (1996), *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)* enables the identification of especially spatial clusters based on the density of the data points. This approach differentiates between *core points*, lying in a *high*-density region, *border points*, lying at the edge of a *high*-density region, and *noise points* (i.e., outliers) lying in a *low*-density region. Accordingly, core points need to be surrounded by a minimal number of other points within proximity of their close neighborhood, while border points need to closely neighbor a core point. Noise points fulfill none of these requirements. *DBSCAN* clusters are built by starting with any core point that has yet to belong to a cluster and then successively adding core points or border points that lie in close proximity to a core point already belonging to a cluster (Tan et al., 2018). The *DBSCAN* algorithm can be parameterized by the minimal density within a cluster (*minPts*), in other words, the minimal number of data points that have to lie within the neighborhood of a core

**Fig. 5** Density-based spatial clustering of applications with noise (Source: Authors' illustration)



point as well as the size of the neighborhood ( $\epsilon$ , epsilon); thus, the maximal distance between two neighboring points (Fig. 5).

As *DBSCAN* uses a density-based definition of a cluster, it is relatively resistant to noise and can handle clusters of arbitrary shapes and sizes. Therefore, *DBSCAN* can find many clusters that would otherwise be undetectable when using *k*-means. *DBSCAN*, however, shows weaknesses when the clusters have widely varying densities or are built on the basis of high-dimensional data due to density being difficult to define for such data. In the case of high-dimensional data, *DBSCAN* can be of low performance since the computation of the nearest neighbors requires computing all pairwise proximities (ibid, 2018). Additionally, defining the neighborhood size (*epsilon*) and the number of neighbors (*minPts*) appropriately can be difficult in certain application areas. Here, the hierarchical extension *HDBSCAN* may be more suitable.

## 1.4 Cluster Evaluation and Profiling

Typically, in the final stages of cluster analysis, the obtained clusters need to be interpreted (Larose & Larose, 2014). On the one hand, *profiling* identifies and describes the most typical characteristics of each cluster usually by means of investigating the maximal score values of the cluster variables. On the other hand, through *labeling*, a tag, which most accurately describes its nature, is assigned to each cluster. For the final validation step, significance tests between cluster variables (e.g., ANOVA) along with a multiple discriminant analysis, which estimates the share of cases correctly classified to cluster membership on the basis of a discriminant function composed by the cluster variables, is recommended (Hair et al., 2014). As previously mentioned, to identify the optimal number of clusters, an analyst may refer to agglomeration schedules and the dendrogram as well as to coefficients, such as Silhouette scores, Calinski-Harabasz index, and Davies-Bouldin (Tan et al., 2018). Ultimately, however, it remains the analyst's judgment, based on theoretical

and practical knowledge, to decide which variable sets should be used to build the clusters and to determine the final number of clusters that best represent a set of cases (Baggio & Klobas, 2017). Cluster analysis is, therefore, considered both a “science and an art” (Hair et al., 2014).

## 2 Practical Demonstration

In this section, we will explain step-by-step how the clustering approaches presented above can be executed on real tourism data by using the data science platform *RapidMiner*<sup>®</sup> ([www.rapidminer.com](http://www.rapidminer.com)).

### 2.1 *k*-Means Clustering

Considered the most prominent clustering task, customer segmentation is a concrete example that can be solved by *k*-means. Here, we used real customer data from a winter destination’s booking system as our dataset (cf. Table 1). The dataset contains customer information including guest’s age, first year of a guest’s arrival at the destination, number of past bookings, preferred duration of the trip, booking channel (i.e., 0 = web, 1 = phone), average cancellation rate, days between booking and arrival, price per booking, and the average number of booked products, rooms, ski passes, ski equipment, and ski school services per booking. The dataset consists of 5172 instances/customers (rows). Partitional clustering, like *k*-means, intends to divide the complete dataset into groups of customers that are as similar as possible in relation to the characteristics listed above.

A dataset typically requires certain steps of preprocessing in order to comply with the specific prerequisites of the selected data mining algorithm and to reach optimal and reliable results (Pyle, 1999; Tan et al., 2018). In the case of clustering, the first task is to select an appropriate subset of available attributes (cluster variables), which should serve as ideal characteristics to group similar examples (cases) together. Customer segmentation can be restricted, for example, to demographic characteristics or past booking behavior only (Dolnicar, 2021). In our case, however, we used all the attributes listed above. Additionally, an attribute’s data type must also be checked. Although the *k*-means algorithm is capable of handling any of the usual data types, like numeric or nominal attributes, it is recommended to transform nominal attributes into numeric dummy attributes as this transformation simplifies the visualization of the results (Pyle, 1999). Accordingly, we transformed the attribute *TripDuration* into  $TripDuration = Week$ ,  $TripDuration = ShortWeek$ , etc. with numerical values of 0/1. Another critical step regarding preprocessing is the normalization of the attributes’ value ranges (Everitt et al., 2011; Tan et al., 2018). As the similarity of the examples corresponds to the distance between the examples in an *n*-dimensional space, the size of the value range of an attribute determines its influence on the similarity calculation. In our case, for example, the

**Table 1** Dataset with customer data from a winter destination

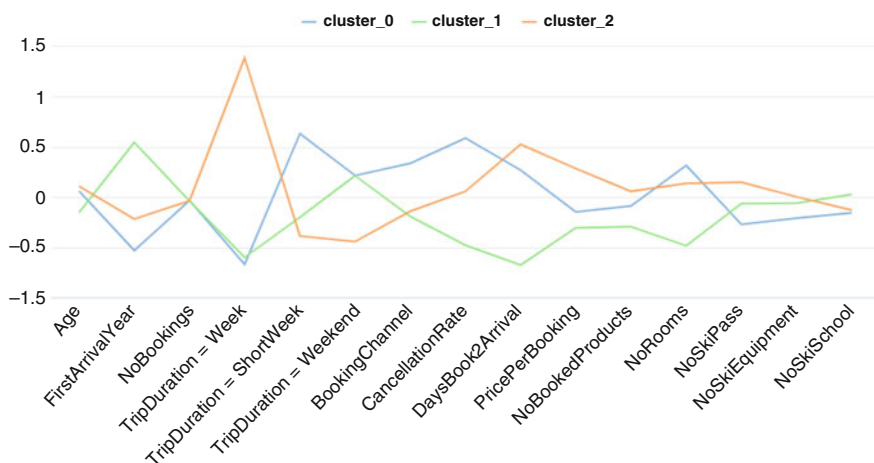
Row no.	Age	FirstArrivalYear	NoBookings	TripDuration	Booking Channel	Cancellation Rate	DaysBook 2/Arrival	PricePer Booking	NoBooked Products	NoRooms	NoSki Pass	NoSki Equipment	NoSki School
1	31	2005	5	Weekend	0	0	148,800	3281	1400	1	0	0	0
2	44	2004	4	N/A	0.500	0.500	273,250	152,500	2500	1	0	0	0
3	47	2004	10	N/A	1	0	247,100	57	1100	0.900	0	0.200	0
4	52	2004	29	N/A	1	0.034	192,207	37,931	1655	1.586	0	0	0
5	39	2005	2	Week	1	0.500	136	9170	9	0.500	1	3,500	3
6	54	2004	26	Week	1	0.269	242,269	0	1077	1	0	0	0

Source: Authors' illustration

booking price would have a much stronger influence on the calculation of a similarity measure than the cancellation rate. As this influential power is entirely accidental, we normalized all value domains via Z-score standardization, setting the average of each attribute to zero and the standard deviation to one (Everitt et al., 2011). In the final step of preprocessing, we eliminated outliers by removing all instances (i.e., cases) with attribute values outside the range of  $-4$  to  $4$  (in our case, 231 cases) since such values can be viewed as extreme values after having executed the Z-score standardization. Outliers have to be eliminated because the  $k$ -means algorithm is particularly sensitive to extreme values (ibid, 2011; Hair et al., 2014).

As the central part of our analysis, a  $k$ -means clustering can now be executed. First, since they are regarded as the most important parameters, the number of clusters  $k$  and a similarity measure, such as the *Euclidean distance*, the *Chebychev distance*, or the *cosine similarity* must be chosen (Tan et al., 2018). Optimizing these parameters should, on the one hand, be viewed from a mathematical perspective, that means, reaching optimal quality measures, like the within-cluster variation or the Davies Bouldin measure, which we calculate on the determined cluster model. On the other hand, the found clusters should be easily interpretable and make sense either from a business perspective or as input for consecutive steps of analysis, for example, as a dimension reduction technique or as input for a classification task. In our case, a  $k$ -means clustering with  $k = 3$  along with the similarity measure *cosine similarity* reached good results with an average within-cluster variation of 0.573 and a Davies Bouldin measure of 0.164 (both measures should be minimized when comparing different clustering solutions). The resulting cluster model is depicted in Fig. 6.

As can be seen, cluster 0 (1472 customers) represents older and long-standing customers, staying for a short week or weekend, booking mainly via phone, showing a high cancellation rate, booking quite far in advance, having a low overall booking price, and are booking mainly accommodation services (thus, labeled as “Older



**Fig. 6** Centroid plot  $k$ -means clustering (Source: Authors' illustration)

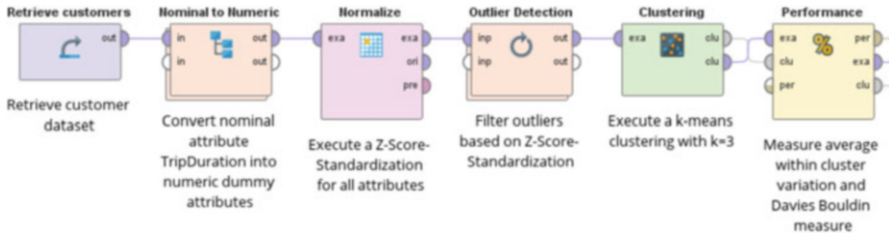


Fig. 7 Rapidminer process for *k*-means clustering (Source: Authors’ illustration)

*weekend customers*”). By contrast, cluster 2 (1538 customers) represents, again, older and long-standing customers, but this time staying for a full week, booking mainly via the Web far in advance with a high booking price, and booking mainly accommodation services as well as ski passes and ski equipment (thus, labeled as “*Older full-week skiing customers*”). Finally, cluster 1 (1931 customers) represents young and relatively new customers, staying over the weekend, booking mainly via the Web in a quite short-term/last-minute manner with a low cancellation rate and a low booking price, and mainly booking ski passes, ski equipment, and ski school services (thus, labeled as “*Young and spontaneous weekend skiing customers*”).

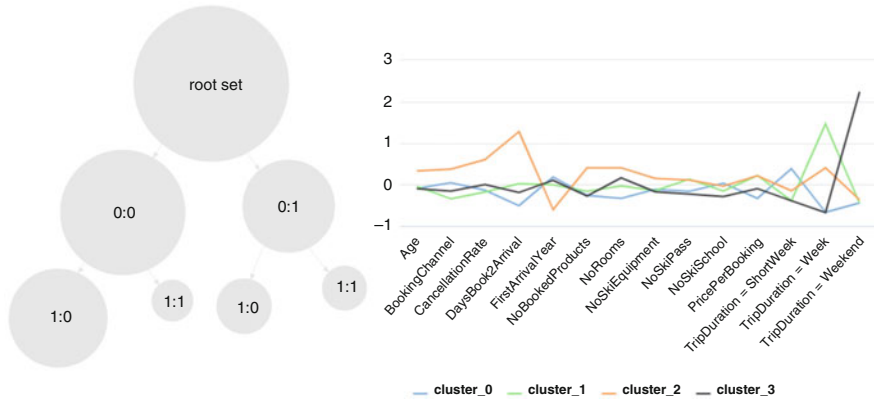
All the steps described above were executed using the data mining toolset *Rapidminer*. Figure 7 shows the overall analysis process consisting of a chain of operators, each receiving some input (e.g., the dataset), incorporating preprocessing or analysis steps, producing some output (e.g., the transformed dataset or a learned model), and passing the output on to the next operator. In this case, the first operator read the dataset, the second transformed the nominal attribute *TripDuration* into numeric flag attributes, the third normalized all the attributes, the fourth detected and deleted outliers, the fifth executed the *k*-means clustering, the sixth calculated the performance measures, and the seventh calculated the cluster centroids for all the clusters.

## 2.2 Hierarchical Clustering

As highlighted in the introductory section, in contrast to partitional clustering, hierarchical clustering successively divides (top-down) or groups (bottom-up) cases into clusters at different stages, resulting in a cluster hierarchy. Consequently, cases belong to a cluster on each stage or level of this hierarchy (Everitt et al., 2011; Tan et al., 2018).

### 2.2.1 Top-Down Clustering

In this subsection, a *top-down* clustering is applied to the dataset above (and the same preprocessing steps are executed). On the left, Fig. 8 shows the results of a

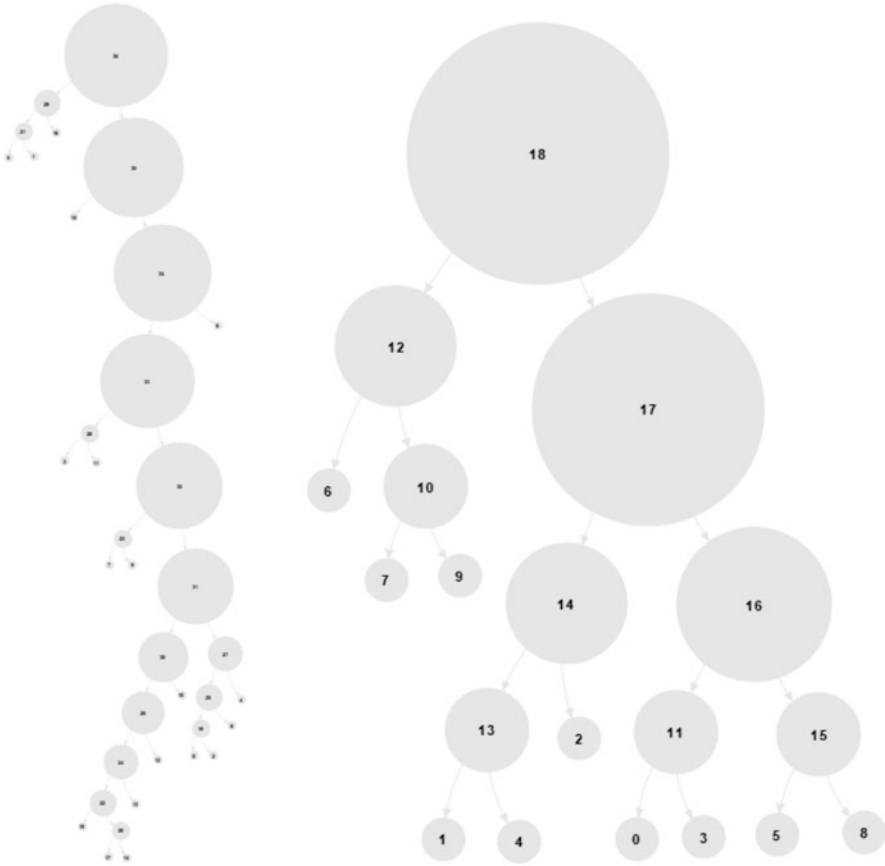


**Fig. 8** Top-down clustering: Cluster hierarchy (left) and cluster centroid plot (right) (Source: Authors' illustration)

top-down clustering for two levels, thereby illustrating the successful division of the complete dataset into clusters. The number of clusters created in each split and the number of levels of the hierarchy can be specified as parameters of the top-down clustering approach. The clustering itself is, once again, executed by a partitional clustering approach; in our case, a  $k$ -means clustering. Although each example belongs to a different cluster on each level of the hierarchy, all clusters on the same level constitute an exact partitioning. Figure 8 on the right shows a cluster centroid plot for the four clusters located on the lowest level of the hierarchy. Through this approach, hierarchical clustering allows for the thorough analysis of clusters at different granular levels. The *Rapidminer* process remained the same as in Fig. 7 apart from the  $k$ -means operator being replaced by the *top-down clustering* operator.

### 2.2.2 Agglomerative (Bottom-Up) Clustering

For the second hierarchical clustering approach, we now apply *agglomerative (bottom-up)* clustering to the dataset above (for demonstration purposes, 20 examples from the total dataset were randomly selected). Agglomerative clustering iteratively joins the two clusters that are most similar, where similarity is measured based on the two most similar cases of the two clusters (*single linkage*), the instances that are most dissimilar (*complete linkage*), or the cluster centroids (*average linkage*). On the left, Fig. 9 shows the cluster hierarchy for the similarity mode *single linkage*, while the right illustrates the one for *complete linkage*. As can easily be observed, the single linkage mode (left) tends to form slender clusters by

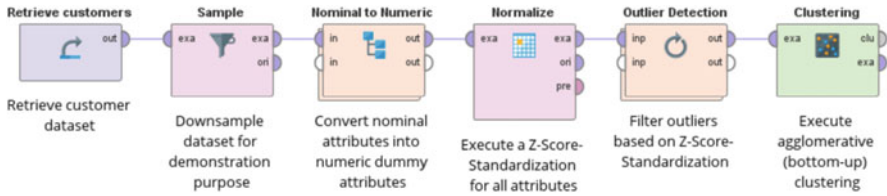


**Fig. 9** Agglomerative clustering with single linkage (left) and complete linkage (right) (Source: Authors' illustration)

successively adding single examples or small clusters to one main cluster, getting bigger and bigger. On the other hand, the complete linkage mode (right) creates a much more balanced cluster hierarchy and, thus, represents the preferred approach in our case.

Figure 10 shows the *Rapidminer* process for the agglomerative clustering approach. The operator *Sample* creates a subsample by randomly selecting 20 examples for demonstration purposes. All preprocessing steps were the same as in Fig. 7, although clustering is executed via the operator *Agglomerative Clustering*.



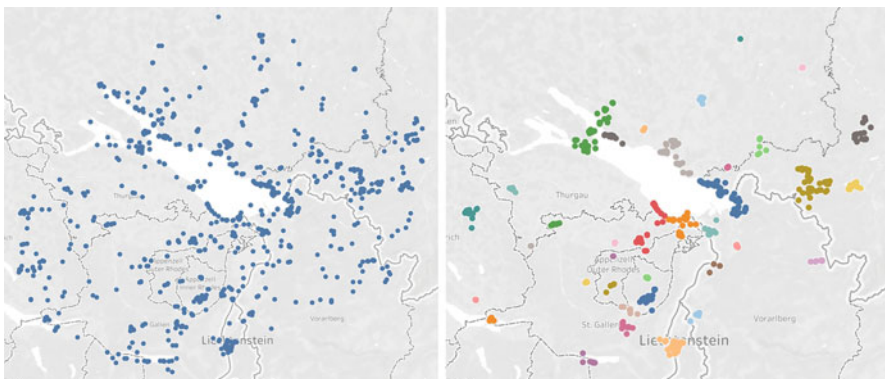


**Fig. 10** *Rapidminer* process for agglomerative (bottom-up) clustering (Source: Authors' illustration)

### 2.2.3 DBSCAN

As noted earlier, *DBSCAN* is a density-based spatial clustering algorithm. In this example, we will apply *DBSCAN* to geo-coded Flickr photo uploads collected for the region of Lake Constance, Germany. The dataset contains 4121 Flickr photo uploads specified by the latitude and longitude of the photo's geographic position. The intention of clustering is to group together photo uploads that are located quite close together in order to identify points of interest (POIs). In this specific case, no data preprocessing steps are necessary as both attributes have a numeric format and a normalized value domain. Additionally, *DBSCAN* identifies outliers automatically; thus, no separate outlier detection is required (Tan et al., 2018).

In contrast to *k*-means, the number of clusters is not specified by the user, but, rather, it is identified automatically. Instead, the user can define the minimal number of examples (*minPts*) that should exist in the direct neighborhood of a cluster member and the size, in other words, radius (*epsilon*), of the neighborhood. Often, optimizing these parameters is, unfortunately, not an easy task (Tan et al., 2018). In our case,  $\epsilon = 0.03$  (i.e., 3.3 km) and  $\text{minPts} = 13$  led to satisfying results. As a similarity measure, we used the *Euclidean distance*. Figure 11 shows the geo-coded Flickr photo uploads on the left and the clusters (i.e., the POIs) identified by the *DBSCAN* clustering algorithm on the right.



**Fig. 11** Flickr photo uploads (left) and *DBSCAN* clustering (right) (Source: Authors' illustration)

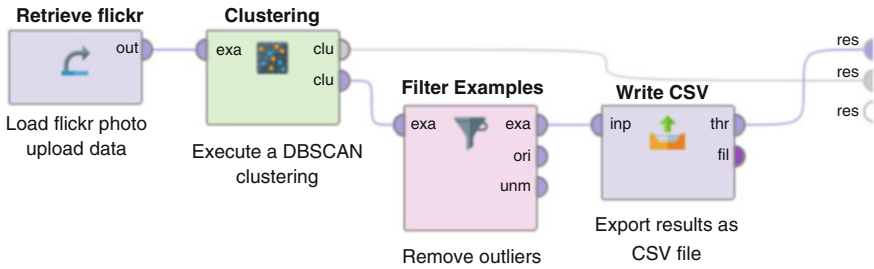


Fig. 12 Rapidminer process for *DBSCAN* clustering (Source: Authors' illustration)

Figure 12 shows the *Rapidminer* process for executing a *DBSCAN* clustering on Flickr photo uploads. The process, first, loaded the Flickr photo upload data and then, as with *DBSCAN* no preprocessing steps are necessary, directly executed the *DBSCAN* clustering. Outliers were automatically identified by the *DBSCAN* algorithm and removed from the resulting clustered dataset by means of the operator *Filter Examples*. Finally, the resulting dataset was stored as a CSV file and was used as input for tableau ([www.tableau.com](http://www.tableau.com)) in order to create the map-based visualizations shown in Fig. 11.

### 3 Research-Case

Traditional data sources, like guest surveys, visitor censuses, or on-site observations, impose a high amount of manual work and, thus, do not enable data gathering and analysis automatically and in real-time (Fuchs et al., 2014; Höpken et al., 2015; Önder et al., 2016). A study by Höpken et al. (2020) presents an approach that uses uploaded photos on the social media platform *Flickr* to analyze tourists' movement patterns when visiting points of interest (POIs), such as sights or attractions, in the destination city of Munich, Germany. By employing and comparatively assessing *DBSCAN* and *k*-means clustering for differing use scenarios, photo uploads on Flickr were clustered to POIs (Tan et al., 2018). Resulting POI visitation trajectories then served as input to analyze tourists' spatial behavior by association rule analysis and sequential pattern mining (ibid, 2020).

Data extraction was executed based on Flickr's application programming interface (API), such as *flickr.photo.search* to extract photo meta-data and *flickr.people.getInfo* to extract user information (e.g., user location). For each photo within the relevant geographic area, the following meta-data was extracted: photo id, owner id, owner name, latitude, longitude, location, date taken, and date uploaded. Following previous studies, users who continued to upload photos within an overall time period of more than 21 days as well as users who specified Munich as their home location were viewed as non-tourism users. Therefore, their photo uploads were removed

from the dataset. Data was extracted from Flickr for the year 2015, resulting in 13,545 photo uploads from 971 tourists (ibid, 2020).

Next, clustering was employed to aggregate Flickr photo uploads to POIs based on their physical location (i.e., geo-coordinates) (ibid, 2020). The two clustering algorithms, *DBSCAN* and *k-means*, were both evaluated concerning their suitability to identify meaningful clusters corresponding to relevant POIs. Compared to the *k-means* algorithm, *DBSCAN* is known to have the capacity to identify clusters of arbitrary shape without any restrictions; thus, there is no need to specify the number of clusters. Also, as *DBSCAN* identifies noise points explicitly, no outlier detection is necessary. Identified clusters were filtered based on *minimal popularity*, in other words, the number of tourists within a cluster (Hu et al., 2015). A cluster is considered popular if at least 2% of all the tourists involved with the photo uploads within the respective time period and area belong to the cluster.

First, a *k-nearest neighbor* distance plot showing the average distance of each point to its *k* nearest neighbors was employed to identify optimal *DBSCAN* parameter values with  $minPts = 3$  and  $\epsilon = 0,0009$  (i.e., 99 m), respectively (Höpken et al., 2020). Moreover, the number *k* of clusters found by *DBSCAN* was used for *k-means* to guarantee comparability of the two clustering approaches. The *DBSCAN* clustering model was characterized by one big cluster (containing 5909 photos), representing the city center of Munich and a high amount of relatively small and non-popular clusters (cf. 15 *DBSCAN* vs. 70 *k-means* popular clusters). In fact, *DBSCAN* grouped together all closely connected photo uploads, tending to generate large and often slender clusters (e.g., the cluster representing the city center of Munich). Put differently, *DBSCAN* was able to identify widespread and arbitrarily shaped clusters (not bound to hyper-ellipsoid or hyper-spherical clusters), which is an advantage in the case of identifying POIs on a larger geographic scale, for instance, for the *entire* urban region of Munich. On a smaller geographic scale, however, and in our case for the city center of Munich, *DBSCAN* identified all closely connected POIs as one single big cluster. In contrast, on such a small-scale granular level, like the city center of Munich, *k-means* clustering identified closely connected POIs correctly. This is mainly due to the fact that POIs in a city center environment tend to have a point-like form rather than a slender structure, constituting ideal conditions for the partitioning clustering approach of *k-means*, which requires the clusters to be of equal size and of hyper-ellipsoid or globular form (Liu, 2011). More concretely, while *DBSCAN* grouped all photo uploads of the central area into one big cluster, *k-means* identified 11 different clusters and, thus, correctly recognized corresponding POIs. One POI (Marienplatz) was separated into different clusters due to *k-means* well-known limitation of not being able to identify slender clusters properly (Höpken et al., 2020). Another POI (Feldherrenhalle) was merged with a neighboring POI (Odeonsplatz) due to *k-means* characteristic of trying to build clusters of similar size. In general, however, the results demonstrate that *k-means*' limitations do not substantially come into effect in regards to POIs in a city center environment as they mostly have a point-like structure and are typically of similar sizes. Overall, the assignment proves that clusters of Flickr photo uploads correspond to tourism-relevant POIs and, therefore, photo-sharing platforms like

Flickr can be constituted as a valuable source for analyzing tourists' POI visitation behavior.

In a final step, popular POIs identified through cluster analysis served as input for *association rule analysis* and *sequential pattern mining*. *Association rule analysis* aims at identifying which items or characteristics often “go together” within a dataset (Larose & Larose, 2014). Items, in the case of this study, correspond to clusters, or, POIs visited by tourists and an association rule  $X \rightarrow Y$  meaning that a tourist visiting POI  $X$  will often visit POI  $Y$  as well (Höpken et al., 2020). In contrast to association rule analysis, *sequential pattern mining* considers the temporal order of items within a transaction (Larose & Larose, 2014). Thus, while a frequent item set represents items co-occurring, a frequent sequence represents a specific order in which items often occur (ibid, 2014). To identify sequential patterns, the Generalized Sequential Pattern (GSP) algorithm was employed, while the FP-Growth algorithm was applied to find frequent item sets (Liu, 2011). An exemplarily strong rule found, for instance, that tourists visiting the POIs “Kaufhaus der Sinne” (206) and “Altes Rathaus” (178) would most likely also visit POI “Heilig-Geist-Kirche” (190). This rule is supported by 1.4% of all transactions and holds true for 100% of all transactions containing the antecedents 206 and 178. The particularly high lift of 16.09 means that, when visiting POIs 206 and 178, it is over 16 times more likely that a tourist will also visit POI 190 when compared to the average likelihood of visiting POI 190 (Höpken et al., 2020).

Finally, when comparing the two clustering approaches *DBSCAN* and *k-means*, it can be summarized that *DBSCAN* identified 15 popular clusters, leading to 45 frequent item sets, 60 association rules, and 370 frequent sequences, while *k-means* identified 70 popular clusters, leading to 534 frequent item sets, 1432 association rules, and 4760 frequent sequences (ibid, 2020). Sequential pattern mining identified frequent visitation sequences of short (1-h) and medium (4-days) duration with support between 0.6% and 1.7%, respectively. Figure 13 displays the most frequent tourist routes in the old town of Munich identified via the *k-means* method.

The proposed approach demonstrates its ability to analyze tourists' spatial behavior and movement patterns based on uploaded photo data from *Flickr*. Compared to traditional data gathering techniques, the approach offers the advantage of being fully automatic and, thus, executable in a real-time environment (Kolas et al., 2015). The identified POIs visitation behavior allows for more detailed explanations regarding the attractiveness of various POIs depending on visitor characteristics including gender or country of origin. Furthermore, it also determines visitation time as an important input for tourism planning and marketing activities (Höpken et al., 2020).



**Fig. 13** Frequent tourist routes in the old town of Munich identified via *k*-means (The authors thank Marcel Müller for Fig. 13 extracted from the MA thesis “*Big Data als Quelle für die Forschung im Tourismus unter Verwendung personenbezogener Geodaten von Fotos*” (2017, p. 51) supervised by Prof. W. Höpken, University of Applied Science Ravensburg-Weingarten, and co-examined by Prof. M. Fuchs.)

### Service Section

**Main Application Fields:** Cluster analysis is an unsupervised machine learning technique aiming to build groups of similar cases. Its most popular field of application lies in customer segmentation, for example, for customer relationship management or for the analysis of web usage behavior (e.g., as input for website adaptation, targeting, or recommender services). Additionally, cluster analysis can solve classification tasks if no pre-classified training data are available (e.g., segmenting financial behavior into benign and suspicious categories). In the area of text mining, cluster analysis is often used in the form of keyword clustering so as to find topics within a natural language text. Finally, cluster analysis can be applied within the field of network analysis in order to identify, for instance, groups of people closely connected on a social network.

**Limitations and Pitfalls:** As clustering is an unsupervised machine learning technique, no concrete definition of what is right and what is wrong exists. Consequently, there are no absolute quality metrics, such as accuracy in the case of classification; thus, one can only judge whether one clustering

(continued)

approach performs better or worse on a given dataset than another. Besides applying such a mathematical evaluation, a cluster model also has to be evaluated from a semantic or business perspective. This makes it difficult to judge whether a cluster model constitutes a meaningful and reliable result.

Additionally, each clustering approach has its own set of limitations, especially when it comes to the form of found clusters. On the one hand, *k*-means can only identify hyper-ellipsoid clusters, while, on the other hand, *DBSCAN* can identify clusters of any shape but tends to form long and slender clusters, which might be inappropriate in certain application domains. Furthermore, *k*-means requires the process of predefining the number of clusters; thus, an inappropriate cluster number might lead to inappropriate results. *DBSCAN*, contrarily, requires careful specification of the neighborhood size, which, in some application domains, may be difficult to define.

**Similar Methods and Methods to Combine with:** Numerous alternative clustering techniques have been invented over time and are used in certain application domains. One thereof, which is quite well-known, is a specific form of *artificial neural networks*, so-called *self-organizing maps* (SOMs), for example, *Kohonen networks* (Bloom, 2004). Moreover, *k*-medoids or *x*-means are used as specific extensions of the *k*-means algorithm. Finally, the *Louvain algorithm* for community detection is a method to extract clusters (communities) from large networks (Blondel et al., 2008).

In general, cluster analysis is used on its own as an unsupervised machine learning technique, for instance, in the case of customer segmentation. Additionally, cluster analysis can serve as a dimension reduction technique (similar to factor analysis) or as a reduction of the search space as input for a downstream analysis, such as a classification or an association rule analysis.

**Code:** The RapidMiner workflows are available at: <https://github.com/DataScience-in-Tourism/Chapter-8-Clustering>

## References

- Baggio, R., & Klobas, J. (2017). *Quantitative methods in tourism: A handbook* (2nd ed.). Chanel View Publications.
- Blondel, V., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10(P10008), 1–12.
- Bloom, J. (2004). Tourist market segmentation with linear and non-linear techniques. *Tourism Management*, 25(6), 723–733.
- Dietz, L. W., Sen, A., Roy, R., & Wörndl, W. (2020). Mining trips from location-based social networks for clustering travelers and destinations. *Journal of Information Technology and Tourism*, 22(1), 131–166.
- Dolnicar, S. (2021). Market segmentation for e-Tourism. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-tourism*. Springer Nature. [https://doi.org/10.1007/978-3-030-05324-6\\_53-1](https://doi.org/10.1007/978-3-030-05324-6_53-1)

- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, & U. Fayyad (Eds.), *Proceedings of the second international conference on knowledge discovery and data mining, KDD-96* (pp. 226–231). AAAI Press.
- Everitt, B., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster analysis* (5th ed.). Arnold Publishers.
- Fuchs, M., Höpken, W., & Lexhagen, M. (2014). Big data analytics for knowledge generation in tourism destinations: A case from Sweden. *Journal of Destination Marketing and Management*, 3(4), 198–209.
- Fuchs, M., Fossgard, K., Stensland, S., & Chekalina, T. (2021). Innovation and creativity in nature-based tourism: A critical reflection and empirical assessment. In V. Haukeland & P. Fredman (Eds.), *Nordic perspectives on nature-based tourism* (pp. 175–193). Edward Elgar Publishing.
- Hair, J. F., Black, B., Black, W. C., Babin, B. J., & Anderson, R. (2014). *Multivariate data analysis* (7th ed.). New International Edition, Pearson Education.
- Höpken, W., Fuchs, M., Keil, D., & Lexhagen, M. (2015). Business intelligence for cross-process knowledge extraction at tourism destinations. *Journal of Information Technology and Tourism*, 15(2), 101–130.
- Höpken, W., Müller, M., Fuchs, M., & Lexhagen, M. (2020). Flickr data for analyzing tourists' spatial behavior and movement patterns: A comparison of clustering techniques. *Journal of Hospitality and Tourism Technology*, 11(1), 69–82.
- Hu, Y., Gao, S., Janowicz, K., Yu, B., Li, W., & Prasad, S. (2015). Extracting and understanding urban areas of interest using geotagged photos. *Computers, Environment and Urban Systems*, 54, 240–254.
- Hudson, S., & Ritchie, B. (2002). Understanding the domestic market using Cluster Analysis: A case study of the marketing efforts of Travel Alberta. *Journal of Vacation Marketing*, 8(3), 263–276.
- Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., & Wu, A. (2002). An efficient *k*-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 881–892.
- Kolas, N., Höpken, W., Fuchs, M., & Lexhagen, M. (2015). Information gathering by ubiquitous services for CRM in tourism destinations: An explorative study from Sweden. In I. Tussyadiah & A. Inversini (Eds.), *Information and communication technologies in tourism* (pp. 73–85). Springer.
- Larose, D. T., & Larose, C. D. (2014). Discovering knowledge in data: An introduction to data mining, Chapter 10. In *Hierarchical & k-means clustering* (2nd ed., pp. 209–227). Wiley.
- Liu, B. (2011). Web data mining: Exploring hyperlinks, contents and usage data, Chapter 4. In *Unsupervised learning* (2nd ed., pp. 133–168). Springer.
- Lloyd, S. (1982). Least squares quantization in PCM. *Journal IEEE Transactions on Information Theory*, 28(2), 129–137.
- Neuburger, L., & Egger, R. (2020). Travel risk perception and travel behavior during the COVID-19 pandemic 2020: A case study of the DACH region. *Current Issues in Tourism*. <https://doi.org/10.1080/13683500.2020.1803807>
- Önder, I., Koerbitz, W., & Hubmann-Haidvogel, A. (2016). Tracing tourists by their digital footprints. *Journal of Travel Research*, 55(5), 566–573.
- Pitman, A., Zanker, M., Fuchs, M., & Lexhagen, M. (2010). Web usage mining in tourism: A query term analysis and clustering approach. In U. Gretzel, R. Law, & M. Fuchs (Eds.), *Information and communication technologies in tourism* (pp. 393–403). Springer.

- Pyle, D. (1999). *Data preparation for data mining*. Morgan Kaufmann Publisher.
- Scholochow, C., Fuchs, M., & Höpken, W. (2010). ICT-efficiency and effectiveness in the hotel sector: A three stage DEA approach. In U. Gretzel, R. Law, & M. Fuchs (Eds.), *Information and communication technologies in tourism* (pp. 13–24). Springer.
- Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2018). Introduction to data mining, Chapter 7. In *Cluster analysis: Basic concepts and algorithms* (2nd ed., pp. 525–612). Pearson Education.
- Ward, J. H., Jr. (1963). Hierarchical grouping to optimize an objective Function. *Journal of the American Statistical Association*, 58, 236–244.



# Dimensionality Reduction



## Overview, Technical Details, and Some Applications

Nikolay Oskolkov

### Learning Objectives

- Understand high-dimensional data
- Explain *The Curse of Dimensionality*
- Matrix factorization dimensionality reduction (PCA)
- Neighbor graph dimension reduction (tSNE and UMAP)

## 1 Introduction and Theoretical Foundations

When it comes to data analysis, situations dealing with high-dimensional data may arise often and can lead to numerous challenges (Clarke et al., 2008). This is very common with images and texts as well as biological and other types of data. The difficulty of working with high-dimensional data is mainly reflected through the poor performance of common statistical models due to violations in the fundamental assumptions of the models, a problem known as *The Curse of Dimensionality* (Altman & Krzywinski, 2018).

To define the concept of high-dimensional data, it is important to realize that, generally, any data can be characterized by the number of statistical observations (we will denote this parameter as  $n$ ) and the number of the data's descriptive features/variables (we will denote this parameter as  $p$ ). The latter can be thought of as any trait or attribute related to the phenomenon of a study, for example, pixels for image data, words for text, or genes for biological data. In contrast, the former

---

N. Oskolkov (✉)

Department of Biology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Lund University, Lund, Sweden

e-mail: [Nikolay.Oskolkov@biol.lu.se](mailto:Nikolay.Oskolkov@biol.lu.se); [nikolay.oskolkov@scilifelab.se](mailto:nikolay.oskolkov@scilifelab.se)

reflects the number of times that specific trait/attribute was observed in an experiment. The more noise and uncertainty there is throughout the process, the more statistical observations relating to the traits are needed in order to compensate for this. It appears that if  $n$  is not fixed and is too small, increasing  $p$  can lead to data sparsity, multicollinearity, and overfitting. In turn, this technically results in singularities and divergences in the mathematical equations underlying traditional statistical analysis (Altman & Krzywinski, 2018). In particular, for the inequality  $p \gg n$  (which reads as “ $p$  much greater than  $n$ ”; i.e.  $p$  is at least  $\sim 10$  times greater than  $n$ ), the assumption of normal distribution, which is typical for classical mathematical statistics, is no longer valid and may lead to misleading scientific conclusions. As previously mentioned, this problem is traditionally referred to as The Curse of Dimensionality (Altman & Krzywinski, 2018).

Finding a way to overcome The Curse of Dimensionality that occurs in high-dimensional data is one of the most important challenges the field of data science faces. Regularization techniques (Tibshirani, 1996) and dimensionality reduction (Li, 2010) are common approaches used to mitigate this problem. The latter is a regular standard of Exploratory Data Analysis (EDA), which is typically considered to be the first step in data analysis. In addition, dimensionality reduction not only allows for the preprocessing of high-dimensional data but also provides a valuable visualization of the data points, further shedding light on the data structure and potential meaningful patterns that can already be visible at this stage of analysis.

There are a number of different dimensionality reduction techniques at one’s disposal, and they are generally divided into linear and non-linear methods. The linear dimensionality reduction techniques are typically based on linear algebra, have a sound mathematical foundation, and incorporate a so-called matrix factorization (also known as matrix decomposition) approach (Stein-O’Brien et al., 2018). The idea behind this approach is to approximate a data matrix via a product of at least two other matrices, with one of them being a new representation of the initial data matrix but with a reduced dimension parameter  $p$ . Linear dimensionality reduction methods include techniques such as Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS), Singular Value Decomposition (SVD), Independent Component Analysis (ICA), and Non-negative Matrix Factorization (NMF),<sup>1</sup> amongst others (Stein-O’Brien et al., 2018). PCA is perhaps the most popular and well-defined linear dimensionality reduction technique provided in the literature.

## 1.1 PCA

The intention behind PCA is to reduce  $p$  features (or attributes) into a few latent factors and to preserve as much variation in the data as possible while constructing the approximation (Ringnér, 2008). Therefore, PCA is often explained as a

---

<sup>1</sup>Compare Chapter “Topic Modeling.”

procedure in which searching for orthogonal directions of maximal variation in the data is the main goal. These directions are mathematically proven to be equivalent to the eigenvectors in linear algebra and are found via the so-called eigenvalue decomposition problem (Stein-O'Brien et al., 2018). The eigenvalue decomposition leads to a certain fraction of variation in the data associated with each orthogonal principal component (PC). This is an important property of PCA as it allows for crude separation of the signals from the noise within a data set. Thus, PCA is often used as a noise reduction step in data preprocessing and is especially useful when applied to data with correlated features/attributes. The correlated data structure is what gives the algorithm the possibility to collapse multiple non-independent features/attributes into a few independent/orthogonal principal components (PCs). Furthermore, the PCs can be used to replace the original high-dimensional data ( $p$  dimensions) with data with reduced dimensions ( $k$  dimensions;  $k < p$ ), fulfilling the assumptions from traditional frequentist statistics ( $n > k$ ). For these reasons, PCA can be considered a very useful data preprocessing step.

In practice, algorithmically, PCA is often computed via Singular Value Decomposition (SVD), which is numerically much faster to solve than the canonical eigenvalue decomposition problem. The linearity of PCA comes from the fact that the procedure uses only affine and mathematically linear data transformations, such as rotation, shift, flip, etc. While the PCA problem can be solved analytically, Multi-Dimensional Scaling (MDS), on the other hand, represents a machine learning interpretation of PCA that explicitly applies the numeric optimization of a cost function (i.e. the mean square error (MSE) between the initial data and its factorized approximation) (Mead, 1992). The particular choice of MSE cost function implies that MDS essentially assumes normal distribution of the data and, hence, belongs to the family of linear dimensionality reduction techniques. Although they are algorithmically very different, PCA and MDS are conceptually identical, and the terms are often used interchangeably (Udell et al., 2016).

Despite their simplicity and interpretability, linear dimensionality reduction techniques may not be very optimal in reflecting non-linear structures of some types of high-dimensional data, such as images, texts, or single cell gene expression. For these data, other non-linear dimensionality reduction techniques have been developed (van der Maaten & Hinton, 2008). These techniques have very different mathematical assumptions when compared to linear dimensionality reduction methods and are based on converting the data into a neighbor graph and then laying out the graph. Since graphs represent non-parametric, non-linear mathematical objects, this family of methods performs non-linear dimensionality reduction. Multiple comparisons of linear vs. non-linear dimension reduction methods for, for example, images of hand-written digits (MNIST) and Google News word vectors (text data), can be found in McInnes et al. (2018).

## 1.2 tSNE

T-distributed Stochastic Neighbor Embedding (tSNE) was developed by Laurens van der Maaten and Geoffrey Hinton in 2008 (van der Maaten & Hinton, 2008) and, since then, has become an increasingly popular and widely-used technique across different research areas, varying from astronomy and molecular cell biology to marketing and management. In particular, tSNE has recently been embraced as a popular technique to visualize tourist trends (Li et al., 2020), landscape characteristics (Payntar et al., 2021), and a region's points of interest (POIs) (Liu et al., 2020), amongst others.

The idea behind tSNE is to convert original high-dimensional data into a graph object by computing pairwise probabilities of observing points at a certain distance from each other. The pairwise probabilities in the high-dimensional space (original data) have a normalized Gaussian form and can be viewed as the graph edges' weights. The Gaussian function explicitly contains a bandwidth that determines how far the data points can affect each other, a concept known in theoretical physics as the "screening radius." The "screening radius" defines a very important tSNE hyperparameter, known as *perplexity*. Simply put, perplexity can be referred to as a typical number of nearest neighbors in contact with each other. This assumes that the data points outside of the perplexity radius do not strongly influence each other and do not contribute to the pairwise "interactions." As one can see, this idea regarding tSNE, along with many others, was clearly inspired by formalisms from theoretical physics.

As the next step, tSNE builds pairwise probabilities of observing data points at a certain distance in a lower dimensional space. For this purpose, tSNE uses Student's t-distribution. Thanks to its "heavy tails," the particular elevated shape of the function at large distances between data points, the distribution has its advantages when compared with the Gaussian probability density function. This facilitates the data points that are distant in the high-dimensional space to be pushed farther apart in the low-dimensional space in order to avoid crowding the data points and, in turn, to prevent unsatisfactory visualization. Thereafter, the previously constructed high-dimensional graph can be laid out on a lower dimensional space by minimizing a special cost function called Kullback-Leibler divergence. The Kullback-Leibler divergence simply tries to make the high-dimensional probabilities as similar to the low-dimensional probabilities as possible. It is numerically optimized through the gradient descent algorithm, which is typical for data science and machine learning. Generally speaking, tSNE tries to embed the high-dimensional graph into a low-dimensional space by preserving pairwise distances between the data points' nearest neighbors to the largest possible extent.

The preservation of distances between only the nearest neighbors (where perplexity also comes into play) implies that tSNE preserves so-called *local structure* of the data. This differs largely from PCA and MDS since they try to preserve pairwise distances between all data points or the so-called *global data structure*. The local data structure preservation implemented by tSNE has substantial advantages when

working with particular non-linear types of data, such as the Swiss Roll (Yin, 2007), where concentrating on preserving pairwise links between closest neighbors can be more informative than attempting to keep all pairwise connections between the data points. However, besides the benefits of a well-reduced representation of non-linear data, local structure preservation also has its disadvantages. For example, when data points form distinct clusters on a tSNE plot, the sizes of the clusters as well as the distances between them and their mutual positions are not necessarily meaningful and should be interpreted with caution. This means that tSNE can identify the presence of the clusters (local structure) but does not provide information about any hierarchical relationship between the clusters (global structure). Take Li et al. (2018) study as an example; although visualizing the process of sentimental classification of tourism online reviews is possible, the results do not mirror the hierarchical structure of documents and can only be interpreted at the word level. This is because tSNE's design ensures the preservation of short distances between points within clusters but is unable to preserve long distances between points belonging to different clusters.

It is also important to mention that, due to its technical algorithmic limitations, tSNE, in practice, can only embed high-dimensional data into 2 or 3 dimensions. Thus, this restricts tSNE from being a general-purpose dimensionality reduction technique and, instead, it is often used purely for data visualization purposes. Furthermore, tSNE usually experiences problems when working with pure high-dimensional data and needs the data to be preprocessed in such a way that the number of noisy features/attributes are reduced. Therefore, one often uses PCA as a denoising step prior to applying tSNE, making it a so-called pre-dimensionality reduction.

Finally, tSNE has a few important hyperparameters to take into account, such as perplexity, the initial number of PCA dimensions to be inserted into tSNE, learning rate, the maximum number of tSNE iterations, and the initial coordinates of the low-dimensional embeddings. Generally, there are no strict rules on how to optimize tSNE's hyperparameters, and they depend largely on the data set. In addition, since dimensionality reduction is an unsupervised problem, one cannot apply cross-validation, which is otherwise the standard way of tuning hyperparameters for supervised machine learning. Nonetheless, there are some empirical rules of thumb (provided in the practical section of this chapter), which can at least be used to initially guess the hyperparameter values and, at a later point in time, manually tweak them further for a better tSNE picture.

### 1.3 UMAP

Uniform Manifold Approximation and Projection (UMAP) dimensionality reduction was developed by Leland McInnes, John Healy, and James Melville in 2018 (McInnes et al., 2018) and immediately prevailed in many areas, including Natural Language Processing (NLP) and single cell biology. Although tSNE and UMAP are

similar in many aspects, the latter offers an improved version of the tSNE algorithm, making it an alternative popular dimensionality reduction tool.

In a nutshell, UMAP utilizes the same neighbor graph projection idea as tSNE. However, omitting normalization of the high- and low-dimensional probabilities led to an increase in the speed of the algorithm. In addition, the Kulback-Leibler cost function was generalized to a cross-entropy, which presumably allowed UMAP to preserve more of the global data structure than tSNE (Becht et al., 2018). Furthermore, in contrast to tSNE's 2–3 dimensions, UMAP is capable of embedding data into a number of latent components; thus, UMAP can be used not only for visualization purposes but also as a general-purpose machine learning dimensionality reduction technique for, for instance, data preprocessing (similar to PCA). Also, unlike tSNE, UMAP's *local connectivity* feature seems to warrant a better capacity for working directly with high-dimensional data without the pre-dimensionality reduction step. Nevertheless, as this feature of UMAP has yet to be fully explored and understood, in practice, it is still recommended to denoise high-dimensional data with, for example, PCA, before inserting any input into UMAP. Lastly, despite the fact that random initialization was seen as a default and, arguably, an essential part of the tSNE algorithm, UMAP implemented non-random initialization with Laplacian Eigenmap, further contributing to the improved global structure preservation.

Understanding the role of UMAP in the context of other dimensionality reduction methods continues to be an active area of research. Yet, although it seems that a better global structure preservation has been established for UMAP, and at least for single cell biology (Becht et al., 2018), alternative reports suggest this might not be the case for other types of data (Kobak & Linderman, 2019).

In the final part of this section, relations between dimensionality reduction and clustering, another unsupervised machine learning technique sometimes used in addition to dimensionality reduction, will be briefly discussed. While dimensionality reduction is simply a way of compressing data without any attempt to assign individual data points to identified patterns (clusters), the goal of cluster analysis is to divide data points into groups based on their similarities. Such pairwise similarities between data points are often established via Euclidean distance (e.g. for k-means or hierarchical clustering), and therefore they may behave inadequately for high-dimensional data due to The Curse of Dimensionality (Aggarwal et al., 2001). Therefore, for true high-dimensional data, no meaningful clustering of data points can be obtained without preprocessing/transforming the data or regularizing clustering algorithms in advance. Dimensionality reduction is a suitable data preprocessing procedure that, at least in theory, can mitigate the negative effects of The Curse of Dimensionality. Hence, when working with high-dimensional data, it is advisable to perform dimensionality reduction prior to clustering. However, it would be naive to perform clustering on 2–3 dimensions of tSNE, PCA, or UMAP as the intrinsic dimensionality of high-dimensional data might not be 2 or 3. Therefore, an appropriate step would be to replace the original high-dimensional data with a number of latent variables (e.g. PCA or UMAP components) and run cluster analysis in the new space with reduced dimensions. Finding the sufficient number of components for a robust cluster analysis depends on the particular data set; nonetheless, a

general principal for estimating the number of reduced dimensions for a cluster algorithm is based on randomization of the data, which disentangles and removes noisy components from the signal-containing components.

## 2 Practical Demonstration

In this section, a step-by-step tutorial on how to apply different dimensionality reduction techniques (PCA, MDS, tSNE, and UMAP) to a real-world tourism-related text data set will be demonstrated. The data set was collected by scraping more than 100,000 Instagram images of Austria posted by tourists (Arefieva et al., 2021). The images were annotated via Google Cloud Vision in order to summarize tourists' impressions about Austria, and a Doc2Vec model, which was trained on a large tourism-related corpus (see chapter "Text Representations and Word Embeddings"), was used to convert the text annotations into numeric representations by building 100 numeric vectors that served as the new features/attributes in place of the raw text. Here, the 100 numeric vectors corresponding to the 100,000+ Instagram images, or more precisely, image annotations, will be used in order to visualize the data with different dimensionality reduction techniques. For this purpose, R language for statistical programming will be used (R Core Team, 2019).

The first step is to read the tab-delimited data set ("read.delim" function in base R), and in order to speed up computations, randomly select 10,000 images/text annotations instead of using the full data set. The "sample" function below extracts 10,000 random rows.

```
data <- read.delim("Vectors.tsv", header = FALSE, sep = "\t")
data <- data[sample(nrow(data), 10000), ]
```

Next, start making a PCA plot. For this, use the "prcomp" function in base R that reads the data as input. It is generally recommended to perform logarithm transformation of the data before applying "prcomp" in order to make the data more normally distributed. Before computing PCA, one important action to take is to center the data. This is, however, performed by default in "prcomp"; again, it is merely to ensure that the data looks as normally distributed as possible. Since the data contains negative values, due to its preparation via the Doc2Vec model, a logarithm function cannot be applied. Therefore, an offset, which is the absolute value of the minimal negative value of the matrix, must be added to each value of the data matrix. The offset does not change the patterns present in the data since PCA is invariant to affine transformations, including any shifts caused by a certain offset value.

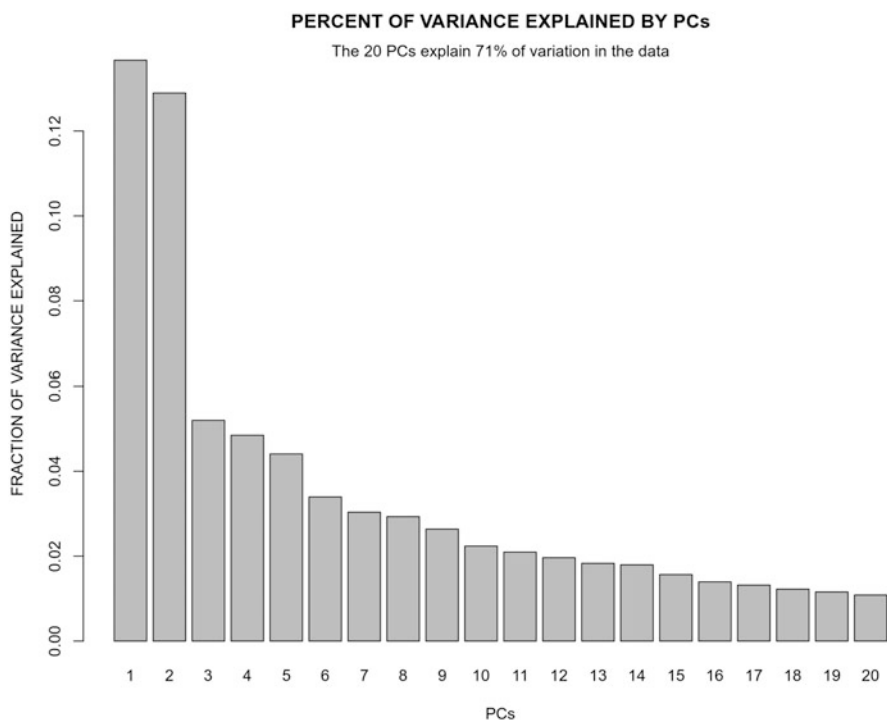
```
PC <- prcomp(log10(data + abs(min(data)) + 1))
```

For exploratory analysis of the data, it is very informative to check the fraction of variation explained by each principal component (PC). This figure is particularly

useful in helping to understand how much signal there is in the data. The information regarding variance explained is contained in the “sdev” (standard deviation) variable of the “PC” object, which contains the results from “prcomp.” The fraction of the variance explained by each principal component (PC) can be computed by dividing the squared “sdev” (remember, variance is a squared standard deviation) for each principal component by the total sum of the squared “sdev” from all components. Here, the “barplot” function in base R can be used to visualize the variance explained by the 20 leading principal components.

```
vars <- PC$sdev^2
vars <- vars / sum(vars)
barplot(vars[1:20], names.arg = 1:20, xlab = "PCs", ylab = "FRACTION OF
VARIANCE EXPLAINED", main = "PERCENT OF VARIANCE EXPLAINED BY PCs")
mtext(paste0("The 20 PCs explain ", round(sum(vars[1:20])*100,0), "%
of variation in the data"))
```

In Fig. 1, one can observe that the first two principal components are responsible for a much higher fraction of variation in the data compared to the rest of the components. This is a good sign as it demonstrates that the features in the data correlate to some extent and can be collapsed into a few principal components, thus

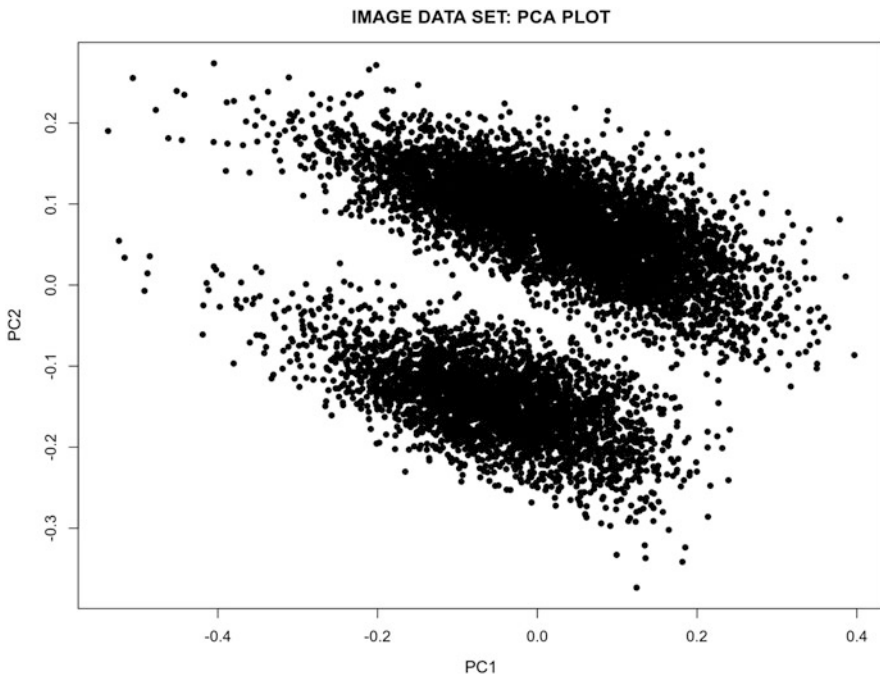


**Fig. 1** Variance explained by leading principal components



reducing the overall dimensions of the data set. When the variance explained figure has a somewhat flat, rather than decreasing, profile, this implies that all principal components (PCs) contain an approximately equal amount of variation. Usually, this is the case when all the features/attributes in the data are nearly independent. In other words, there is no redundancy or correlation structure present in the data, and the dimensions of the data set cannot be reduced any further from the original ones. This tends to occur often with genomic data (Altman & Krzywinski, 2018), implying that PCA might not be particularly useful for reducing dimensions in this research area; as a result, leading PCs typically explain very little variation in genomic data. However, this is not the case for the data set analyzed here.

It can be concluded that the variance explained by the top 20 principal components is as much as 71%, which is quite a substantial fraction of variation maintained in the data, even after reducing dimensions from 100 to 20. The criterion of percentage of variation explained by top N principal components (PCs) is often used to determine how many PCs should be selected to represent the original data set without losing too much information/variation. Later in this section, the top 20 principal components will be used in place of the original 100 vectors as new data, denoised by PCA, to input into tSNE and UMAP. This action is typically important since tSNE has difficulty working directly with pure unprocessed high-dimensional data. Finally, a PCA plot using only the first two leading principal components will be illustrated, Fig. 2.



**Fig. 2** PCA dimensionality reduction plot of the annotated Instagram image data set

```
plot(PC$x[,1:2], main = "IMAGE DATA SET: PCA PLOT", xlab = "PC1", ylab =
"PC2", col = "black", cex = 0.8, pch = 19)
```

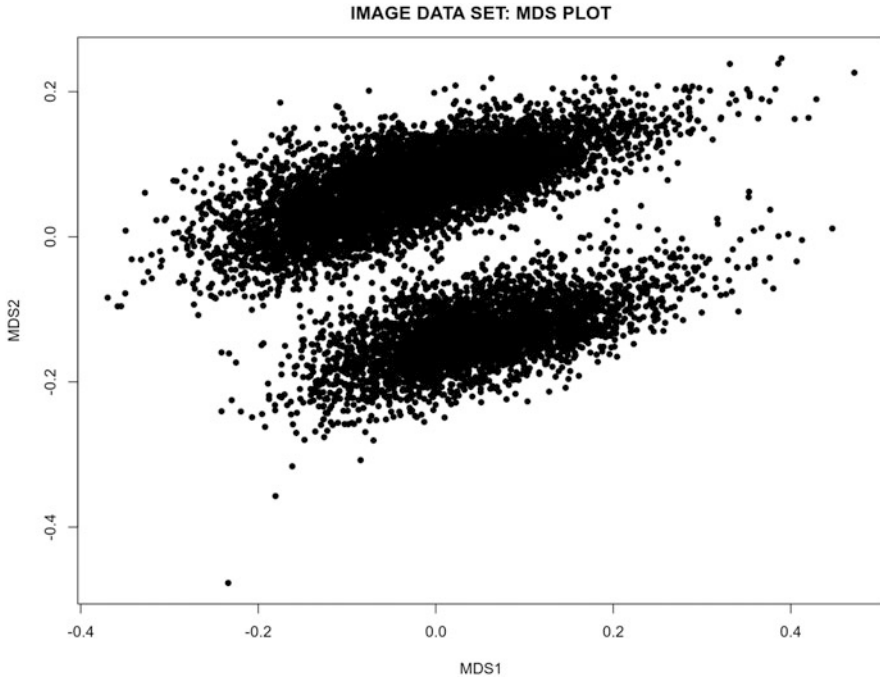
Two distinct clusters of points can be observed, which means that the data has some structure; nonetheless, the interpretation of the clusters remains unclear at this stage. An alternative (unfortunate) outcome could have involved the data points forming a big round unstructured shape in the center of the figure. In that case, it would have implied that all directions were nearly equivalent, and little potential would have been available for dimensionality reduction. Fortunately, however, our data set seems quite promising.

For comparison, let us compute MDS and demonstrate that PCA and MDS produce virtually identical outputs. For this purpose, the “cmdscale” function in base R will be used. This function accepts a matrix of pairwise distances between data points as input. Such matrix can be computed using the “dist” R function, which, by default, applies Euclidean distance; however, one can also choose Minkowski, Canberra, Manhattan, or other distances more suitable for data that is not normally distributed. Keep in mind that the “dist” function will require a lot of RAM; hence the MDS method is extremely memory-expensive, as well as problematic when it comes to computing large data sets with thousands of data points. In addition, the numeric optimization of the MDS cost function is a very time-consuming process, requiring much patience. If you wish to plot your dimensionality reduction faster, switching to PCA might be a better option. Below, the code for MDS will be presented, and the data structure produced by this dimensionality reduction technique will be discussed in more detail.

```
d <- dist(log10(data + abs(min(data)) + 1))
mds <- cmdscale(d, k = 2)
plot(mds[, 1:2], main = "IMAGE DATA SET: MDS PLOT", xlab = "MDS1", ylab =
"MDS2", col = "black", cex = 0.8, pch = 19)
```

One can observe that the end result of the PCA plot in Fig. 2 looks nearly indistinguishable from the MDS plot in Fig. 3, despite the algorithmic differences, memory usage, and computation time. Once again, two major clusters, which at this point are still unavailable for interpretation, can be seen.

Next, a comparison between the PCA and MDS figures and a tSNE plot will be made. To run tSNE in R, the “Rtsne” library, which implements the Barnes-Hut version of the algorithm, will be used (van der Maaten, 2013). This version optimizes time and memory consumption by ignoring distant data point neighbors and focusing only on computing pairwise distances between close neighbors. The “Rtsne” function incorporates a few of the algorithm’s important hyperparameters, such as perplexity, the initial number of PCA dimensions to feed into tSNE, and the maximum number of tSNE iterations. Since no good versions of hyperparameter tuning seem to exist for tSNE, and cross-validation for unsupervised machine learning problems is also not an option, a few empirical rules of thumb that might



**Fig. 3** MDS dimensionality reduction plot of the annotated Instagram image data set

be useful when guessing appropriate values for tSNE hyperparameters will be discussed below.

Since perplexity has a meaning of the number of  $k$ -nearest neighbors (KNN) for each data point, one can take some advice from the KNN classification machine learning process. According to a rule of thumb in this area, an optimal  $k$  for a KNN classifier can be guessed based on the square root of the number of data points. This notion has deep roots in the mathematics of Brownian diffusion from physics; in other words, an example would be when a random traveling agent deviates from its start by the square root of the number of steps. For our problem, with 10,000 data points, an optimal perplexity of 100 will be selected. Note that large data sets require larger optimal perplexity values.

As mentioned earlier in the theoretical introduction section, tSNE typically does not work well with pure high-dimensional data, and, therefore, it is recommended to use PCA as a pre-dimensionality reduction (or denoising) step in order to reduce the number of noisy features/attributes beforehand. The idea of the pre-dimensionality reduction step with PCA is to replace the raw data with a number of leading principal components, thus getting rid of the “long tail” in the plot relating to the fraction of explained variance by each principal component. The “long tail” is assumed to contain redundant and less important variables that can be omitted without surrendering too much information. The Heuristic Elbow method (see PCAtools

R/Bioconductor library in the reference list) can serve as a rule to aid in determining the optimal number of principal components to feed into tSNE. Another possible approach involves comparing observed variation (computed via PCA) in the original data against the variation of the corresponding randomized/shuffled data. The randomization allows for the background noise level in the data to be inferred. Therefore, the optimal number of principal components (PCs) to keep for downstream analysis is the number of PCs that explain the amount of variance above the “by chance” level. In our example, and for the sake of simplicity, the top 20 principal components were selected to input into tSNE since they explain 71% of the variation, and the profile in Fig. 1 seems to saturate at around 20 PCs.

The maximum number of tSNE iterations typically lies between 300 and 1000. However, this parameter depends on the number of data points and should be increased for large data sets, otherwise the gradient descent optimization will have little time to converge. One way to ensure that the gradient descent has converged is to look at the x- and y-axes of the tSNE plot. The range of values on the axes should be approximately 50–100; thus, as an example, if the range 10 is provided, then this is an indication that tSNE failed to converge.

Although at its beginning stages, random initialization was considered an essential part of the tSNE algorithm (van der Maaten & Hinton, 2008), recent developments have helped improve this option, and, nowadays, a popular way to initialize the low-dimensional embeddings is via PCA. PCA initialization of low-dimensional tSNE embeddings might be important for a better preservation of the global structure of the data, which, in turn, might lead to more correct distances between clusters as well.

To develop a better intuition in regards to tSNE hyperparameters, a good option would be to import the data into online tools, such as the Tensorboard Embedding Projector (<https://projector.tensorflow.org/>), and visually inspect the effects of different tSNE hyperparameters.

Now, the next steps will involve specifying the three tSNE hyperparameters, running tSNE, and plotting a 2D representation of the annotated image data set.

```
library("Rtsne")
set.seed(12)
optPerp <- round(sqrt(dim(data)[1]), 0)
tsne.out <- Rtsne(log10(data + abs(min(data)) + 1), initial_dims =
20, verbose = TRUE, perplexity = optPerp, max_iter = 1000, Y_init = PC$x[,1:
2])
```

In the code above, an optimal perplexity as the square root of the number of data points was defined first. Thereafter, the “Rtsne” function specifying the optimal perplexity, the number of principal components (`initial_dims = 20`), the maximal number of iterations (`max_iter = 1000`), and the tSNE initialization via PCA (`Y_init = PC$x[,1:2]`) was applied. The previously computed top 2 PCs were used in this case. Finally, the 2D embeddings constructed by tSNE were plotted using the “plot” function from base R.

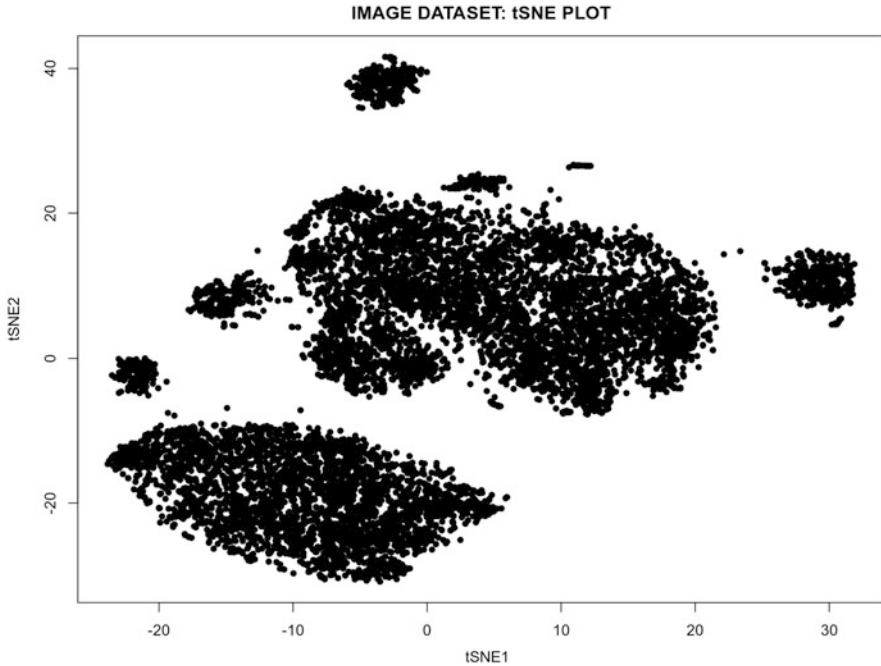


Fig. 4 tSNE dimensionality reduction plot of the annotated Instagram image data set

Looking at the tSNE plot in Fig. 4, again, two large clusters similar to the PCA plot can be observed. However, at least four additional smaller clusters, not previously identified in Figs. 2 and 3, are also visible. This shows that tSNE can demonstrate a finer heterogeneity in the data in comparison to what PCA can achieve. Yet, as emphasized in the theoretical introduction section, one should be cautious when interpreting the distances between observed clusters on a tSNE plot as they do not always guarantee correct preservation.

Next, tSNE clusters will be compared to the clusters produced by UMAP, and a discussion of UMAP hyperparameters will follow. Despite most of the hyperparameters being similar to those from tSNE, UMAP has a few additional specific ones. To compute and run UMAP, an efficient “uwot” R library, developed by James Melville (Melville et al., 2020), will be used (Fig. 5).

```
library("uwot")
set.seed(123)
optPerp <- round(sqrt(dim(data)[1]), 0)
umap.out <- umap(log10(data + abs(min(data)) + 1), n_neighbors =
optPerp, pca = 20, min_dist = 0.3, metric = "euclidean", init = "pca",
verbose = TRUE, n_threads = 4)
```

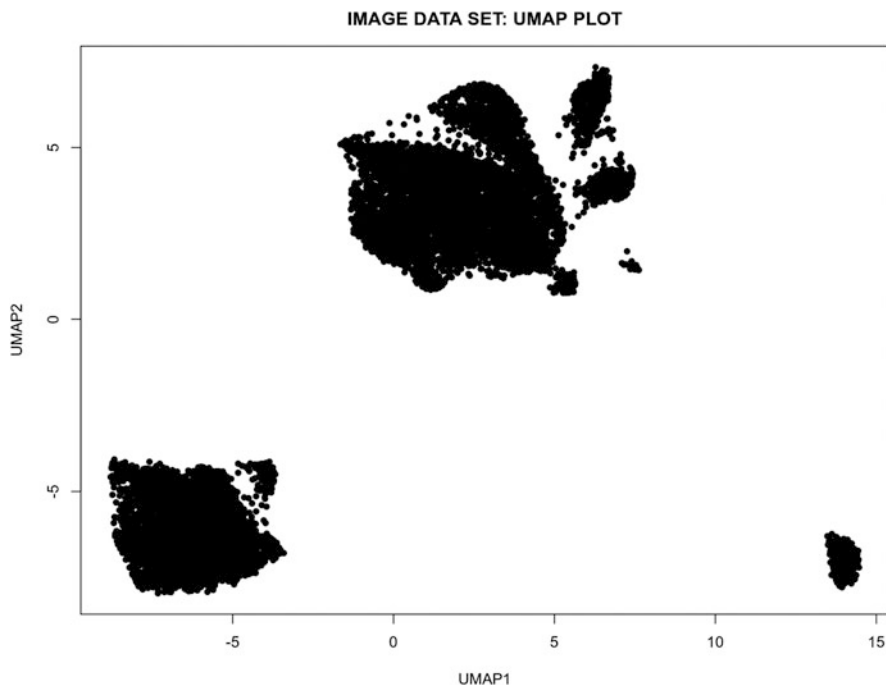


Fig. 5 UMAP dimensionality reduction plot of the annotated Instagram image data set

In the code depicted, the “`n_neighbors`” hyperparameter is nearly equivalent to perplexity in tSNE; therefore, likewise as with the tSNE process above, it can also be specified as the square root of the number of data points. In contrast, “`min_dist`” is a hyperparameter specific to UMAP that has no counterpart in the tSNE algorithm. In layman terms, one can explain this hyperparameter as a measure of the clusters’ density. Essentially, low “`min_dist`” values signify that UMAP assigns almost identical coordinates to the data points that are close to each other in the low-dimensional space. This efficiently leads to a more densely packed (low “`min_dist`”) or more spread-apart (high “`min_dist`”) display of the clusters. Here, “`min_dist = 0.3`” will be used, providing moderately densely packed clusters. Decreasing the value down to 0.1–0.01 will produce very tightly-packed shapes/forms that, however, might be more advantageous for running clustering algorithms on the top of UMAP.

As was the case for tSNE, the original data will be replaced with the top 20 principal components as a denoising step, regulated by the “`pca`” hyperparameter. Another interesting advantage of UMAP is that it is more robust when it comes to different types of data, including binary, categorical, and continuous data. While, for tSNE, one has to pre-compute a matrix of pairwise distances between data points in order to use it with, for example, binary data, UMAP can complete this step automatically by specifying an appropriate “`metric`” hyperparameter. In our case,

“metric = euclidean” will be used; but Hamming, Dice, and other metrics are also available, making UMAP more flexible and generalizable for different types of data, such as binary or categorical (Shirkhorshidi et al., 2015). Moreover, the “uwot” library offers multi-threading, an option that tSNE still lacks, in which four parallel threads with “n\_threads = 4” is utilized here.

Similar to tSNE, two large clusters and a few smaller clusters can be noted. One can immediately recognize that the UMAP clusters appear to be denser than those from tSNE. This, however, can be regulated via the “min\_dist” hyperparameter. Furthermore, the inter-cluster distances also seem to be larger than the tSNE ones. This feature can be useful if one aims to run clustering algorithms on reduced UMAP dimensions. In this case, clustering algorithms such as k-means, HDBSCAN, or Louvain (Blondel et al., 2008) do an excellent job as the UMAP clusters are very distinct and easily distinguishable from each other. On the other hand, it is typically not recommended to cluster on 2D tSNE or UMAP representations if the intrinsic dimensionality of the data is not 2. This may lead to the loss of interesting data patterns or, even worse, to biased clustering results. In contrast to tSNE, which delivers only 2–3 low-dimensional components due to the algorithm’s computational limitations, UMAP can provide several components. In this way, clustering on a number of UMAP components may be more promising than clustering on raw data that might face The Curse of Dimensionality and be sensitive to the choice of appropriate distance metric.

### Service Section

**Main Application Fields:** Dimensionality reduction is typically used for Exploratory Data Analysis (EDA) as a data preprocessing and visualization step. One of the major goals of dimensionality reduction is overcoming The Curse of Dimensionality, a typical problem when working with high-dimensional data. Linear dimensionality reduction techniques such as PCA and MDS are considered general-purpose methods and can be used to reduce data complexity and noise as well as discover hidden structures in the data. For certain types of data, linear dimensionality reduction techniques are unable to provide enough resolution and are thus combined with other non-linear techniques, such as tSNE and UMAP. The non-linear dimensionality reduction techniques can lead to finer heterogeneity of the data, which, in turn, is more informative when searching for hidden structures in the data.

**Limitations and Pitfalls:** Dimensionality reduction techniques rely on the choice of a distance metric, which is often hard to select properly for real-world data. Non-linear dimensionality reduction techniques such as tSNE and UMAP heavily depend on hyperparameter tuning and often provide misleading results when not used appropriately. In addition, interpretation of the structures discovered by non-linear dimensionality reduction can be problematic due to the non-parametric nature of the methods.

(continued)

**Similar Methods and Methods to Combine with:** Dimensionality reduction is often combined with cluster analysis. Both represent unsupervised machine learning techniques widely used for data exploration. Other linear dimensionality reduction methods are as follows: Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), and non-metric MDS. Non-linear dimensionality reduction methods include Isomaps, Locally Linear Embedding (LLE), and LargeVis.

**Code:** The code in R is available at: <https://github.com/DataScience-in-Tourism/Chapter-9-Dimensionality-Reduction>

## Further Readings and Other Sources

Leland McInnes talk at PyData meeting, New York 2018 on dimensionality reduction <https://www.youtube.com/watch?v=9iol3Lk6kyU&t=1157s>

Leland McInnes talk at PyData meeting, Los Angeles 2019, Topological Techniques for Unsupervised Learning, <https://www.youtube.com/watch?v=7pAVPjwBppo&t=1514s>

How to use tSNE effectively <https://distill.pub/2016/misread-tsne/>

Tutorials with Smallvis from James Melville <https://jlmelville.github.io/smallvis/>

Understanding UMAP: <https://pair-code.github.io/understanding-umap/>

How exactly UMAP works: <https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668>

tSNE vs. UMAP, Global Structure: <https://towardsdatascience.com/tsne-vs-umap-global-structure-4d8045acba17>

How to Cluster in High Dimensions: <https://towardsdatascience.com/how-to-cluster-in-high-dimensions-4ef693bacc6>

How to tune hyperparameters of tSNE: <https://towardsdatascience.com/how-to-tune-hyperparameters-of-tsne-7c0596a18868>

## References

- Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche & V. Vianu (Eds.), *Database theory—ICDT 2001. ICDT 2001* (Lecture Notes in Computer Science) (Vol. 1973). Springer. [https://doi.org/10.1007/3-540-44503-X\\_27](https://doi.org/10.1007/3-540-44503-X_27)
- Altman, N., & Krzywinski, M. (2018). The curse(s) of dimensionality. *Nature Methods*, 15, 399–400. <https://doi.org/10.1038/s41592-018-0019-x>
- Arefieva, V., Yu, J., & Egger, R. (2021). A machine learning approach to cluster destination image on instagram. *JTMA*. <https://doi.org/10.1016/j.tourman.2021.104318>
- Becht, E., McInnes, L., Healy, J., Dutertre, C. A., Kwok, I. W. H., Ng, L. G., Ginhoux, F., & Newell, E. W. (2018). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37, 38–44. <https://doi.org/10.1038/nbt.4314>



- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. ArXiv:0803.0476.
- Clarke, R., Ressom, H., Wang, A., et al. (2008). The properties of high-dimensional data spaces: Implications for exploring gene and protein expression data. *Nature Reviews Cancer*, 8, 37–49. <https://doi.org/10.1038/nrc2294>
- Tensorboard. <https://projector.tensorflow.org/>
- Kobak, D., & Linderman, G. C. (2019). UMAP does not preserve global structure any better than t-SNE when using the same initialization. *bioRxiv*. <https://doi.org/10.1101/2019.12.19.877522>
- Li, L. (2010). Dimension reduction for high-dimensional data. In H. Bang, X. Zhou, H. van Epps, & M. Mazumdar (Eds.), *Statistical methods in molecular biology. Methods in molecular biology (methods and protocols)* (Vol. 620). Humana Press. [https://doi.org/10.1007/978-1-60761-580-4\\_14](https://doi.org/10.1007/978-1-60761-580-4_14)
- Li, Q., Li, S., Hu, J., Zhang, S., & Hu, J. (2018). Tourism review sentiment classification using a bidirectional recurrent neural network with an attention mechanism and topic-enriched word vectors. *Sustainability*, 10(9), 3313.
- Li, X., Kang, Y., & Li, F. (2020). Forecasting with time series imaging. *Expert Systems with Applications*, 160, 113680.
- Liu, K., Yin, L., Lu, F., & Mou, N. (2020). Visualizing and exploring POI configurations of urban regions on POI-type semantic space. *Cities*, 99, 102610.
- McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *ArXiv e-prints 1802.03426*.
- Mead, A. (1992). Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society Series D (The Statistician)*, 41(1), 27–39. <https://doi.org/10.2307/2348634>
- Melville, J., Lun, A., Djekidel, M. N., & Hao, Y. (2020). *uwot: The uniform manifold approximation and projection (UMAP) method for dimensionality reduction*. <https://cran.r-project.org/web/packages/uwot/index.html>
- Payntar, N. D., Hsiao, W. L., Covey, R. A., & Grauman, K. (2021). Learning patterns of tourist movement and photography from geotagged photos at archaeological heritage sites in Cuzco, Peru. *Tourism Management*, 82, 104165.
- PCAtools R library. <https://bioconductor.org/packages/devel/bioc/vignettes/PCAtools/inst/doc/PCAtools.html>
- R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Ringnér, M. (2008). What is principal component analysis? *Nature Biotechnology*, 26(3), 303–304. <https://doi.org/10.1038/nbt0308-303>
- Shirkhorshidi, A. S., Aghabozorgi, S., & Wah, T. Y. (2015). A comparison study on similarity and dissimilarity measures in clustering continuous data. *PLoS One*, 10(12), e0144059. <https://doi.org/10.1371/journal.pone.0144059>
- Stein-O'Brien, G. L., Arora, R., Culhane, A. C., Favorov, A. V., Garmire, L. X., Greene, C. S., Goff, L. A., Li, Y., Ngom, A., Ochs, M. F., Xu, Y., & Fertig, E. J. (2018). Enter the matrix: Factorization uncovers knowledge from omics. *Trends in Genetics*, 34(10), 790–805. <https://doi.org/10.1016/j.tig.2018.07.003>
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1), 267–288.
- Udell, M., Horn, C., Zadeh, R., & Boyd, S. (2016). Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1), 1–118. <https://doi.org/10.1561/22000000055>
- van der Maaten, L. (2013). *Barnes-Hut tSNE*. <https://arxiv.org/abs/1301.3342>
- van der Maaten, L. J. P., & Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579–2605.
- Yin, H. (2007). Nonlinear dimensionality reduction and data visualization: A review. *The International Journal of Automation and Computing*, 4, 294–303. <https://doi.org/10.1007/s11633-007-0294-y>

# Classification



## Data-Driven Categorization of Objects in Tourism

Ulrich Bodenhofer and Andreas Stöckl

### Learning Objectives

- Illustrate the use of classification methods in tourism
- Explain the most basic methods of classification
- Explain the methods for evaluating classification results
- Demonstrate how various methods can be applied to a classification task in Python using scikit-learn and Jupyter

## 1 Introduction and Theoretical Foundations

### 1.1 Motivation and Basic Concepts

Classification, that is, the task of assigning objects to categories, is an omnipresent subject in many disciplines. Notorious examples include, for instance, (1) medical diagnosis, where the “objects” take the place of patients along with their symptoms and the categories are usually “positive” or “negative” (i.e., patient does/does not have the disease), (2) quality control, where the objects are the products being checked and the categories are “ok” or “non-good,” or (3) character recognition, where the objects are images showing characters and the categories are the actual letters/digits that are supposedly shown on the images.

The above three examples are concerned with inferring the correct categorization of an object without any reference to time. However, classification can also concern

---

U. Bodenhofer · A. Stöckl (✉)

School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Hagenberg, Austria

e-mail: [ulrich.bodenhofer@fh-hagenberg.at](mailto:ulrich.bodenhofer@fh-hagenberg.at); [andreas.stoeckl@fh-hagenberg.at](mailto:andreas.stoeckl@fh-hagenberg.at)

itself with the categorization of an object as it will occur in the future. In such a case, one does not simply categorize that object but, rather, *predicts* its category. Such tasks, in particular, tend to appear frequently in tourism, as the following four examples demonstrate:

- **Lead scoring**—the objects are potential customers, and the task is to predict whether or not the potential customer will eventually become a paying customer.
- **Opportunity scoring**—the objects are requests by customers or offers made to them, and the task is to predict whether or not the customer will eventually buy/book.
- **Cross-/upselling**—the objects are customers along with their purchase/booking history, and the task is to predict whether or not the customer will eventually buy/book another product (cross-selling), an additional package, or an upgrade for a better offer (upselling).
- **Churn prediction**—the objects are customers along with their purchase/booking history, and the task is to predict whether or not the customer will eventually stop buying/booking or leave an ongoing contract or subscription.

Other advanced applications of classification within the tourism sector range from predicting hotel ratings (Veloso et al., 2019) and classifying hosts (Ramos-Henríquez et al., 2021) to distinguishing tourists from non-tourists in passive mobile data (Reif & Schmücker, 2020).

Formally speaking, a *classifier* (or *classification function*) is a mapping  $g : X \rightarrow Y$  from an object set  $X$  to a (usually finite) set of categories  $Y$ . If only two categories are present, one speaks of *binary classification*; otherwise, one speaks of *multi-class classification*. Binary classification tasks occur whenever one wishes to answer a yes/no question about an object. In such a case, it is very common to use the two labels +1 (positive) and  $-1$  (negative) or, equally common, 1 and 0. Such tasks appear, for instance, in medical diagnosis and quality control but also in opportunity scoring (see above).

In the following, we will not identify whether objects should be categorized as they are in the present (first set of examples above) or whether the categorization relates to the objects' future (tourism examples above). For the sake of simplicity, we will merely speak of *prediction* when applying a classifier  $g : X \rightarrow Y$  to a new object.

Classification functions can be defined explicitly, for example, by designing formulas or a set of decision rules. However, in many practical situations, it might be challenging to do so; thus, it has become standard to design classification functions in a data-driven fashion via *machine learning*. In other words, the classification function is identified/trained by considering sample objects for which the correct category is known. This *data-driven design of classification functions* is the subject of the present chapter.

In the following, we assume that a well-defined set of features represents the objects we want to classify. If so, we can write up a data set as a table in the following way:

$$\begin{array}{cccccc}
 x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} & \\
 x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} & \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \\
 x_{l-1,1} & x_{l-1,2} & \cdots & x_{l-1,d-1} & x_{l-1,d} & \\
 x_{l,1} & x_{l,2} & \cdots & x_{l,d-1} & x_{l,d} & 
 \end{array}$$

Every row corresponds to one sample/object, meaning we have  $l$  samples in total. Furthermore, every sample is represented by  $d$  features, each of which corresponds to one column in the table. Thus,  $x_{i,j}$  is the entry that is in the  $i$ -th row and  $j$ -th column, and it contains the value of the  $j$ -th feature for the  $i$ -th sample. As an example, if every row corresponds to one customer and the fourth column corresponds to the age of that customer, then  $x_{34,4}$  is the age of the 34th customer. Features/columns can be characterized by different types as well; the most common case is that features/columns are *numerical*, i.e., numbers from either a continuous or discrete scale, or features/columns can also be *categorical*, i.e., the column values belong to a finite set of categories.

As previously mentioned, this chapter is concerned with the data-driven design of classifiers. To learn a classification function from the data, the data objects need to be labeled, i.e., we must know which category each sample object belongs to. Hence, our representation of the data objects is complemented by a label vector  $\mathbf{y}$  that we can conveniently append to our data matrix as  $d + 1$ -st column:

$$\begin{array}{cccccc}
 x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} & y_1 \\
 x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} & y_2 \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 x_{l-1,1} & x_{l-1,2} & \cdots & x_{l-1,d-1} & x_{l-1,d} & y_{l-1} \\
 x_{l,1} & x_{l,2} & \cdots & x_{l,d-1} & x_{l,d} & y_l
 \end{array}$$

Given such a matrix, we can use a method that builds a classification function so that, for each row in this table, the category  $y_i$  is predicted as accurately as possible based on the  $d$  input features  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$ . Therefore, we try to find a classification function  $g : X \rightarrow Y$  such that  $g(\mathbf{x}_i)$  matches  $y_i$  as closely as possible for every sample. The standard way is to search for a classification function that minimizes the *misclassification rate*, that is, the percentage of samples for which  $g(\mathbf{x}_i) \neq y_i$ . Alternative objectives, other than the misclassification rate, might also appear in practice (see next section below), and the objective may be augmented with regularization terms (Hastie et al., 2009; Vapnik, 1998).

## 1.2 Evaluation

### 1.2.1 Generalization Error

Given a classification function  $g : X \rightarrow Y$ , regardless of whether or not it had been trained from data or designed explicitly, we are typically interested in how it will perform when applied to real data. For instance, a system should be accurate in predicting the bookings of future customers rather than in reproducing bookings from the past. The same holds true in medical diagnosis, quality control, handwritten character recognition, and so on.

When it comes to a particular measure for how well a classification performs (e.g., the misclassification rate defined above), the so-called *generalization error* is the expected value of this quality measure on future data. In simple terms, the generalization error is the average performance measure on a large number of future samples. Since the generalization error is defined as an expected value relating to future data, it relies on the inherent probability distribution of future data. In realistic, practical cases, this probability distribution is unknown and cannot be estimated in a feasible way.

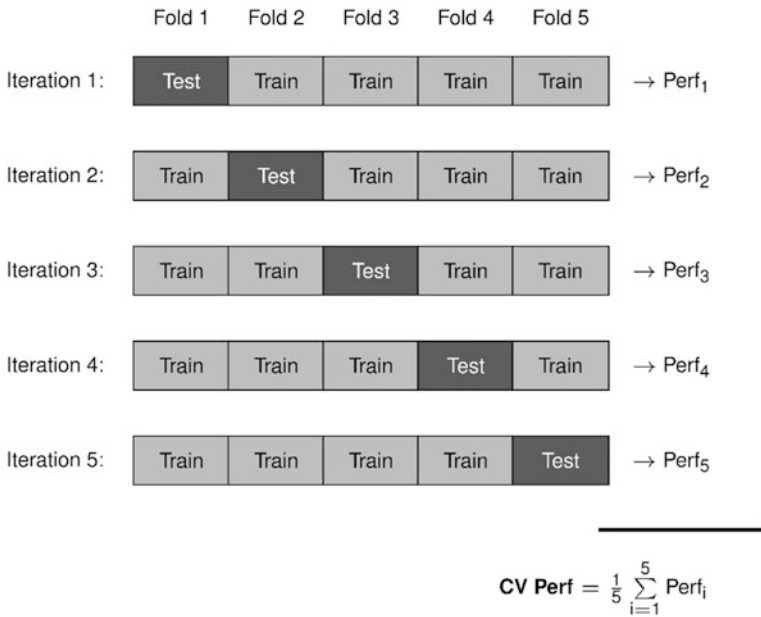
### 1.2.2 Hold-Out Method and Cross Validation

Since we cannot determine the generalization error precisely, we need to estimate it. The *hold-out method* (also called *test set method*) is a very simple, yet effective, way of doing so. First, assume we are given a data set with labeled samples, as shown above. We can then select a certain portion of the data as *training set*, i.e., the part of the data that we use to train our classification function on, and the remaining data as a *test set*, i.e., the part of the data that we use to evaluate the classification performance of our classification function. A 70%/30% split into training and test sets is very common; however, any other split ratios can also be useful. Under the assumption that the data have been independently sampled from the same source and that the split between the training and test samples has been randomized,<sup>1</sup> the quality measure on the test set leads to an *unbiased estimate of the true generalization error* (Hastie et al., 2009).

While the hold-out method is quite straightforward, it does come with a major disadvantage: splitting data into two sets poses a severe limitation for applications with a small number of samples. If we allocate more samples to the test set, the estimation of the generalization error will be good, but, due to the small training set, the model quality will suffer. If, on the contrary, we allocate more samples to the training set, the model will be more accurate, but the quality of the generalization error estimate will suffer.

---

<sup>1</sup>In more technical terms, samples need to be independently sampled from the same distribution. This property is usually abbreviated as i.i.d. (= independent identically distributed).



**Fig. 1** Five-fold cross validation

This is where *cross validation* comes in, offering an alternative solution without this disadvantage. We first randomly partition the data into a certain number  $k$  of non-overlapping subsets, a.k.a., *folds*. Thus, we speak of  $k$ -fold cross validation (e.g., ten-fold cross validation is very common). Next, each of the  $k$  folds serves as the test set once, while the remaining  $k - 1$  are used to train a classification function that is then evaluated on the withheld test set. In this way, we train  $k$  models on  $k - 1$  training folds, each of which is evaluated on the withheld test fold. This procedure does not evaluate a single model but, rather, evaluates the general model training procedure. A theoretical result ensures that cross validation provides an almost unbiased estimate of generalization performance (Hastie et al., 2009; Luntz & Brailovsky, 1969). Figure 1 illustrates cross validation for the example  $k = 5$ .

### 1.2.3 Hyperparameter Selection

Almost all machine learning methods have parameters that are trained to minimize a certain performance measure on a labeled training set. Most machine learning methods additionally have *hyperparameters*, that is, parameters that are not trained but need to be specified prior to being trained by the user. These hyperparameters usually control the complexity of the models to address the overfitting-underfitting trade-off (Hastie et al., 2009; Vapnik, 1998) or the training process (e.g., learning

rates). For more details, see Chapter “Hyperparameter Tuning” and further sources from the literature (Bergstra et al., 2011; Tran et al., 2020).

### 1.2.4 Evaluation Measures

Thus far, we have only vaguely touched upon the topic of classification performance in which merely one specific measure has been mentioned: the classification error, i.e., the relative frequency of incorrectly classified samples. Based on this measure, we can define generalization error as the expected/average classification error on future samples. Needless to say, classification error is not the only measure of classification performance that exists. Depending on the actual task and its requirements, other measures might indeed be helpful as well. Therefore, this section will provide a brief overview of such measures. For more details, also refer to Chap. 13.

#### Confusion Matrix and Evaluation Measures Computed Therefrom

Let us first consider a binary classification task where the two classes are +1 (positive class) and −1 (negative class), and suppose that we have a labeled set of samples, i.e., one for which the correct classification is known and a classification function  $g : X \rightarrow Y$ . Given a sample  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$  with actual class  $y_i$ , we end up with four possible scenarios:

- $y_i = +1$  and  $g(\mathbf{x}_i) = +1$ :  $\mathbf{x}_i$  is a positive sample that is correctly classified as positive—this is called a **true positive (TP)**.
- $y_i = +1$  and  $g(\mathbf{x}_i) = -1$ :  $\mathbf{x}_i$  is a positive sample that is incorrectly classified as negative—this is called a **false negative (FN)**.
- $y_i = -1$  and  $g(\mathbf{x}_i) = +1$ :  $\mathbf{x}_i$  is a negative sample that is incorrectly classified as positive—this is called a **false positive (FP)**.
- $y_i = -1$  and  $g(\mathbf{x}_i) = -1$ :  $\mathbf{x}_i$  is a negative sample that is correctly classified as negative—this is called a **true negative (TN)**.

A confusion table is a matrix where each row corresponds to a true class and each column corresponds to a putative class yielded by a classification function. The entry in the  $i$ -th row and  $j$ -th column corresponds to the number of samples that belong to the  $i$ -th class and are classified/predicted as the  $j$ -th class by the classification function. Naturally, in the case of binary classification, we obtain a  $2 \times 2$  matrix containing the numbers of true positives, false negatives, false positives, and true positives, as shown in Table 1.

**Table 1** Confusion table for binary classification

		predicted class $g(\mathbf{x})$	
		+1	−1
True class $y$	+1	<b>#TP</b>	<b>#FN</b>
	−1	<b>#FP</b>	<b>#TN</b>

Clearly, the higher the numbers in the diagonal (which correspond to correct classifications), the better the result. However, in general, it is difficult to evaluate four numbers simultaneously, particularly because absolute numbers’ actual meaning also strongly rely on the number of samples and the ratio of positive and negative samples. It is common, therefore, to compute certain classification performance measures from the confusion table that allow for a unified evaluation of classification results. These include the following:

Accuracy:	$ACC = \frac{\#TP + \#TN}{\#TP + \#FN + \#FP + \#TN}$
Classification error:	$ERR = 1 - ACC = \frac{\#FP + \#FN}{\#TP + \#FN + \#FP + \#TN}$
True positive rate (aka recall/sensitivity):	$TPR = \frac{\#TP}{\#TP + \#FN}$
False negative rate:	$FNR = \frac{\#FN}{\#TP + \#FN}$
False positive rate:	$FPR = \frac{\#FP}{\#FP + \#TN}$
True negative rate (aka specificity):	$TNR = \frac{\#TN}{\#FP + \#TN}$
Positive predictive value (aka precision):	$PPV = \frac{\#TP}{\#TP + \#FP}$
Negative predictive value:	$NPV = \frac{\#FN}{\#FN + \#TN}$
False discovery rate:	$FDR = 1 - PPV = \frac{\#FP}{\#TP + \#FP}$

All of these measures can be applied to training sets (though this is of limited value), validation sets, and test sets. If the test set consists of samples drawn independently from the same distribution as the training set, then the value obtained for the test set is an unbiased estimate of the measure’s expected value on future data (see section “Hold-out Method and Cross Validation”).

Some of the measures depicted above are of limited value in cases where an unbalanced classification task is present, that is, when the numbers of samples differ significantly for the two classes. Assume that the data are distributed in such a way that 1% of the samples belong to the positive class, and 99% of the samples belong to the negative class. In such a situation, a trivial classifier, assigning every sample to the negative class, would have an accuracy of 99%—which would otherwise likely be an outstanding value. For such unbalanced situations, the following specific performance measures have been developed:

Balanced accuracy:	$BACC = \frac{TPR + TNR}{2}$
$F_1$ score:	$F_1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}$
Matthews correlation coefficient:	$MCC = \frac{\#TP \cdot \#TN - \#FP \cdot \#FN}{\sqrt{(\#TP + \#FN) \cdot (\#FP + \#TN) \cdot (\#TP + \#FP) \cdot (\#FN + \#TN)}}$

Balanced accuracy can be considered the simplest measure as it weighs both classes equally and, therefore, prevents the larger class from overruling the smaller one. The  $F_1$  score, on the other hand, is a measure geared particularly towards situations in which the positive class is the smaller one. It neglects true negatives and concentrates on whether positive classifications are correct. Lastly, the Matthews correlation coefficient (MCC) computes the determinant of the confusion table and normalizes it by taking the square root of the product of all the row and column



**Table 2** Confusion table for a classification task with  $M$  classes

		predicted class $g(\mathbf{x})$				
		1	...	$j$	...	$M$
True class $y$	1	$C_{1,1}$	...	$C_{1,j}$	...	$C_{1,M}$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
	$i$	$C_{i,1}$	...	$C_{i,j}$	...	$C_{i,M}$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
	$M$	$C_{M,1}$	...	$C_{M,j}$	...	$C_{M,M}$

**Table 3** Binary confusion table obtained by singling out one class

		predicted class $g(\mathbf{x})$	
		$=i$	$\neq i$
True class $y$	$=i$	$\#TP_i$	$\#FN_i$
	$\neq i$	$\#FP_i$	$\#TN_i$

sums. This ensures that the MCC always provides values between  $+1$  (perfect classification) and  $-1$  (also perfect classification yet with swapped classes). An MCC with a value around  $0$  means that the classification is more or less random. Finally, it is worth noting that the MCC is not a concept of its own, but merely a handy formula for the common Pearson correlation coefficient between true and predicted classes.

Now assume that we have a multi-class task, a classification task with more than two categories. Once again, we have defined confusion tables, like the one above, as follows: the entry in the  $i$ -th row and  $j$ -th column corresponds to the number of samples that belong to the  $i$ -th class and are classified as the  $j$ -th class by the classification function. This definition carries over to multi-class tasks as well; in the example below, classes are denoted with  $1 \dots M$ , where  $M$  is the total number of classes (see Table 2).

The interpretation of this table is the same as earlier in that correct classifications are found diagonally in the table, while off-diagonal entries correspond to misclassifications. The two most apparent classification performance measures for multi-class tasks can be defined as follows:

- **Accuracy:**  $ACC = \frac{\sum_i C_{i,i}}{\sum_{i,j} C_{i,j}}$
- **Classification error:**  $ERR = 1 - ACC = \frac{\sum_{i \neq j} C_{i,j}}{\sum_{i,j} C_{i,j}}$

The other measures cannot be generalized to multi-class tasks in a direct manner; however, it is possible to consider a binary classification task separately for each class. Doing so for the  $i$ -th class results in a binary confusion table, as shown in Table 3.

For example, a sample is a false positive with regard to the  $i$ -th class if the sample is classified as belonging to this class, whereas the sample actually belongs to a different class. Correspondingly, we can define all of the above measures analogously—but per class. Let us consider the following example:

- **True positive rate (aka recall/sensitivity):**  $TPR_i = \frac{TP_i}{TP_i + FN_i}$

Then we can also transfer the following concept to multi-class tasks:

- **Balanced accuracy:**  $BACC = \frac{1}{M} \sum_{i=1}^M TPR_i$

## ROC Analysis

The performance measures mentioned above are defined for final classification results, i.e., when each sample is directly assigned to a class/category. Most classification methods, however, do yield continuously valued scores that are often, but not always, probabilities. The scores then have to be turned into final classifications by thresholding or by assigning a sample to the class that has been given the highest score.

In binary classification, it is common to have a one-dimensional classification score that is interpreted as the qualitative likelihood of the sample belonging to the positive class. In other words, the higher the score, the more positive the sample appears to be, and the lower the score, the more negative the sample appears to be. Such a classification score can be turned into a final binary classification by applying a simple threshold in which all samples with scores above the threshold are assigned to the positive class and all samples with scores below the threshold are assigned to the negative class. The choice of this threshold is not always straightforward, as a higher threshold avoids false positives but favors false negatives, thus leading to better TNR/specificity yet to worse TPR/sensitivity. Contrarily, lowering the threshold reduces false negatives but may introduce additional false positives, resulting in worse TNR/specificity yet better TPR/sensitivity. This is commonly known as the sensitivity-specificity trade-off in medical diagnostics (Florkowski, 2008) but appears analogously in any other binary classification task.

By considering all thresholds, receiver operator characteristic (ROC) curves allow for the evaluation of a continuous classification score on a labeled test set (Fawcett, 2006; Florkowski, 2008). As such, a ROC curve plots the false positive rate on the horizontal axis against the true positive rate on the vertical axis for all possible thresholds. The following figure shows an example of a ROC curve (Fig. 2):

It is clear that a threshold higher than the highest score would classify all samples as negative, resulting in  $TPR = 0$  and  $FPR = 0$ . This marks one end of the curve on the bottom left-hand corner. On the other hand, a threshold as high as the lowest score would classify all samples as positive, resulting in  $TPR = 1$  and  $FPR = 1$ , as can be seen on the top right-hand corner. In between these two points, the curve is monotonic, meaning that it either goes up or right but never down or left.

The better a classification score, the more it assigns higher scores to positive samples and lower scores to negative samples. In an ideal case, all positive samples would be ranked above the negative ones, and a perfect threshold that precisely separates the positive from the negative samples would exist. Such a situation would

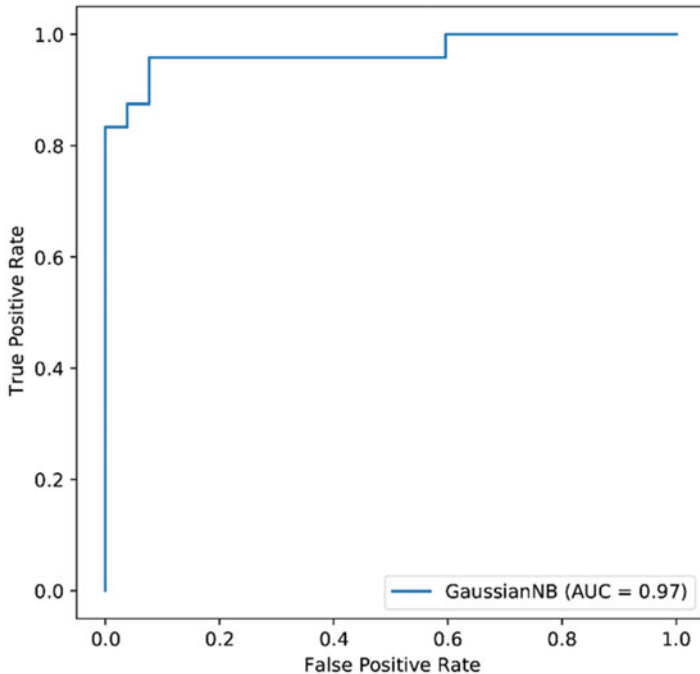


Fig. 2 Example of a ROC curve

result in a ROC curve consisting of a vertical line up to  $TPR = 1$  and  $FPR = 0$  as well as a horizontal line connecting this point with the upper right-hand corner. If, on the other hand, the classification score is independent of the true class, then the classification score ranks positive and negative samples randomly. This results in a ROC curve close to the diagonal.

ROC curves provide a good visual assessment of the *ranking performance of a classification function*, that is, the ability to rank up (give higher scores to) positive samples and/or rank down (give lower scores to) negative samples. However, ROC curves in their entirety do not allow for a quantitative evaluation of ranking performance. As the above example suggests, the closer the curve comes to the top left-hand corner, the better the ranking performance is. Noticeably, the *area under the ROC curve* (ROC-AUC or short AUC) is a simple, expressive quantitative measure of the general ranking performance of a classification score.

## Categorical Cross-Entropy

Several classification methods yield specific classification scores—class probabilities. In a binary case (for mathematical convenience, we assume  $y \in \{0, 1\}$ ), this is typically a single classification score that corresponds to the posterior class probability  $g(\mathbf{x}) = p(y = 1 | \mathbf{x})$ . Needless to say, the converse probability follows immediately:  $p(y = 0 | \mathbf{x}) = 1 - p(y = 1 | \mathbf{x})$ . Such a classification function should, of course, assign high probabilities  $p(y = 1 | \mathbf{x})$  to samples belonging to class 1 and high probabilities  $p(y = 0 | \mathbf{x})$  to samples belonging to class 0, where the latter is equivalent to assigning low probabilities  $p(y = 1 | \mathbf{x})$  to samples belonging to class 0. In this sense, given a sample  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$  with actual class  $y_i$ , we want to maximize  $g(\mathbf{x}_i) = p(y_i = 1 | \mathbf{x}_i)$  if  $y_i = 1$  and minimize it if  $y_i = 0$ . This notion can be unified into a single measure as follows:

$$g(\mathbf{x}_i)^{y_i} \cdot (1 - g(\mathbf{x}_i))^{(1-y_i)}$$

At first glance, this formula appears more complicated than it truly is; looking at it from a different perspective, since  $y_i$  is either 0 or 1, it simplifies to  $g(\mathbf{x}_i)$  if  $y_i = 1$  and to  $1 - g(\mathbf{x}_i)$  if  $y_i = 0$ . Maximizing this measure is the equivalent to minimizing its negative logarithm. Ultimately, this leads to the so-called binary *cross-entropy*:

$$H(g(\mathbf{x}_i), y_i) = -\log \left[ g(\mathbf{x}_i)^{y_i} \cdot (1 - g(\mathbf{x}_i))^{(1-y_i)} \right] = -y_i \cdot \log(g(\mathbf{x}_i)) - (1 - y_i) \cdot \log(1 - g(\mathbf{x}_i))$$

We can also define the total cross-entropy of the entire data set by simply averaging the above measure for all the samples. Moreover, for multi-class tasks, a straightforward extension of the binary cross-entropy also exists (for details, see, e.g., Murphy, 2012). What all of these variants have in common is that these performance measures depend on their first arguments (the classification scores) in a differentiable way. This is particularly advantageous in cases where one needs to optimize the classification function (resp. its parameters) via gradient-based optimization methods.

## 1.3 Data Preprocessing

### 1.3.1 One-Hot Encoding

Most classification methods that we will deal with subsequently require data to be exclusively numerical. In case the original data table contains categorical features, one can always map them to numerical columns by applying the so-called *one-hot encoding*, which replaces a categorical feature with one binary column per category.

**Table 4** Example of how to one-hot encode a categorical feature with three categories “A,” “B,” and “C.” The left column shows the original feature, while the right-hand side shows the three resulting features—one per category

		One binary feature per category		
Original feature		A	B	C
↓		↓	↓	↓
A		1	0	0
A		1	0	0
C	→	0	0	1
B		0	1	0
A		1	0	0
C		0	0	1

This is best explained by means of an example; as such, Table 4 shows how to encode a categorical feature with three categories “A,” “B,” and “C”

### 1.3.2 Feature Scaling/Normalization

Moreover, some of the classification methods are sensitive to the scaling of input features. To tackle this issue, it is common to apply a standardized *scaling* procedure to all input features; this process is also known as *normalization*. The most common variants thereof are as follows:

- Scale and shift each feature/column so that the values in this column have a mean of 0 and a standard deviation of 1. This can be achieved by subtracting the column mean and dividing by the standard deviation of the values in this column.
- Scale and shift each feature/column so that each column has a minimum value of  $-1$  (or 0) and a maximum value of  $+1$ . To have a minimum value of 0 and a maximum value of 1, this can be achieved by subtracting the column-wise minimum from each value and dividing by the range (difference between maximum and minimum) of the column. To have a minimum value of  $-1$  and a maximum value of  $+1$ , this can be achieved by first applying the aforementioned transformation to  $[0, 1]$  and then subtracting  $-1/2$  and multiplying the values by 2.

### 1.3.3 Projection Methods

If the input space is high-dimensional, it might help to exploit possible correlations between input features and reduce the inputs to the most essential underlying components. For this purpose, projection methods are most suitable (Oskolkov, 2021). In particular, principal component analysis (PCA) is very useful for reducing input data to a compact set of expressive yet abstract features (Jolliffe, 2014). For further information, see Chap. 9.

### 1.3.4 Missing Values Imputation

The classification methods that will be introduced in the next section require all feature values of all samples to be known. In other words, any missing values in the data cannot be handled properly. If some data items are missing, which can indeed frequently appear in practice, one either needs to remove the corresponding items (rows or columns) or apply an imputation method. A *missing values imputation method* refers to an algorithm that guesses or predicts missing values in the data by exploiting information about the distribution of input data or their possible correlations.

The simplest method for numerical data is to impute missing values via the median or mean of the respective column. This is often called *strawman imputation*. For categorical columns, it is also common to impute with the most frequent category. When more complex schemes and exploiting dependencies across features is involved, usually regression or even advanced machine learning methods are used to impute missing values. For more detailed insights, refer to the literature (e.g., Allison, 2002; Horton & Kleinman, 2007; Stekhoven & Bühlmann, 2012).

### 1.3.5 General Caveat

The abovementioned pre-processing methods are often used in conjunction with the hold-out method or cross validation. Regardless of whether normalization, projection methods, or missing values imputation is used, all of these methods require the process of fitting the parameters to the data set that is currently under investigation. It is additionally important to note that one should apply these transformations in such a way that the parameters of the transformations are only determined from the training set. If not, an undesirable hidden bias towards the test set may occur.

## 1.4 Classification Methods

In the following, we will highlight the most important machine learning methods for data-driven classification. This list is far from exhaustive, but it includes the most well-known options. Note that the subsequent chapter on regression (Stöckl & Bodenhofer, 2021) relies on many of the concepts introduced here.

### 1.4.1 K-Nearest Neighbor

The *nearest neighbor classifier* is, more or less, the simplest type of classifier. It is non-parametric in the sense that it does not require fitting certain parameters to a training set. Instead, the labeled training set itself is the model. It works as follows—

given a new input sample, the training sample is determined whose input vector has the smallest distance to the new sample. The new sample is then classified as the same category/class as the closest training sample (Fix & Hodges, 1951).

It is well known that the nearest neighbor classifier is very prone to noisy data and outliers. Therefore, to counteract this disadvantage, it is quite common to consider the  $k$  samples closest to the sample that should be classified, instead of considering the closest neighbor only. Given these  $k$  nearest neighbors, the new sample is classified as the category/class that appears most often among the  $k$  nearest neighbors. This simple modification reduces the risk of overfitting to outliers or noisy training samples. However, the larger  $k$  is chosen, the less it can model more subtle patterns in the data, thus leading to underfitting. As such,  $k$  is a hyperparameter that needs to be chosen carefully in order to adequately address the underfitting-overfitting trade-off. Since  $k$  is the only hyperparameter, hyperparameter selection tends to be relatively easy, and grid search is feasible.

#### Advantages

- No training required.
- Simple and easy.
- Simple hyperparameter selection.

#### Disadvantages

- Sensitive to input scaling; thus, normalization is advisable.
- Sensitive to irrelevant/noisy features; thus, feature selection is advisable.
- Choice of distance measure depends on the application; may not be straightforward.
- The computational complexity of predictions depends on the number of training sets; costly for large training sets.

### 1.4.2 Logistic Regression

*Logistic regression* is a linear model that separates the input space into two classes using a linear hyperplane. In the case of two-dimensional inputs, this corresponds to a straight line separating the input plane into two half-planes, one for each class. The classification function, however, is not just a linear function of the inputs but, rather, a linear function that is transformed to the unit interval  $[0,1]$  using a sigmoid (= inverse logit) function:

$$g(\mathbf{x}) = g(x_1, \dots, x_d) = \text{logit}^{-1}\left(\beta_0 + \sum_{i=1}^d \beta_i x_i\right), \text{ where } \text{logit}^{-1}(y) = \frac{1}{1 + e^{-y}}.$$

This model is therefore a sum of inputs weighted by coefficients  $\beta_i$ , while  $\beta_0$  is a constant (the so-called intercept) that shifts the separating hyperplane away from the origin. The sigmoid function transforms the linear regressor to the unit interval  $[0, 1]$ . If the parameters are optimized in such a way that the cross-entropy is minimized, the output of the classification function can be interpreted as an estimate of the posterior class probability  $p(y = 1 | \mathbf{x})$  (Cox, 1966; Cramer, 2002).

Logistic regression can be further augmented by adding a regularization term, with its goal being to aid feature selection and avoid overfitting. The most important representatives are logistic ridge regression (also known as L2 regularization; Hoerl, 1962; Le Cessie & van Houwelingen, 1992), LASSO (least absolute shrinkage and selection operator, also known as L1 regularization; Tibshirani, 1996), and elastic net (Zou & Hastie, 2005).

### Advantages

- Computationally efficient.
- Simple and interpretable models.
- Built-in feature selection (in particular, the regularized variants); thus, insensitive to irrelevant/noisy features.

### Disadvantages

- The regularized variants are sensitive to input scaling; thus, normalization is advisable.
- Only reasonable in cases where a linear model is appropriate/called for.

## 1.4.3 Naïve Bayes

We now come to a family of classifiers that also yield posterior probabilities  $p(y = j | \mathbf{x})$ , i.e., the probability that an input sample  $\mathbf{x}$  belongs to the  $j$ -th class. *Bayes classifiers* are based on the well-known Bayes theorem in which the posterior probability can be broken down as follows:

$$p(y = j | \mathbf{x}) = \frac{p(y = j) \cdot p(\mathbf{x} | y = j)}{p(\mathbf{x})}$$

The denominator in this formula,  $p(\mathbf{x})$ , the so-called *evidence*, can be omitted since it does not depend on the class  $j$ ; therefore, it is sufficient to consider the following formula:



$$p(y = j | \mathbf{x}) \propto p(y = j) \cdot p(\mathbf{x} | y = j)$$

The distribution of classes  $p(y = j)$  can easily be identified from the data, while, in many practical cases, the input distribution of classes  $p(\mathbf{x} | y = j)$ , the so-called *likelihood*, cannot be identified from the data in a meaningful way. *Naïve Bayes classifiers* are based on the simple idea of approximating the likelihood of the (usually false) assumption that input features are statistically independent (Hand & Yu, 2001):

$$p(\mathbf{x} | y = j) = \prod_{i=1}^d p(x_i | y = j)$$

This approximation renders the likelihood tractable. The marginal class distributions  $p(x_i | y = j)$  can be estimated, for instance, by assuming multinomial distributions for categorical features or by using kernel density estimations or simply Gaussian distributions for numerical features (John & Langley, 1995).

#### Advantages

- Computationally efficient; scalable to large numbers of samples and features.
- Simple and easy.
- Can handle both categorical and numerical features.
- Insensitive to irrelevant/noisy features.

#### Disadvantages

- Independence assumption is hardly satisfied in practice, which can hamper classification performance.

### 1.4.4 Decision Trees

*Decision trees* are rule-based classifiers that classify samples by consecutively answering questions about input features. Consider the following example:

The first question displayed in the root node of the tree is whether the input feature “run” is smaller than or equal to 0.265. If a sample fulfills this condition, then we can continue on to the left branch of the tree with the question of whether the input feature “triathlon” is smaller than or equal to 0.092. This procedure continues until we end up at a leaf node. Every leaf node is associated with a final class (“not booked” or “booked” for the example above).

While it is easy to see how classification via applying a decision tree works, it is unclear as to how a tree is trained on the basis of a labeled data set. In the example shown above, the training set consists of 176 samples, where 120 belong to the “not

booked” class and 56 belong to the “booked” class. A common decision tree learning algorithm considers all possible questions/splits and evaluates them based on a so-called *splitting criterion*, that is, a numerical measure that evaluates a split and chooses the one that has the best value. For the example in Fig. 3, the split is chosen along the feature “run” with a threshold of 0.265. This split would classify 68 training samples into the left branch (of which 20 belong to the “not booked” class and 48 belong to the “booked” class) and 108 training samples into the right branch (of which 100 belong to the “not booked” class and only eight belong to the “booked” class). In this way, this split affects the class distributions in either branch to a very different extent than the original distribution, which is precisely what is desired. If the class distribution does not change, then the split has failed to have an effect.

Generally speaking, the splitting criterion is an objective measure for deciding which split appears to be most helpful for the task of separating the classes. Common variants include the entropy-based *information gain* (as used in the ID.3 and C4.5 algorithms; see Quinlan, 1986, 1993) or the *Gini impurity* (as used in the CART algorithm; see Breiman et al., 1984). Once one split has been determined, the same procedure can be applied recursively to the subsets of the training data in either branch. Hence, a tree is created. This recursive procedure stops whenever a stopping criterion is fulfilled, thus forming a leaf node. The following stopping criteria are commonly used: (1) a maximum tree depth has been reached; (2) the number of samples belonging to a branch has fallen below a certain threshold; (3) the samples belonging to a branch dominantly fall in one category. Some decision tree learning algorithm variants grow the trees to full size and then collapse unnecessary sub-trees during a post-processing step called *pruning*.

The final classification of a new sample is completed once all split questions, starting from the root node of the tree up to a leaf node, have been answered. Thereafter, the new sample is assigned to the class that appeared to be most frequent in the training samples that ended up in this leaf node. For instance, if we assume that, in the example above, we finish in the leaf node that appears fourth from the left side, the sample would be classified as the “booked” class since 37 of the 38 training samples assigned to this leaf node had been belonging to this class.

Decision trees are widely considered as *white-box models*, i.e., models that are easy to comprehend/interpret, even for non-experts. Nonetheless, in decision tree learning, it remains difficult to handle the underfitting-overfitting trade-off since the depth of the trees is the only means of addressing this trade-off. Moreover, decision trees only consider splits along single features, often rendering the resulting models inaccurate.

### Advantages

- Computationally efficient; scalable to large numbers of samples and features.
- Simple and easy.
- Models are easily interpretable by humans.

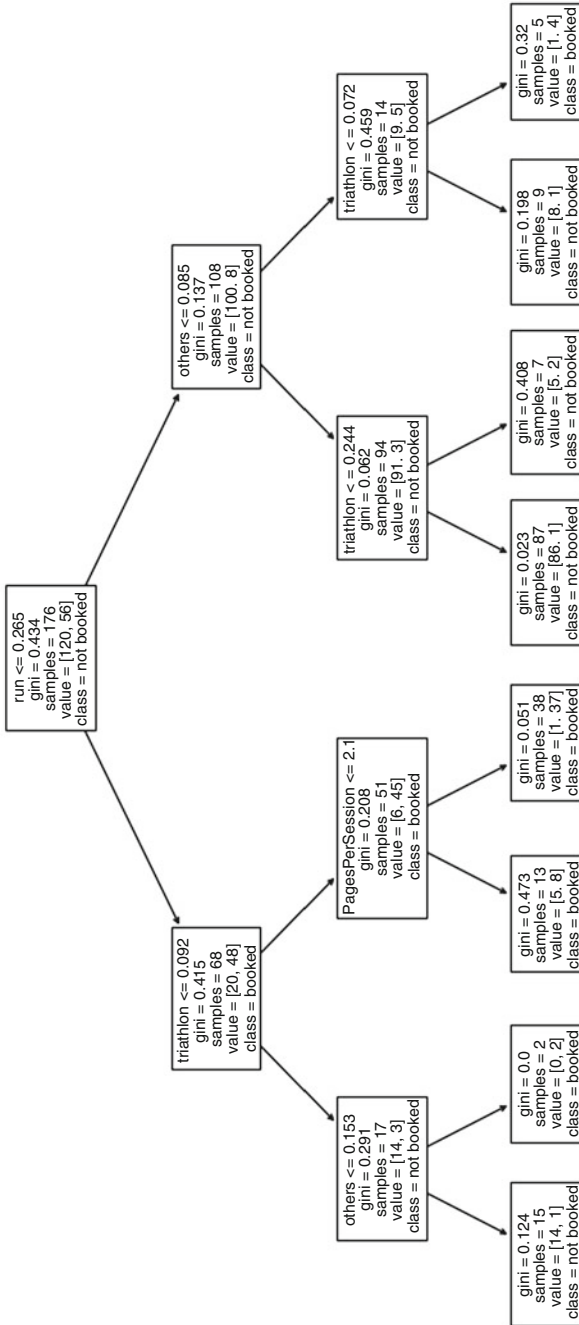


Fig. 3 Example of a decision tree of depth 3

- Can handle categorical and numerical features.
- Built-in feature selection; thus, insensitive to irrelevant/noisy features.
- For numerical features, only the order of the values is relevant; thus, insensitive to input scaling; no normalization required.

#### Disadvantages

- Limited accuracy.
- Difficult to address the underfitting-overfitting trade-off.

### 1.4.5 Random Forest

It has been established that aggregating an ensemble of multiple classifiers can lead to more robust and more accurate classification results (Opitz & Maclin, 1999). This is where *random forests*, which are considered ensembles of decision trees, play their part. The standard variant of random forests (Breiman et al., 1984; Breiman, 2001) uses CART decision trees along with two random components<sup>2</sup>:

1. Each tree is trained on a random sub-sample of the training set. While multiple sampling schemes are indeed available, it is most common to select as many samples as we have training samples, but these samples are drawn from the training set *with replacement*. Thus, each tree leaves out some of the training samples.
2. In each split, only a random sub-sample of features is considered, and the size of this sub-sample is a hyperparameter of the algorithm. The most common variants to consider for each split include  $d/2$  or  $\sqrt{d}$  features ( $d$  is the total number of input features, as defined above). This modification ensures that not always the same features are used. Instead, the trees are also forced to take less indicative, yet still useful, features into account.

Once a certain number of decision trees have been trained in this way, the final classification of a new sample can be performed by computing the classification results for every tree and assigning the sample to the class that the trees predicted most often. Additionally, one can view the relative frequencies of predicted classes, as obtained by the ensemble of trees, as an estimated posterior class probability  $p(y = j | \mathbf{x})$ .

Furthermore, random forests are characterized by the fact that increasing the number of trees (which is another hyperparameter) does not lead to increased overfitting (Breiman, 2001). Instead, to be more specific, increasing the number of

---

<sup>2</sup>Without such random components, decision tree learning algorithms would be deterministic, meaning that they would always lead to the exact same decision tree model if they were trained on the same data set again.

trees actually leads to more robust ensembles. Therefore, one should aim to choose a larger number of trees if the computational resources allow so. Since this hyperparameter is rather uncritical, random forests have the advantage of giving very accurate and valuable models even when default parameters are set.

Random forests are additionally advantageous in that the individual *importances of features* can be extracted relatively easily from the trained model. This offers insight into and facilitates feature selection. Moreover, another outstanding property of random forests is that estimates of the generalization error can even be computed from the training set. This can be accomplished by only considering the training samples for each tree that a tree has not previously used for training. The resulting so-called *out-of-bag (OOB) estimates* have been proven to provide unbiased estimates of the performance on future samples (Breiman, 2001).

#### Advantages

- Easy to apply; good results even without hyperparameter tuning.
- Built-in feature selection (in particular, the regularized variants); thus, insensitive to irrelevant/noisy features.
- For numerical features, only the order of the values is relevant; thus, insensitive to input scaling; no normalization required.
- Feature importances can be computed.
- Out-of-bag (OOB) estimates allow for estimating performance on future data.

#### Disadvantages

- Computationally expensive for large data sets.

### 1.4.6 Gradient Tree Boosting

Random forests aggregate a certain number of models, all of which are trained in parallel for the same task. This ensembling approach is referred to as *bagging*. *Boosting*, on the other hand, is an alternative ensembling strategy that trains models serially. With this approach, a simple model is first trained on the original training set, and the second model is then trained to improve the errors that the first model had made (in hopes that the second model would then be able to improve the first one). This is accomplished by (1) training the second model to predict the first model's residuals (i.e., the differences between the first model's predictions and the actual labels) and (2) by reweighting the training set so that the samples with high residuals receive more attention during the second round of training. This process is repeated, i.e., models are added to improve the previous ensemble's errors, until a certain stopping criterion is fulfilled (Schapire, 1990, 2003).

*Gradient tree boosting*, as the name suggests, is an approach that implies “boosting” the decision trees. It considers boosting as an optimization problem and uses gradient descent to solve this problem (Mason et al., 1999). Compared to random forests, gradient tree boosting suffices with fewer trees; therefore, the computational cost is usually much lower. Gradient tree boosting also often leads to highly accurate models, although sometimes with a higher tendency of overfitting when compared to random forests.

#### Advantages

- Easy to apply; good results also without hyperparameter tuning.
- Built-in feature selection (in particular, the regularized variants); thus, insensitive to irrelevant/noisy features.
- For numerical features, only the order of the values is relevant; thus, insensitive to input scaling; no normalization required.
- Feature importances can be computed.

#### Disadvantages

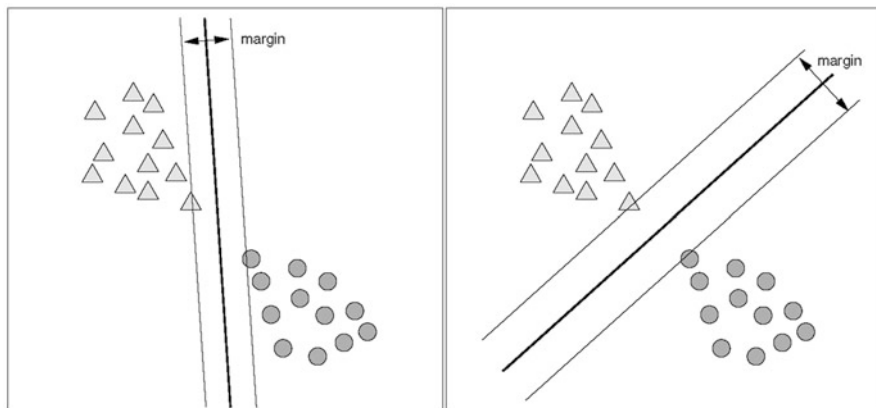
- Higher tendency to overfit than random forests do.

### 1.4.7 Support Vector Machines

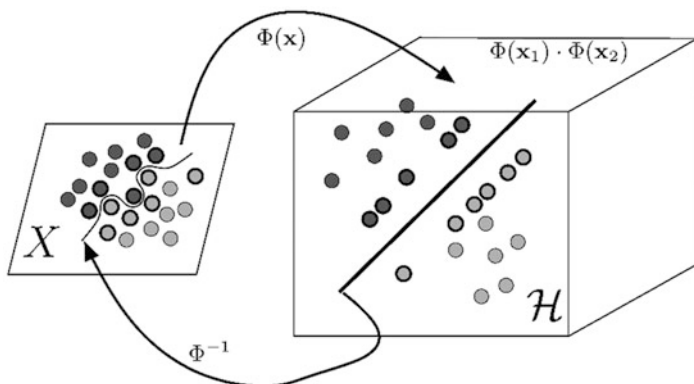
In their simplest form, *support vector machines (SVMs)* are linear classifiers for binary classification. They separate a training set precisely in such a way that the distance of the separating hyperplane to the closest training sample is maximized. Provided that the training set consists of two classes that are perfectly separable (i.e., a linear classifier that separates the two classes with no errors exists), this maximization of the minimum distance is equivalent to rotating the separating hyperplane so that the “tube” between the two classes (also known as *margin* in the following) is as wide as possible. The following images aim to illustrate this principle (Fig. 4).

It is clear that a large distance to either class is desirable; if we expect new samples to predominantly appear at the same location as the training samples, the risk of misclassifying new samples decreases if the classification boundary is sufficiently far away. While this is only a vague intuitive explanation, margin maximization helps to minimize the misclassification rate on future data (or, to be more precise, an upper bound thereof, but that is a theoretical detail falling outside of this scope; see Cortes & Vapnik, 1995; Vapnik, 1998). Margin maximization leads to a constrained quadratic optimization problem for which a global solution that can be efficiently determined exists.

Naturally, the abovementioned principle only works under the unrealistic assumption that the training set is linearly separable. To overcome this limitation,



**Fig. 4** The principle of margin maximization. In the left panel, the margin is not optimal, whereas in the right panel, the separating hyperplane is properly rotated to maximize the margin



**Fig. 5** The idea behind a non-linear SVM—the data is projected to a higher-dimensional space  $H$  using a non-linear mapping  $\phi$  in hopes that the data becomes separable in this high-dimensional space

it is possible to relax the margin maximization problem by adding a term that penalizes violations of the margin constraints, leading to the so-called *soft-margin SVM*. In this way, the resulting optimization problem can be solved as efficiently as the standard margin maximization problem (Cortes & Vapnik, 1995).

SVMs can also be generalized to non-linear classifiers quite easily; one only has to specify a *kernel*, that is, a two-place function for computing the similarity between inputs. In more formal terms, the kernel corresponds to a scalar product in a high-dimensional space to which the inputs are transformed by a possibly non-linear mapping (Schölkopf & Smola, 2002), as depicted schematically in Fig. 5. The underlying mathematical theory ensures that neither the high-dimensional space  $H$  nor the non-linear mapping  $\phi$  need to be considered. As such, all that needs to be

accounted for is a suitable kernel function, which has to be chosen a priori according to the application. This is commonly referred to as the *kernel trick*.

For numerical data, the following standard kernel functions are commonly used:

<b>Linear kernel:</b>	$k(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^d x_j \cdot x'_j$
<b>Polynomial kernel:</b>	$k(\mathbf{x}, \mathbf{x}') = \left( \sum_{j=1}^d x_j \cdot x'_j + \alpha \right)^\beta$
<b>Radial basis function (RBF) kernel:</b>	$k(\mathbf{x}, \mathbf{x}') = \exp \left( -\frac{1}{2\sigma^2} \sum_{j=1}^d (x_j - x'_j)^2 \right)$

The RBF kernel (a.k.a. Gauss kernel) is especially suitable in many practical situations and is, therefore, very common. However, SVMs can be applied to any type of data, including unstructured data, if an appropriate and suitable kernel is available. This is why SVMs excel in, for instance, classifying biological sequences (Ben-Hur et al., 2008; Palme et al., 2015).

Interestingly, regardless of whether the linear kernel, which corresponds to the original linear SVM, or a non-linear kernel is used, the internal optimization algorithm always remains the same. In either case, the final SVM is a classification function that assigns a real-valued score to a new input sample  $\mathbf{x}$ :

$$g(\mathbf{x}) = b + \sum_{i=1}^l \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i)$$

Thus, for a new input sample  $\mathbf{x}$ , the similarity to each training input sample  $\mathbf{x}_i$  as well as a sum is computed, where each similarity value  $k(\mathbf{x}, \mathbf{x}_i)$  is weighted by a non-negative weighting factor  $\alpha_i$  and the true class  $y_i \in \{-1, +1\}$  of each training sample. Finally, a constant offset  $b$  is added. This  $b$  and the factors  $\alpha_1 \cdots \alpha_l$  are the actual parameters of the SVM that were fitted to the data during training. Note that the score  $g(\mathbf{x})$  is not a probability but, rather, an abstract value that can have both signs. Usually, a zero threshold is applied, and samples with a positive score are assigned to the positive class, while samples with a negative score are assigned to the negative class.

The most common form, the *C-SVC* (C-support vector classifier), only has one hyperparameter, the cost factor  $C$ , which is merely an upper bound of the factors  $\alpha_i$ . The cost factor  $C$  controls the overall influence of a single training sample on the final classification, indicating that a  $C$  too large may lead to an overfitted SVM, while a  $C$  too small may lead to an underfitted SVM. Hence, hyperparameter selection needs to be applied in order to find the optimal value for  $C$ . The fact that the C-SVC only has one hyperparameter makes hyperparameter optimization relatively easy, and even though the use of the RBF kernel introduces another hyperparameter to be optimized, two hyperparameters are still easy enough to handle.



As previously noted, SVMs are only able to perform binary classification. In order to deal with multi-class problems, one would need to transform a multi-class problem into multiple binary classification problems. To do so, the most common strategy is to train one SVM for each pair of classes and aggregate the multitude of the classification results by majority voting (Hastie & Tibshirani, 1998). Finally, it is relevant to note that there is also an SVM extension that transforms the abstract score  $g(\mathbf{x})$  into an estimate of the posterior class probability  $p(y = j | \mathbf{x})$  (Platt, 1999; Wu et al., 2004).

#### Advantages

- Low number of hyperparameters; thus, relatively easy hyperparameter optimization.
- Optimization algorithm gives a guaranteed global solution.
- Work well on small data sets.
- Good theoretical foundation.

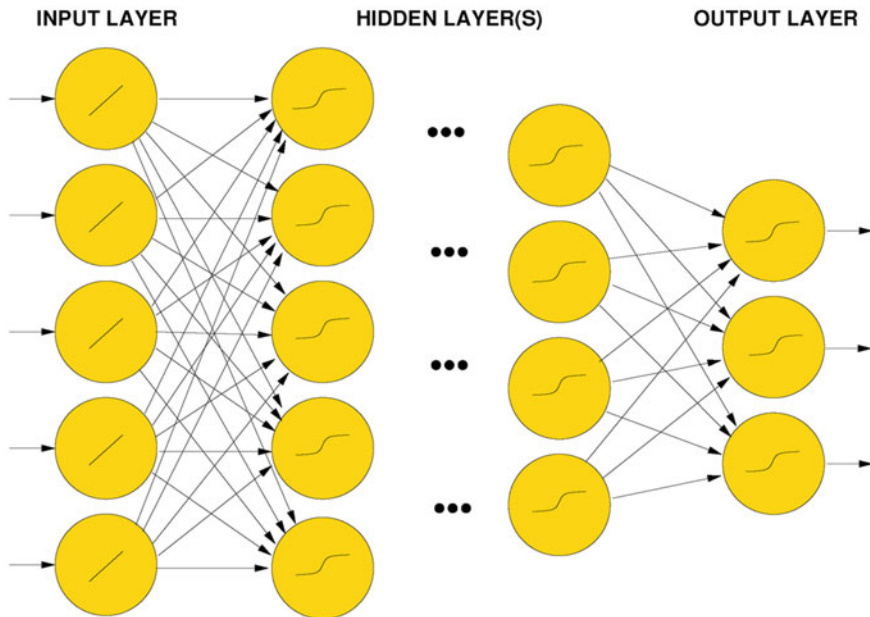
#### Disadvantages

- Sensitive to input scaling; thus, normalization is advisable.
- Sensitive to irrelevant/noisy features; thus, feature selection is advisable.
- Choice of kernel depends on application; may not be straightforward.
- Computationally expensive for larger data sets.

### 1.4.8 Artificial Neural Networks

Last but not least, we introduce the basic concept of *artificial neural networks* (ANNs). There is an abundance of different variants, but what they all have in common is that they involve massively parallel systems interconnecting computational units. Figuratively speaking, these units can be understood as simple models of neural cells/neurons. Although ANNs have been studied since the 1960s, the real breakthrough could be seen within the last ten years. Due to new methods for training ANNs, along with the availability of massive data sets and appropriate computational resources, it has become feasible to train deep neural networks. As a consequence, a whole new field known as Deep Learning has emerged and ignited the Artificial Intelligence (AI) hype of the past decade.

In this chapter, we will concentrate solely on so-called *fully connected networks* (FNNs), i.e., networks that consist of layers of artificial neurons. Such networks process their inputs in a feed-forward fashion, and the neurons have no



**Fig. 6** Schematic illustration of a fully connected neural network (FNN) with hidden layers

memory or state.<sup>3</sup> As laid out above, this is precisely what is sufficient and meaningful for tabular data. For other types of data such as images or sequences (including speech or language), other types of ANNs, for example, convolutional neural networks (CNNs) or recurrent neural networks (RNNs) are used. For an up-to-date view of all available variants, refer to the literature (Goodfellow et al., 2016; LeCun et al., 2015; Schmidhuber, 2015).

As previously mentioned, an FNN consists of multiple layers of static artificial neurons. Figure 6 shows a simple schematic illustration of such a network.

When the network receives a new input  $\mathbf{x}$ , each of the input features is fed into one unit from the input layer (the network in Fig. 6 has  $d = 5$  input features). While the input layer is merely a dummy that outputs its input, the subsequent layers all have the same structure. Each neuron computes its output as a weighted sum of the outputs of the neurons from the previous layer in addition to a so-called bias weight (since all the neurons from one layer forward their outputs to all the neurons in the next layer, such networks are called *fully connected*). Finally, each neuron applies an *activation function* to this weighted sum. More formally, the output of one neuron is computed as follows:

<sup>3</sup>Such networks have often been called *multi-layer perceptrons (MLPs)*, though this can now be considered an outdated term.

$$\varphi\left(w_0 + \sum_i w_i \cdot x_i\right)$$

In this formula, the values  $x_i$  are the outputs of all the neurons from the previous layer,  $w_i$  are the *connection weights*,  $w_0$  is the *bias weight*, and  $\varphi$  is the activation function. The most traditional activation function is the sigmoid (refer to section “Logistic Regression”):

$$\varphi(x) = \frac{1}{1 + e^{-x}}$$

Since the onset of Deep Learning, the following activation functions have been established and are now considered standard:

- **Rectified Linear Units (ReLU; Glorot et al., 2011):**  $\varphi(x) = \max(0, x)$ .
- **Exponential Linear Units (ELU; Clevert et al., 2016):**  $\varphi(x) = \begin{cases} x & x \geq 0 \\ e^x - 1 & x < 0 \end{cases}$

When using a fully connected neural network for classification, the following two cases can appear:

- If we have a binary classification task, the output layer consists of one neuron with a sigmoid activation, where the weights are trained to minimize the binary cross-entropy on the training set. As mentioned in the section “Logistic Regression,” the result can then be interpreted as an estimate of the posterior probability  $p(y = 1 | \mathbf{x})$ .
- If we have a multi-class task, the output layer has as many neurons as there are classes, i.e., one neuron per class (thus, Fig. 6 shows a network that could be used for a 3-class classification task). In this case, a special activation function called *softmax* is used. Here, an exponential function is first applied to the weighted sums of all the output neurons, and then it normalizes the transformed values to a sum of 1 in order to ensure that the final output of each neuron can be interpreted as an estimate of the posterior probability  $p(y = j | \mathbf{x})$ . The weights are trained to minimize the multi-class cross-entropy.

The only question remaining is how the weights are actually trained for a training set. The answer: this is usually done via some variant of *gradient descent*. Generally speaking, the derivatives of a differentiable loss function (in the case of classification, some variant of categorical cross-entropy) with respect to the weights in the network are first computed. Then, the weights are slightly adapted towards the direction of the negative derivatives, which corresponds to moving down the loss surface in the direction of the steepest descent. Though it seems complicated to compute the derivatives, it is possible to compute them by applying a relatively simple layer-wise decomposition that exploits the chain rule. With regard to the latter, the procedure in which derivatives are computed by propagating an error term backwards through a network is commonly called *backpropagation* (although this

term was popularized by Rumelhart et al. (1986), this principle already appeared in many contexts before them; for an overview, see Goodfellow et al., 2016; Schmidhuber, 2015).

Nowadays, it is rarely common to optimize weights by minimizing the global loss function, which considers all training samples simultaneously. This would result in what is known as *batch learning*, a situation in which the weights are only adapted once after each training epoch (i.e., after one pass through the entire training set). Instead, it has turned out beneficial to estimate the gradients from small subsets of the training set, so-called *minibatches*, and to adapt the weights after each minibatch. This strategy is commonly referred to as *stochastic gradient descent (SGD)*. However, most neural network implementations have ceased to use standard SGD and have replaced it with more advanced optimization algorithms, such as RMSProp (Hinton, 2012) or ADAM (Kingma & Ba, 2015).

When it comes to a neural network, the connection weights and the bias weights are considered the trainable parameters, while the network design, i.e., the number of layers, the numbers of neurons in each layer, and the chosen activation functions in the hidden layers, are hyperparameters. These are further complemented by the choice of an optimization algorithm and its hyperparameters, such as learning rate, minibatch size, and total number of epochs. As such, artificial neural networks have a host of hyperparameters; hence, hyperparameter optimization is a complex matter, and grid search is typically infeasible.

### Advantages

- Can exploit the subtlest patterns in the data, provided that the data sets are large enough.
- Work well for large data sets.
- Training and prediction can be sped up immensely by using graphical processing units (GPUs).

### Disadvantages

- Large number of hyperparameters; thus, difficult hyperparameter optimization.
- Sensitive to input scaling; thus, normalization is advisable.
- Black-box models, i.e., it is very difficult to interpret how an ANN makes its predictions.
- Do not work well on small data sets.

## 2 Practical Demonstration

### 2.1 Use Case

Consider a hotel that focuses on sporty guests as the target group and offers various vacation packages and sports courses. Moreover, suppose that the hotel uses web tracking to record the page views of the offers on its website and also records the bookings made. With the help of this data, a classification model should be developed that predicts whether or not a visitor will make a booking. This prediction should be based on the following input below, determined by the visitor's surfing behavior on the hotel's website:

- How often was the site visited?
- How many pages were viewed?
- In which categories were the offers that the visitors viewed?

Such a prediction would enable the hotel to display offers on its website in a personalized way, thus especially prompting visitors for whom the model predicts as high potential. As such, this tourism use case relates closely to *lead scoring*.

In the following, we will describe how to solve this use case in detail using the concepts and methods described earlier on in this chapter. This includes pre-processing the data and, subsequently, applying different classification methods as well as evaluating and comparing them. We will present the most essential steps and results here, while the implementation details (all Python code plus results) can be found in the attached Jupyter notebook.

### 2.2 The Data Set

The data set used in the accompanying notebook is based on real data but has been anonymized for privacy reasons and slightly modified for demonstrative purposes. It is available as a CSV file and contains 138,760 lines. Each line represents one page view or booking. To read the data file and for basic manipulations of the tabular data, we will use the well-known "Pandas" package (McKinney, 2010).

Table 5 shows a preview of the first 10 rows of the data set. The seven columns in this table correspond to the following features:

- **type:** shows whether it was a simple page view or a booking entry
- **page\_id:** ID of the called page
- **category:** category of the vacation offer on the given page
- **price:** price of the offer
- **user\_id:** unique (anonymous) ID of the visitor obtained from tracking (fingerprinting)
- **user\_session:** ID of the visitor's session
- **daytime:** date and time of the event

**Table 5** First 10 rows/samples of the dataset

	type	page_id	category	price	user_id	user_session	daytime
0	View	1005135	Others	748	535871217	c6bd7419-2748-4c56-95b4-8cec9ff8b80d	2019-10-03 00:00:19
1	View	1004767	Corss-country skiing	255	512558158	9a206ba2-37c7-4354-9d31-37ff3bb297ed	2019-10-16 00:00:36
2	View	1005135	Swimming	748	535871217	c6d7419-2748-4c56-95b4-8cec9ff8b80d	2019-10-03 00:00:43
3	View	1002544	Run	464	532085144	77ae546a-542b-414c-a01b-c5ceca7e99cf	2019-10-06 00:00:44
4	View	1005105	Swimming	415	529755884	0b828fb6-99bd-4d26-beb3-302ff5d6102c	2019-10-18 00:00:50
5	View	3701062	Swimming	90	515342595	0e30e1c0-4d3e-4e1a-90e3-ab93b5f5c1a2	2019-10-16 00:01:00
6	View	1004836	Triathlon	131	555447788	94c1a98c-41a3-401e-ad99-439beac4495c	2019-10-20 00:01:06
7	View	1004836	Run	241	546259103	6e2984c8-502e-4fe7-bbba-34087f760175	2019-10-22 00:01:11
8	View	1004856	Cycling	131	515757896	4938043e-e50f-44ad-944d-958d04df62d6	2019-10-20 00:01:17
9	View	1801551	Triathlon	463	515511944	d63ef339-2d6a-411a-95bd-d58c77ae6e4f	2019-10-23 00:01:29

Based on this data set, we will create profiles of visitors, which can then be used for modeling and, more specifically, for training the classification models that should predict whether a visitor will eventually book.

### 2.3 Descriptive Analysis of the Raw Data Set

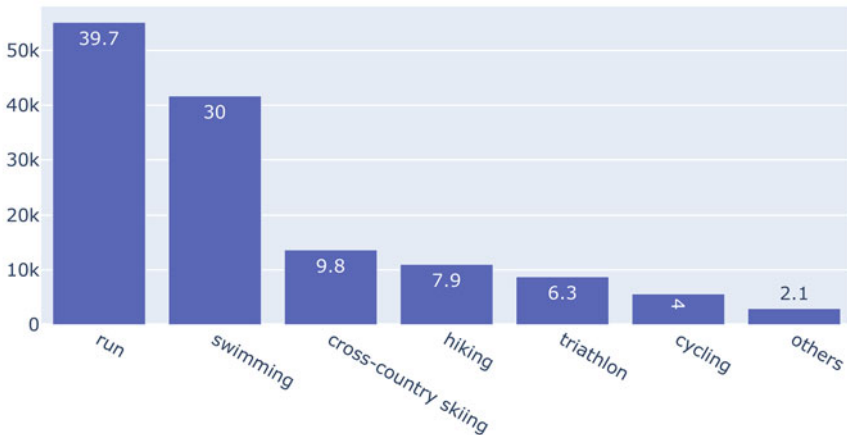
We first try to get an overview of the data by producing some insightful graphics and key figures. Looking at the feature “type,” it seems that there are three different categories: page views (view), shopping cart actions (cart), and bookings (book). Figures 7 and 8 highlight the distribution of these values.

## Types of Events (%)



**Fig. 7** Distribution of the types of events. It becomes clear from this diagram that the overall conversion rate is 3.5%

## Categories of offers in events (in %)



**Fig. 8** Distribution of page views per category

The hotel offers services for different types of sports. Their distribution relating to the set of events is presented in Fig. 8.

From this graph, it can be observed that the majority of visitors look for offers from the “running” category followed by “swimming.” The remaining categories have much fewer page views. Nonetheless, the distribution of bookings per category is actually quite similar (see notebook).

The accompanying Jupyter notebook shows an additional graph with the distribution of page views and bookings over time. Despite some random fluctuations, the numbers of page views and bookings remain relatively constant, even when taking individual categories into consideration. Therefore, we assume that (1) it is

unnecessary to take date/time into account when constructing our classification models, and (2) the models will be valid and perform correctly as long as the market is stable.

## 2.4 Aggregation of Data: Creation of User Profiles

Almost 45,000 interested customers visited the website during the time period under evaluation. This, together with the number of visits, shows that the majority of visitors visited the site only once. As not enough information is provided from visitor behavior to make a meaningful prediction model for such visitors, in the subsequent analyses we will focus our attention on those visitors who viewed the hotel website at least five times. For these, we can create profiles and use them to construct classification models.

Visitors are identified in the data records by a unique ID, even for visits on different days, and we summarized each visitor's data into a separate profile. This profile consists of the number of sessions, the number of page views, and the number of bookings. From there, we could also infer whether or not the user booked. In addition, we also computed total proportions of the user and visited pages regarding the seven sports categories for later use.

Table 6 shows some user profiles with the features for page counts and percentages in the different page categories. The features "Sessions" and "PagesPerSession" can be considered the most basic features used in classification, which will later be augmented by the seven proportions (the seven rightmost features in Table 6). The column "Booked" will become the target of the following classification efforts; in other words, we will try to predict this column from the input columns. Note that the *distribution of the two classes is asymmetrical*: 68.3% of users are non-bookers, while 31.7% are bookers.

Let us illustrate an overview of this data by creating a scatterplot of the number of visits and page views per visit (Fig. 9).

We can see that the red and blue dots are more concentrated in certain areas of this "feature space," which is a good sign as it signifies that the two input features (location of dots on the graph) are somehow related to the class that is to be predicted (the color of the respective dot). In the following, we will train classification models that divide the features space into areas for bookers and non-bookers and create rules to recognize bookers based on their behavior and actions taken on the website.



**Table 6** First 10 user profiles

user_id	Pageviews	Sessions	Items Booked	PagesPer Session	Booked	hiking	cross-country skiing	run	triathlon	swimming	cycling	others
497087183	24	6	0.0	4.000000	0	0.000000	0.083333	0.333333	0.083333	0.500000	0.000000	0.000000
499826904	6	5	0.0	1.200000	0	0.000000	0.166667	0.666667	0.000000	0.000000	0.166667	0.000000
512364387	10	6	1.0	1.666667	1	0.000000	0.000000	0.300000	0.100000	0.400000	0.100000	0.100000
512397324	21	5	0.0	4.200000	0	0.095238	0.095238	0.238095	0.047619	0.380952	0.047619	0.095238
512407916	33	5	1.0	6.600000	1	0.000000	0.090909	0.151515	0.363636	0.090909	0.151515	
512424366	25	7	0.0	3.571429	0	0.040000	0.160000	0.520000	0.040000	0.200000	0.000000	0.040000
512426470	12	6	0.0	2.000000	0	0.083333	0.000000	0.500000	0.083333	0.333333	0.000000	0.000000
512486232	11	5	1.0	2.200000	1	0.181818	0.181818	0.181818	0.181818	0.090909	0.090909	0.090909
512515918	11	6	0.0	1.833333	0	0.000000	0.181818	0.272727	0.000000	0.363636	0.181818	0.000000
512558829	15	5	0.0	3.000000	0	0.133333	0.000000	0.466667	0.133333	0.266667	0.000000	0.000000

## Classification of website users / booking (red) and non booking (blue)

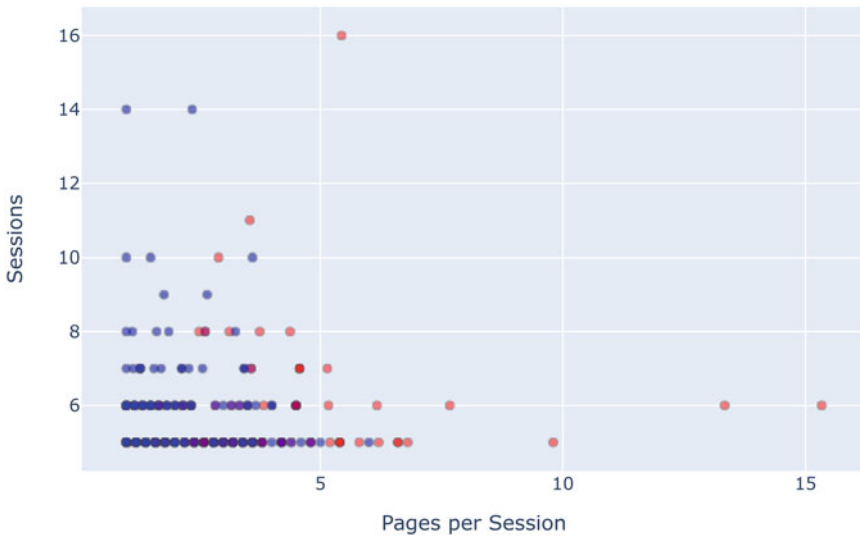


Fig. 9 Visits vs. page views per visit for bookers and non-bookers

## 2.5 Classification of Visitors: Model Building

For the training and evaluation of the classification models, we will use the machine learning framework scikit-learn.<sup>4</sup>

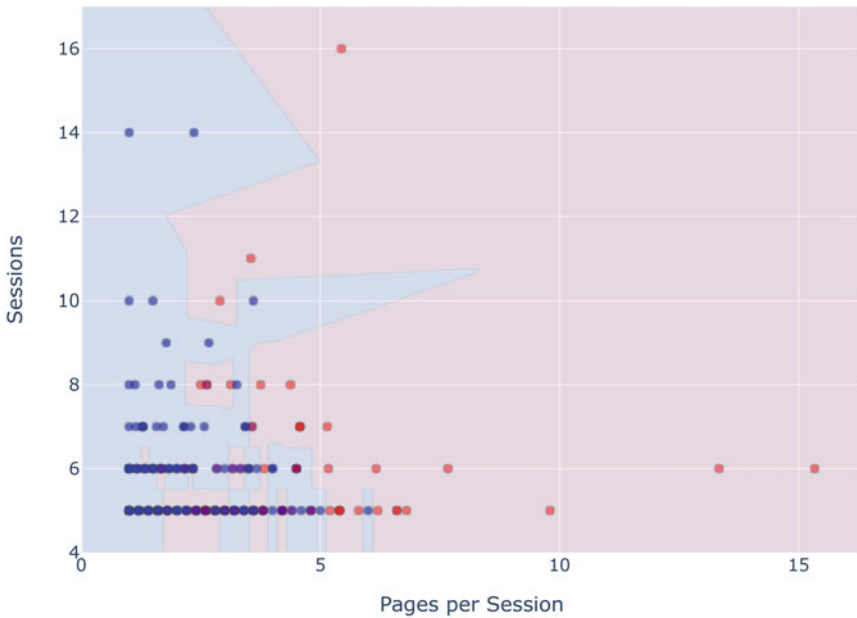
We first have to split the data set into training and test data (hold-out method, see above) in which we use 70% of the data for training and the rest for evaluation. In this example, these comprise of 176 and 76 data samples, respectively. Then, we can use the same models as described in the section “Classification Methods”:

- k-nearest neighbor (KNN) classification
- Logistic regression
- Naïve Bayes classification
- Decision trees
- Random forest
- Gradient tree boosting
- Support vector machine classification
- Artificial neural networks

For each of these methods, we implemented the classification based on two features and, for comparison, additionally added a classification with all features in the Jupyter notebook. This aids in developing an intuition for the procedures and

<sup>4</sup><https://scikit-learn.org>; Pedregosa et al. (2011)

## Classification of Website users / booking (red) and non booking (blue)



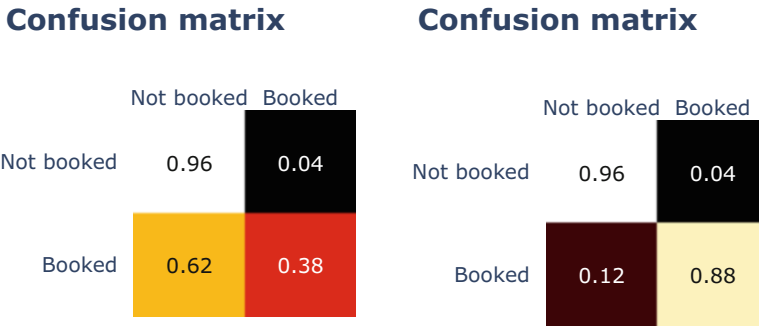
**Fig. 10** Nearest neighbor classification

assessing risks, such as overfitting, better. Fig. 10 provides an illustration of such a plot in which the classification boundaries as obtained by the nearest neighbor classifier (i.e.,  $k = 1$ ) for the simple two-feature setting are exemplified.

Thereafter, we trained and evaluated models with a higher number of features by also taking the proportions of the seven sports categories into account. All models were finally evaluated on a test set by computing the confusion tables as well as the multiple measures of classification performance deduced from them (see section “Confusion Matrix and Evaluation Measures Computed Therefrom”). An example of such confusion matrices is provided in Fig. 11.

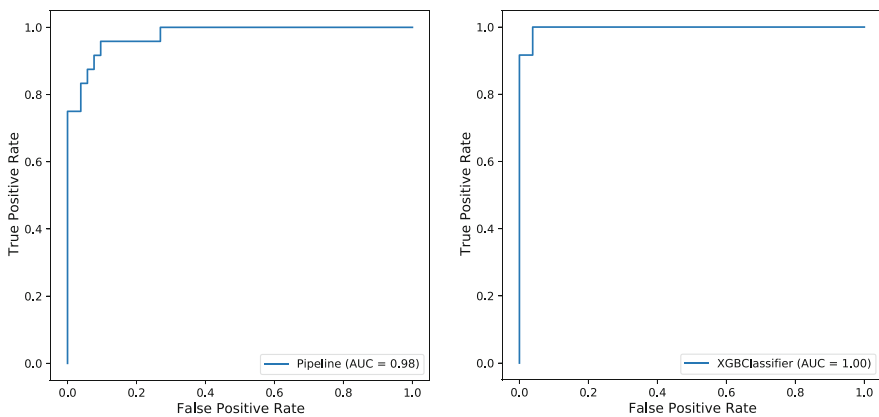
Clearly, using all the features gives us much better results than using only the two basic input features. This could also be observed across all classification methods. We performed normalization where it was necessary and meaningful, and we also proceeded with hyperparameter selection for  $k$ -nearest neighbor classification and for support vector machine classification. Lastly, all models except the  $k$ -nearest neighbor classification were additionally evaluated in terms of their ranking performance through the computation of ROC curves and the corresponding AUC measures.

All models performed relatively poorly on the two-dimensional data set, with accuracies ranging between 75% and 80%. Since this is an unbalanced data set, it is worthwhile to consider balanced accuracies as a more meaningful measure of classification performance. The balanced accuracies ranged between 64% and 69%. Interestingly, the best results were obtained via linear SVM.



**Fig. 11** Confusion matrices for k-nearest neighbors classifier with two features (left) and with all features (right). Note that the confusion tables do not show absolute numbers of true positives, false positives, etc. but, rather, the corresponding rates (true positive rate, false positive rate, etc.)

When using all nine features, the situation changed completely. More or less all of the classification methods performed much better than at random (note that random guessing or assigning all samples to the non-bookers class would result in an accuracy of 68.3%), with accuracies ranging from 92% (Naïve Bayes) to 97% (random forest and gradient tree boosting) and balanced accuracies ranging from 91% (Naïve Bayes) to 98% (gradient tree boosting). Regarding ranking performance, the best AUC values were close to 1.00 (random forest and gradient tree boosting). Surprisingly, logistic regression also performed very well with an accuracy of 0.96, a balanced accuracy of 0.95, and an AUC of 0.99. Moreover, non-linear SVMs and artificial neural networks were almost on par, at least in terms of ranking performance (all three methods showed an AUC of 0.99). Figure 12 demonstrates the ranking performance of different classifiers based on their ROC curves.



**Fig. 12** ROC curves for a neural network (left) and a boosted tree ensemble (right). Both classifiers were trained on all nine features

We would like to emphasize that the use of artificial neural networks in the accompanying Jupyter notebook is for demonstration purposes only. Artificial neural networks are currently the most powerful methods in machine learning and are mainly used for large data sets. The data set used for this use case is relatively small; therefore, neural networks do not appear to be a very promising approach. Since the purpose of this example is merely to act as a teaser, we used the implementation of scikit-learn here as well (Pedregosa et al., 2011), even though more advanced implementations are available in packages such as Tensorflow/Keras (<https://www.tensorflow.org/>; Abadi et al., 2015; Chollet, 2018) or PyTorch (<https://pytorch.org/>; Paszke et al., 2019).

### 2.5.1 Summary of Results

All details and results can be found in the accompanying Jupyter notebook, which contains the full source code and results, but also further references relating to this chapter and comments about implementation details. The general results can be summarized as follows:

1. Using only two features allows us to visualize the classification boundaries nicely, but it does not give good results.
2. All methods perform much better when using the full data set with all features.
3. All methods give good accuracies, but the best results were obtained via tree ensembles, followed by SVM, ANNs, and logistic regression.
4. Normalization of the features has a massive influence on the performance of some of the methods.

## 2.6 *Application of the Models*

After having trained a successful model, we can then put it into practice. This can be done by recording user interactions on the website, again with the tracking solution, and using the model to classify visitors into potential bookers and non-bookers. The model can establish this in real-time while the visitor is viewing the site.

If, for example, a visitor is classified as a booker based on his/her behavior, although he/she has not yet booked, one can try to persuade him/her to book by addressing him/her directly and offering special promotions or discounts. As such, the classification allows for a differentiated reaction to various visitor groups by, for instance, displaying personalized content.

**Service Section**

**Main Application Fields:** Classifications are relevant for any field in which objects should be categorized or predictions of categories should be made. In tourism, typical use cases include, but are not limited to, the following:

- Lead scoring
- Opportunity scoring
- Cross-/upselling
- Churn prediction
- Prediction of ratings
- Classification of hosts
- Distinguishing tourists from non-tourists in passive mobile data

**Limitations and Pitfalls:** The methods presented in this chapter are used exclusively to identify classification models for which sufficient data are available. This data must consist of the objects to be classified and be characterized by a sufficiently large and meaningful set of input features and their labels (i.e., their true categories). Whether the amount of data and the set of features are sufficient for a decent classification performance cannot be answered a priori but, rather, needs to be evaluated empirically by actually training the classification models and evaluating the results.

The following pitfalls should be avoided:

- When pre-processing the data, biases in test sets must be avoided.
- In order to avoid under- or overfitting, a thorough hyperparameter optimization should be carried out for some of the methods (in particular, k-nearest neighbor, support vector machines, and artificial neural networks).
- It is imperative to perform hyperparameter optimization using a validation set or by using cross validation on the training set. The test set should only be consulted when the model is final; otherwise, the selection of the hyperparameters is biased to the test set, which renders the test set useless.

**Similar Methods and Methods to Combine with:** Regression (see Chap. 11) is a similar approach, with the only difference being that a number, rather than a category, is predicted. In some cases, the user has a free choice of whether to use classification or regression. In our hotel booking example, for instance, we can predict whether a customer books/does not book ( $\rightarrow$  classification), but we could also decide to solve the task by trying to predict the actual amount of bookings or the total price of booked items ( $\rightarrow$  regression).

Classification methods can, or must, be combined with other methods, such as:

(continued)

- Unsupervised methods for pre-processing data (e.g., dimensionality reduction)
- Methods for extracting features from unstructured data (e.g., images, text)

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-10-Classification>

## Further Readings and Other Sources

Logistic regression: <https://medium.com/data-science-group-iitr/logistic-regression-simplified-9b4efe801389>

Naïve Bayes: <https://medium.com/x8-the-ai-community/a-simple-introduction-to-naive-bayes-23538a0395a>

Decision trees: <https://medium.com/swlh/a-beginners-guide-to-decision-trees-84ca34927818>

Random forests: [https://medium.com/@harshdeepsingh\\_35448/understanding-random-forests-aa0cceedbbbb](https://medium.com/@harshdeepsingh_35448/understanding-random-forests-aa0cceedbbbb)

Gradient tree boosting: <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>

Support vector machines: <https://medium.com/@LSchultebraucks/introduction-to-support-vector-machines-9f8161ae2fcb>

Artificial neural networks: <https://medium.com/@purnasaigudikandula/a-beginner-intro-to-neural-networks-543267bda3c8>

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*.
- Allison, P. D. (2002). *Missing data. Quantitative applications in the social sciences* (Vol. 136). SAGE Publications.
- Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., & Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS Comput. Biol.*, 4(10), e1000173. <https://doi.org/10.1371/journal.pcbi.1000173>
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (Eds.). (1984). *Classification and regression trees*. CRC Press.
- Chollet, F. (2018). *Deep learning with python. Safari tech books online*. Manning Publications.
- Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (ELUs). In *Proceedings of the Fourth International Conference on Learning Representations*. San Juan, Puerto Rico.
- Cortes, C., & Vapnik, V. N. (1995). Support vector networks. *Machine Learning*, 20, 273–297.

- Cox, D. R. (1966). Some procedures connected with the logistic qualitative response curve. In F. N. David (Ed.), *Research papers in probability and statistics (festschrift for J. Neyman)* (pp. 55–71). John Wiley & Sons.
- Cramer, J. S. (2002). *The origins of logistic regression* (Tinbergen institute working paper no. 2002-119/4). <https://doi.org/10.2139/ssrn.360300>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861–874.
- Fix, E., & Hodges, J. (1951). Discriminatory analysis. *Nonparametric discrimination; consistency properties*. Randolph Field, TX.
- Florkowski, C. M. (2008). Sensitivity, specificity, receiver-operating characteristic (ROC) curves and likelihood ratios: Communicating the performance of diagnostic tests. *Clinical Biochemist Reviews*, 29(Suppl. 1), S83–S87.
- Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics* (pp. 315–323).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hand, D. J., & Yu, K. (2001). Idiot's Bayes: Not so stupid after all? *International Statistical Review*, 69(3), 385. <https://doi.org/10.2307/1403452>
- Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *The Annals of Statistics*, 26(2), 451–471.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction (second)* Springer series in statistics. Springer.
- Hinton, G. (2012). *Neural networks for machine learning online course*. Retrieved from [http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf)
- Hoerl, A. E. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58, 54–59.
- Horton, N. J., & Kleinman, K. P. (2007). Much ado about nothing: A comparison of missing data methods and software to fit incomplete data regression models. *The American Statistician*, 61(1), 79–90. <https://doi.org/10.1198/000313007X172556>
- John, G. H., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In P. Besnard & S. Hanks (Eds.), *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence* (pp. 338–345). Morgan Kaufmann.
- Jolliffe, I. (2014). Principal component analysis. In B. S. Everitt & D. Howell (Eds.), *Encyclopedia of statistics in behavioral science*. John Wiley & Sons. <https://doi.org/10.1002/9781118445112.stat06472>
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3th International Conference for Learning Representations*, San Diego, CA.
- Le Cessie, S., & van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C*, 41(1), 191. <https://doi.org/10.2307/2347628>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Luntz, A., & Brailovsky, V. (1969). On estimation of characters obtained in statistical procedure of recognition. *Techicheskaya Kibernetica*, 3. (in Russian).
- Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). *Boosting algorithms as gradient descent. Advances in neural information processing systems* (Vol. 12, pp. 512–518). MIT Press.
- McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the Python in Science Conference, Proceedings of the 9th Python in Science Conference* (pp. 56–61). SciPy. <https://doi.org/10.25080/Majora-92bf1922-00a>
- Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.
- Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11, 169–198. <https://doi.org/10.1613/jair.614>
- Oskolkov, N. (2021). Dimensionality reduction. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications*. Springer.



- Palme, J., Hochreiter, S., & Bodenhofer, U. (2015). KeBABS: An R package for kernel-based analysis of biological sequences. *Bioinformatics*, *31*(15), 2574–2576. <https://doi.org/10.1093/bioinformatics/btv176>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* *32* (pp. 8024–8035). Curran Associates.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.
- Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In P. J. Bartlett, B. Schölkopf, D. Schuurmans, & A. J. Smola (Eds.), *Advances in large margin classifiers* (pp. 61–74). MIT Press.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.
- Ramos-Henríquez, J. M., Gutiérrez-Taño, D., & Díaz-Armas, J. (2021). Value proposition operationalization in peer-to-peer platforms using machine learning. *Tourism Management*, *84*, 104288. <https://doi.org/10.1016/j.tourman.2021.104228>
- Reif, J., & Schmücker, D. (2020). Exploring new ways of visitor tracking using big data sources: Opportunities and limits of passive mobile data for tourism. *Journal of Destination Marketing and Management*, *18*, 100481. <https://doi.org/10.1016/j.jdmm.2020.100481>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. <https://doi.org/10.1038/323533a0>
- Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227. <https://doi.org/10.1007/BF00116037>
- Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. C. Holmes, B. Mallick, & B. Yu (Eds.), *Lecture notes in statistics: Vol. 171. Proceedings MSRI workshop on nonlinear estimation and classification* (pp. 149–171). Springer.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels. Adaptive computation and machine learning*. MIT Press.
- Stekhoven, D. J., & Bühlmann, P. (2012). Missforest – Non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112–118. <https://doi.org/10.1093/bioinformatics/btr597>
- Stöckl, A., & Bodenhofer, U. (2021). Regression. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications*. Springer.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *The Journal of the Royal Statistical Society, Series B*, *58*(1), 267–288.
- Tran, N., Schneider, J.-G., Weber, I., & Qin, A. K. (2020). Hyper-parameter optimization in classification: To-do or not-to-do. *Pattern Recognition*, *103*, 107245. <https://doi.org/10.1016/j.patcog.2020.107245>
- Vapnik, V. N. (1998). *Statistical learning theory. Adaptive and learning systems*. Wiley Interscience.
- Veloso, B. M., Leal, F., Malheiro, B., & Burguillo, J. C. (2019). On-line guest profiling and hotel recommendation. *Electronic Commerce Research and Applications*, *34*, 100832. <https://doi.org/10.1016/j.elerap.2019.100832>
- Wu, T.-F., Lin, C.-J., & Weng, R. C. (2004). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, *5*, 975–1005.
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *The Journal of the Royal Statistical Society, Series B*, *67*(2), 301–320.

# Regression



## Data-Driven Modeling of Numerical Relationships in Tourism

Andreas Stöckl and Ulrich Bodenhofer

### Learning Objectives

- Illustrate the use of regression methods in tourism
- Explain the methods for regression
- Explain the methods for evaluating regression results
- Demonstrate how various methods can be applied to a regression task in Python using scikit-learn and Jupyter

## 1 Introduction and Theoretical Foundations

### 1.1 Motivation and Basic Concepts

*Regression* is known as the task of assigning a numerical value to an object based on this object's set of characteristics. In statistical terms, regression corresponds to estimating a numerical quantity based on dependent variables. Such tasks arise in various practical settings, particularly when it comes to *forecasting* all types of values in the real world. Demand forecasting is *the* popular example in tourism (Chen & Wang, 2007; Claveria et al., 2015; Law et al., 2019; Xie et al., 2021). Other applications include forecasting tourist flow (Chen et al., 2015; Livieris et al., 2019) or cross-/upselling potential. Moreover, analogous forecasting tasks appear in production processes, sales analytics, meteorology, and so on. Whenever one does not forecast anything, but attempts to explain one variable by means of other dependent

---

A. Stöckl (✉) · U. Bodenhofer

School of Informatics, Communications and Media, University of Applied Sciences Upper Austria, Hagenberg, Austria

e-mail: [andreas.stoeckl@fh-hagenberg.at](mailto:andreas.stoeckl@fh-hagenberg.at); [ulrich.bodenhofer@fh-hagenberg.at](mailto:ulrich.bodenhofer@fh-hagenberg.at)

variables without any perspective into the future, then one usually speaks of *explanation (functions)* as opposed to forecasting. In the following, we will not explicitly distinguish between these two scenarios.

Formally speaking, a *regressor* (or *regression function*) is a mapping  $f: X \rightarrow \mathbb{R}$  from an object set  $X$  to the real numbers  $\mathbb{R}$ . In some practical tasks, it may be necessary to map objects to multiple values, i.e.,  $f: X \rightarrow \mathbb{R}^M$ . Since this is mainly equivalent to considering  $M$  regressors in parallel, we will stick to one-dimensional regression ( $M = 1$ ) in this chapter. In the following, we will not identify whether the values that we assign to the objects are in the future (as they typically are with *forecasting* values) or in the present. For the sake of simplicity, we will always merely speak of *prediction* when applying a regressor  $f: X \rightarrow \mathbb{R}$  to a new object.

Functions can be defined explicitly, for example, by designing formulas or rule sets. However, in many practical situations, it is difficult to do so; thus, it has become standard to design regression functions in a data-driven fashion via *machine learning*. In other words, the regression function is identified/trained by considering sample objects for which the correct value is known. This *data-driven design of regression functions* is the subject of the present chapter. As in the case of classification, we will search for this function using machine learning approaches by assuming that the “target value” is already known for a number of given objects. From this data set, we then want to construct a function that reconstructs the target value for this known data as accurately as possible and generalize it in the sense that a (hopefully) correct value is calculated for the feature combinations that are still unknown.

As regression tasks are fundamentally similar to those of classification, much of the conceptual introduction provided in the previous chapter (Bodenhofer & Stöckl, 2021) need not be repeated in full detail here. Instead, we will highlight what must be adapted for regression, while referring back to the previous chapter for those concepts and details that are shared between classification and regression. Artificial neural networks, support vector machines, and tree-based methods will, once again, not be described in full detail, but the ways in which regression variants differ from their corresponding classification variants will be presented.

In the following, we adhere to the same data representations as in the previous chapter (Bodenhofer & Stöckl, 2021), i.e., we have an input data matrix consisting of  $d$  features/columns and  $l$  objects/samples/rows. In order to learn a regression function from the data, the target values of these data objects need to be known. This means that one must know which value each sample object is mapped onto. Hence, our representation of the data objects is complemented by a target vector  $y$  that we can conveniently append to our data matrix as  $d + 1$ -st column:

$$\begin{array}{cccccc}
 x_{1,1} & x_{1,2} & \cdots & x_{1,d-1} & x_{1,d} & y_1 \\
 x_{2,1} & x_{2,2} & \cdots & x_{2,d-1} & x_{2,d} & y_2 \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 x_{l-1,1} & x_{l-1,2} & \cdots & x_{l-1,d-1} & x_{l-1,d} & y_{l-1} \\
 x_{l,1} & x_{l,2} & \cdots & x_{l,d-1} & x_{l,d} & y_l
 \end{array}$$

Given such a matrix, we can use a method that builds a regression function so that, for each row in this table, the value  $y_i$  is predicted as accurately as possible based on the  $d$  input features  $\mathbf{x}_i = (x_{i, 1}, \dots, x_{i, d})$ . Therefore, we try to find a regression function  $f: X \rightarrow \mathbb{R}$  to which  $f(\mathbf{x}_i)$  matches  $y_i$  as closely as possible for every instance. The standard way is to search for a regression function that minimizes the *mean squared error* (see below). Additionally, the objective may be augmented with regularization terms (Hastie et al., 2009; Vapnik, 1998).

## 1.2 Evaluation

Most of the concepts described for classification (Bodenhofer & Stöckl, 2021), particularly the concepts of generalization error and its estimation via the test/holdout method or cross validation, carry over to regression without any restrictions. The same holds true for hyperparameter optimization. The only concepts that need changing are the underlying evaluation measures. In classification, accuracy was calculated by the proportion of correctly assigned classes. Contrarily, in regression, each calculated value's deviation from the actual value is determined first, and then this error is averaged.

In the following, assume that we are considering a regression function  $f: X \rightarrow \mathbb{R}$  and that we want to evaluate its prediction performance on a data set similar to the one shown above. For any sample, the difference  $f(\mathbf{x}_i) - y_i$  is merely the deviation between the prediction  $f(\mathbf{x}_i)$  and the true value  $y_i$ . These differences are usually called *residuals*. As previously mentioned, the following measure is the most common one:

- **Mean Squared Error:**  $MSE = \frac{1}{T} \sum_{i=1}^T (f(\mathbf{x}_i) - y_i)^2$ .

The MSE is nothing other than the average of the squared residuals. It has an important advantage of depending on predictions in a differentiable way, which is essential in cases where the MSE is minimized on the training set using gradient-based methods (like artificial neural networks; see below). Secondly, the MSE penalizes large residuals while being more tolerant toward small residuals. The square root of the MSE is usually called the Root Mean Squared Error:  $RMSE = \sqrt{MSE}$ .

Some more measures that are also considered to be standard:

- **Mean Absolute Error:**  $MAE = \frac{1}{T} \sum_{i=1}^T |f(\mathbf{x}_i) - y_i|$
- **Mean Absolute Percentage Error:**  $MAPE = \frac{100}{T} \sum_{i=1}^T \frac{|f(\mathbf{x}_i) - y_i|}{y_i}$

The MAPE gives the average deviation of the predicted value  $f(\mathbf{x}_i)$  from the true value  $y_i$  as a percentage. Even though it is quite easily interpretable for non-experts, it comes with the downside of only being suitable for positive values, such as the numbers of sales of goods, etc. Moreover, due to dividing by the true value  $y_i$ , the measure is numerically unstable if the values  $y_i$  are close to zero and ill-defined if there is any value  $y_i$  landing on zero.

A more advanced, yet very helpful, measure is the so-called *coefficient of determination*  $R^2$  (Glantz & Slinker, 1990). Speaking in a simple manner, it indicates to which extent the errors of a regression model have a smaller variance than that of a simple baseline model that predicts the mean value of true values  $y_i$ :

$$R^2 = 1 - \frac{SS_f}{SS_b}$$

In this equation,  $SS_f = \sum_{i=1}^l (f(x_i) - y_i)^2$  is the sum of the squared residuals and  $SS_b = \sum_{i=1}^l (y_i - \bar{y})^2$  is the total sum of the squared differences of true values  $y_i$  from the mean of true values  $\bar{y}$ . This measure typically gives values between 0 and 1. A value of 1 would mean perfect predictions (since  $R^2 = 0$  can only hold true if all residuals are zero), and a value close to 0 indicates that the regression function predicts the values approximately to the same extent as if we had simply taken the mean of the true values. In the most extreme case, a negative  $R^2$  would mean that the regression function performs even worse than the mean value.

Lastly, another standard measure would be to simply compute the *sample (Pearson) correlation coefficient* between predictions  $f(x_i)$  and the true values  $y_i$ . However, this measure only determines whether predictions and true values are linearly correlated; it does not indicate the actual absolute deviation between predictions and true values (Abdi, 2007).

### 1.3 Regression Methods

In the subsequent section, we will highlight the most important machine learning methods for data-driven regression. This list is far from exhaustive, but it includes the most well-known options. As previously mentioned above, we will not describe the methods in full detail here, but, rather, highlight what needs to be adapted for regression, while also referring back to the previous chapter for a basic introduction of most methods. Furthermore, please note that pre-processing methods will not be described here as these can be employed precisely as described in the previous chapter (Bodenhofer & Stöckl, 2021). Finally, k-nearest neighbor methods and Naïve Bayes methods will be deliberately left out in this chapter since, even though variants of those methods exist for regression, they are not very common, and their practicality is limited.

#### 1.3.1 Linear Regression

*Linear regression*, which can even be traced back to Legendre and Gauss (Stigler, 1993), is the simplest form of performing a regression:

$$f(\mathbf{x}) = f(x_1, \dots, x_d) = \beta_0 + \sum_{i=1}^d \beta_i x_i$$

This model is a sum of inputs  $x_i$  weighted by coefficients  $\beta_i$ , while  $\beta_0$  is a constant (the so-called intercept). Given a training set, the coefficients and the intercept are chosen in such a way that they minimize the MSE on the training set (for this task, a global solution exists that can be found through a well-known simple method).

Like logistic regression for classification, linear regression can be augmented by adding a regularization term to aid feature selection and avoid overfitting. The most important representatives are ridge regression (also known as L2) (Hoerl, 1962), LASSO (least absolute shrinkage and selection operator, also known as L1; Tibshirani, 1996), and elastic net (Zou & Hastie, 2005).

### Advantages

- Computationally efficient.
- Simple and interpretable models.
- Built-in feature selection (in particular, the regularized variants); thus, insensitive to irrelevant/noisy features.

### Disadvantages

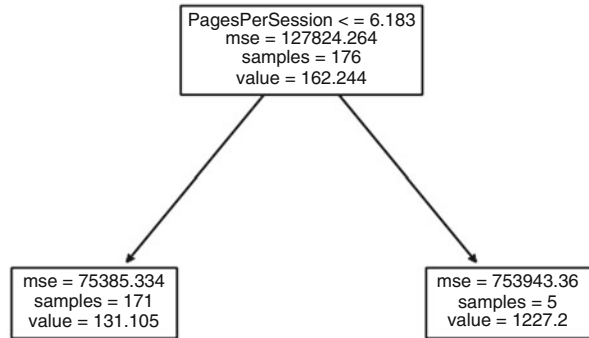
- The regularized variants are sensitive to input scaling; thus, normalization is advisable.
- Only reasonable in cases where a linear model is appropriate/called for.

## 1.3.2 Regression Trees

Like decision trees for classification, *regression trees* are rule-based classifiers that assign samples to a value by consecutively answering questions about input features. Comparatively, the only difference is that the predicted values associated with the leaf nodes are numerical values instead of categories. Consider the following example (Fig. 1):

The question displayed in the root node of this simple tree is whether the input feature “PagesPerSession” is smaller than or equal to 6.183. If a sample fulfills this condition, it is assigned to the left branch; otherwise, it is assigned to the right branch. Both branches lead to leaf nodes. The tree model assigns a value of 131.105 to samples that end up in the left leaf node and 1117.2 to samples in the right leaf node. While most decision tree induction concepts can be carried over to regression trees without any modification, the *splitting criterion* must be adapted for regression. The simplest splitting criterion (also applied in the example above) is based on the

**Fig. 1** Example of a simple regression tree



MSE. Thereafter, the split that leads to the highest overall improvement of the MSE in the resulting sub-branches is chosen (Breiman et al., 1984).

### Advantages

- Computationally efficient; scalable to large numbers of samples and features.
- Simple and easy.
- Models can be easily interpreted by humans (white-box models).
- Can handle categorical and numerical features.
- Built-in feature selection; thus, insensitive to irrelevant/noisy features.
- For numerical features, only the order of the values is relevant; thus, insensitive to input scaling; no normalization required.

### Disadvantages

- The resulting regression function is a piecewise constant step function; therefore, the regression function is discontinuous, and its prediction accuracy is quite limited.
- Difficult to address the underfitting-overfitting trade-off.

### 1.3.3 Regression Tree Ensembles

Both methods mentioned in the previous chapter for constructing ensembles can be used to build regression tree ensembles as well. As such, common variants of random forests and gradient tree boosting can be applied for conducting regression:

- The ensembling strategy for *random forests* can remain unchanged. The final prediction for a new sample is then computed simply by averaging the results of all the trees (unlike random forests for classification, which use majority voting (Breiman et al., 1984; Breiman, 2001)).

- Gradient boosting strategies must be adapted to a greater extent since the objective function is different for regression (Mason et al., 1999).

### Advantages

- Easy to apply; good results also without hyperparameter tuning.
- Built-in feature selection (in particular, the regularized variants); thus, insensitive to irrelevant/noisy features.
- For numerical features, only the order of the values is relevant; thus, insensitive to input scaling; no normalization required.
- Feature importances can be computed.
- Out-of-bag (OOB) estimates allow for estimating performance on future data (random forest only).

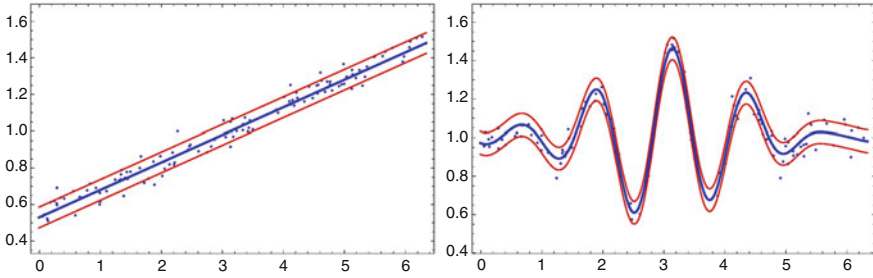
### Disadvantages

- Even though the resulting regression functions are much more fine-grained than single regression trees, the function is still a discontinuous, piecewise constant step function.
- Random forests are computationally expensive for large data sets.
- Boosted regression trees have a higher tendency to overfit than random forests do.

## 1.3.4 Support Vector Regression

While the idea of support vector machines, as laid out in the previous chapter, is genuinely geared toward classification, there is indeed a way of transferring the principle of support vector machines to regression. *Support vector regressors* are based on the notion that an  $\varepsilon$ -tube around the regression function is some kind of margin. Hence, the objective is to find a regression function such that the target values of as many training samples as possible lie within this tube. Given this objective, the resulting mathematical task and its solution are similar to a support vector machine for classification (Drucker et al., 1996; Schölkopf & Smola, 2002; Smola & Schölkopf, 2004). There are two variants; yet, we will only mention the more advanced one, the so-called  $\nu$ -SVR. This variant has two hyperparameters: a cost factor  $C$ , which controls the overall influence of a single training sample on the final regression function, and  $\nu$ , which controls the relative proportion of samples outside the  $\varepsilon$ -tube by adapting the size of the tube  $\varepsilon$  to the data automatically. Support vector regression can work with any kernel, as described in the previous chapter, and, therefore, we can train linear regressors using the support vector regression principle and non-linear functions using, for instance, the RBF kernel.





**Fig. 2** Two examples of one-dimensional regression tasks solved via support vector regression, a linear one on the left, and one using the RBF kernel on the right

Figure 2 presents two simple examples of support vector regressors for a one-dimensional regression. Both regression functions were trained using  $\nu$ -SVR with  $\nu = 0.2$  (i.e., at most, 20% of the samples are outside the  $\varepsilon$ -tube), and the tube width  $\varepsilon$  was automatically adapted to this ratio. The  $\varepsilon$ -tubes are also visualized in these graphs:  $\varepsilon$  corresponds to the distance between the regression function (blue lines) and any of the two bounds (red lines).

#### Advantages

- A low number of hyperparameters; thus, relatively easy hyperparameter optimization.
- Optimization algorithm gives a guaranteed global solution.
- Work well on small data sets.
- Good theoretical foundation.

#### Disadvantages

- Sensitive to input scaling; thus, normalization is advisable.
- Sensitive to irrelevant/noisy features; thus, feature selection is advisable.
- Choice of kernel depends on the application; may not be straightforward.
- Computationally expensive for larger data sets.

### 1.3.5 Artificial Neural Networks

*Artificial neural networks* can be used for regression with only minimal changes. This method is straightforward to use as there are as many neurons in the output layer as there are regression outputs. Since we have exclusively dealt with one-dimensional regression ( $M = 1$ ; see above) up to this point, this indicates that we only have one output neuron. If we wish to predict multiple outputs, then we would need to consequently increase the number of output neurons to equal the number of target

variables  $M$ . Other changes include (1) using the linear function  $\varphi(x) = x$  as the activation function in the output layer and (2) using the MSE as the loss function that is then minimized by gradient descent. Aside from that, the remaining network and the training procedure can remain the same (Goodfellow et al., 2016).

### Advantages

- Can exploit the subtlest patterns in the data, provided that the data sets are large enough.
- Works well for large data sets.
- Training and prediction can be sped up massively by the use of graphical processing units (GPUs).

### Disadvantages

- A large number of hyperparameters; thus, difficult hyperparameter optimization.
- Sensitive to input scaling; thus, normalization is advisable.
- Black-box models, i.e., it is very challenging to interpret how an ANN makes its predictions.
- Do not work well on small data sets.

## 2 Practical Demonstration

### 2.1 Use Case

Consider a hotel that focuses on sporty guests as the target group and offers various vacation packages and sports courses. Moreover, suppose that the hotel uses web tracking to record page views of the offers on its website and also records the bookings made. With the help of this data, a regression model should be developed that predicts the total revenue for each customer. This prediction should be based on the following input below, determined by the visitor's surfing behavior on the hotel's website:

- How often was the site visited?
- How many pages were viewed?
- In which categories were the offers that the visitors viewed?

Such a prediction would enable the hotel to display offers on its website in a personalized way, thus especially prompting visitors for whom the model predicts as high potential. As such, this tourism use case relates closely to *lead scoring*.

In the following, we will describe how to solve this use case in detail using the concepts and methods described earlier on in this chapter. This includes pre-processing the data and, subsequently, applying different regression methods

as well as evaluating and comparing them. We will present the most essential steps and results here, while the implementation details (all Python code plus results) can be found in the attached Jupyter notebook.

## 2.2 The Data

The data set used in the accompanying notebook is based on real data but has been anonymized for privacy reasons and slightly modified for demonstrative purposes. It is available as a CSV file and contains 138,760 lines. Each line represents one page view or booking. To read the data file and for basic manipulations of the tabular data, we will use the well-known “Pandas” package (McKinney, 2010) (Table 1).

After summarizing the recorded data of the user profiles and filtering the visitors with at least five visits within the considered time period, 252 records remained (for details on the generation of user profiles, see the previous chapter (Stöckl & Bodenhofer, 2021)). For each record, we calculated not only the number of visits and page views but also the number of offers booked as well as the revenue generated for each visitor. The latter will be the target that we wish to predict. In the table, this corresponds to the column “booking\_sum.” For a brief glimpse into the data set, see the first ten user profiles depicted in Table 2.

We will provide a simple one-dimensional regression task for illustrative purposes as well. Figure 3 presents a scatter plot of the feature “PagesPerSession” versus the booking sum target.

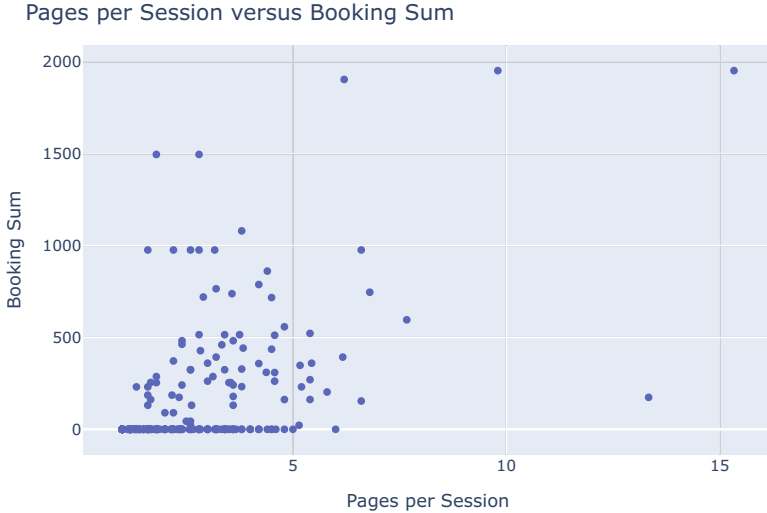
As one can note, there is no clear trend; at most, there seems to be a slight tendency toward more page views being an indication of higher sales, but more information is needed to build a useful model. Nevertheless, we start with a very simple model that tries to forecast the turnover from the numbers of pages per

**Table 1** First five rows/samples of the dataset

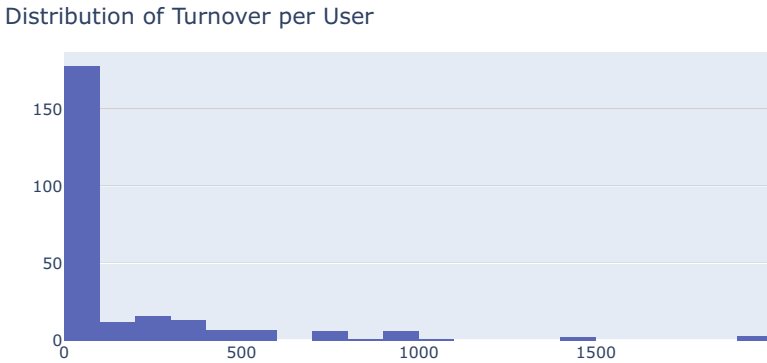
	type	page_id	category	price	user_id	user_session	daytime
0	View	1005135	Hiking	748	535871217	c6bd7419-2748-4c56-95b4-8cec9ff8b80d	2019-10-03 00:00:19
1	View	1004767	Cross-country skiing	255	512558158	9a206ba2-37c7-4354-9d31-37ff3bb297ed	2019-10-16 00:00:36
2	View	1005135	Run	748	535871217	c6bd7419-2748-4c56-95b4-8cec9ff8b80d	2019-10-03 00:00:43
3	View	1002544	Triathlon	464	532085144	77ae546a-542b-414c-a01b-c5ceca7e99cf	2019-10-06 00:00:44
4	View	1005105	Swimming	415	529755884	0b828fb6-99bd-4d26-beb3-3021f5d6102c	2019-10-18 00:00:50

**Table 2** First ten user profiles

user_id	Pageviews	Sessions	Items Booked	PagesPer Session	booking_sum	hiking	cross-country skiing	run	triathlon	swimming	cycling	others
497087183	24	6	0.0	4.000000	0.0	0.000000	0.083333	0.333333	0.083333	0.500000	0.000000	0.000000
499826904	6	5	0.0	1.200000	0.0	0.000000	0.166667	0.666667	0.000000	0.000000	0.166667	0.000000
512364387	10	6	1.0	1.666667	162.0	0.000000	0.000000	0.300000	0.100000	0.400000	0.100000	0.100000
512397324	21	5	0.0	4.200000	0.0	0.095238	0.95238	0.238095	0.047619	0.380952	0.047619	0.095238
512407916	33	5	1.0	0.600000	154.0	0.000000	0.090909	0.151515	0.151515	0.363636	0.090909	0.151515
512424366	25	7	0.0	3.571429	0.0	0.040000	0.160000	0.520000	0.040000	0.200000	0.000000	0.040000
512426470	12	6	0.0	2.000000	0.0	0.83333	0.000000	0.500000	0.083333	0.333333	0.000000	0.000000
512486232	11	5	1.0	2.200000	90.0	0.181818	0.181818	0.181818	0.181818	0.090909	0.090909	0.090909
512515918	11	6	0.0	1.833333	0.0	0.000000	0.181818	0.272727	0.000000	0.363636	0.181818	0.000000
512558829	15	5	0.0	3.000000	0.0	0.133333	0.000000	0.466667	0.133333	0.266667	0.000000	0.000000



**Fig. 3** Pages per session versus booking sum



**Fig. 4** Distribution of turnover per user (column “booking\_sum”)

session. One more fact that can be observed from the scatter plot above is that the booking sums seem unevenly distributed, with a strong emphasis placed on the lower end (i.e., many zeros). The following histogram also confirms this impression (Fig. 4).

### 2.3 Splitting Training and Test Data

Before we start with the calculations, we have to think about how we judge a model’s quality. For this purpose, we need to divide our user profile data set into two parts,

one for model generation (“training data set”—70%) and one for assessing the quality of the model (“test data set”—30%).

### 2.4 Prediction of Visitors’ Turnover: Model Building

For the training and evaluation of the regression models, we use the machine learning framework scikit-learn (<https://scikit-learn.org/>; Pedregosa et al., 2011).

We first have to split the data set into training and test data (holdout method, see above) in which we use 70% of the data for training and the rest for evaluation. In this example, these comprise 176 and 76 data samples, respectively. Then, we can use the same models as described in section “Regression Methods”:

- Linear regression
- Regression trees
- Random forest
- Gradient tree boosting
- Support vector machine regression
- Artificial neural networks

For each of these methods, we implemented regression based on one feature and added a second regression using all features in the accompanying Jupyter notebook. This aids in developing an intuition for the procedures and assessing risks, such as overfitting, better.

Let us first have a look at a simple linear regression model with only one variable. Figure 5 shows a scatter plot of the training and test data along with the linear regression function.

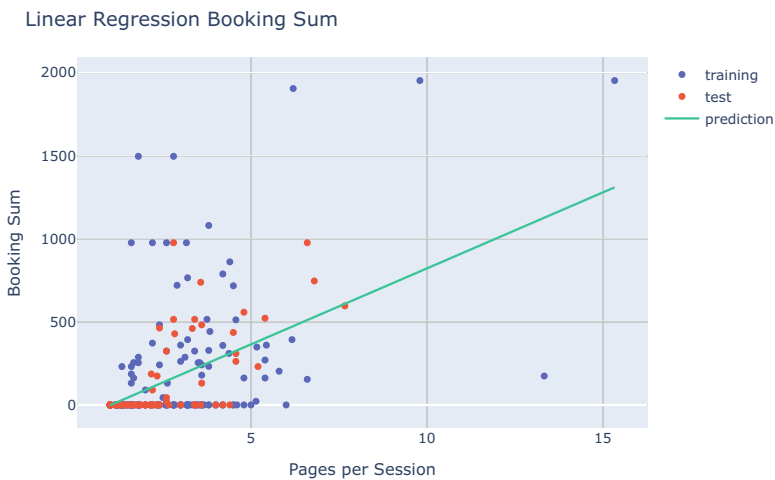
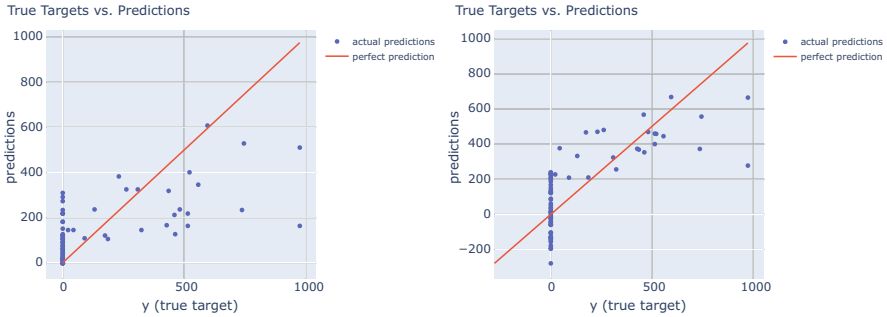


Fig. 5 Linear regressor along with training and test data



**Fig. 6** Scatter plot that contrasts prediction and true values. A perfect prediction would lie on the orange-colored diagonal line. Left panel: Linear regression with one input feature. Right panel: Linear regression with nine input features

In order to be able to judge the quality of the model not only in a visual sense, we will now apply measurements for the regression performance. All results in the notebook include the results for MSE, RMSE, MAE, and the coefficient of determination  $R^2$ . Moreover, for each method, we also provide a scatter plot that contrasts the predictions with the true values in addition to a diagonal line that corresponds to a perfect prediction. This scatter plot would even work for regression with many input features and allows for an additional visual investigation of the regression performance. Figure 6 depicts an example of such a scatter plot.

For linear regression, we obtained an RMSE of 184.4, an MAE of 122.9, and an  $R^2$  score of 0.43 when using only one feature. When taking all nine features, we obtained an RMSE of 169.6, an MAE of 130.5, and an  $R^2$  score of 0.52. The latter explains about 52% of the variance, which is a good start, yet also promises some room for improvement.

Solely for demonstration purposes, we also tried to train a linear regression model using two input features, shown below in the 3D scatter plot (Fig. 7).

We will now move on to other methods, starting with regression trees. As already observed from the scatter plot and the regression function in the one-dimensional case, it seems that the simple regression tree overfits the training data (see Fig. 8).

Unfortunately, for this use case, all results obtained from the regression trees were quite disappointing in the end. Despite trying various options, the results were poor; the best  $R^2$  score for the full model with all features was 0.13, which is unacceptably low.

Using random forest resulted in an unsatisfactory  $R^2$  score of 0.22 for the one-dimensional task. When using all nine features, however, the results were much better ( $R^2$  score of 0.51, while the RMSE was 172.2 and the MAE was 94.6 for the test set). We also performed a hyperparameter optimization for random forests using random search (in order to demonstrate this technique as well). The results were good, but they are unable to outperform the model with the default settings.

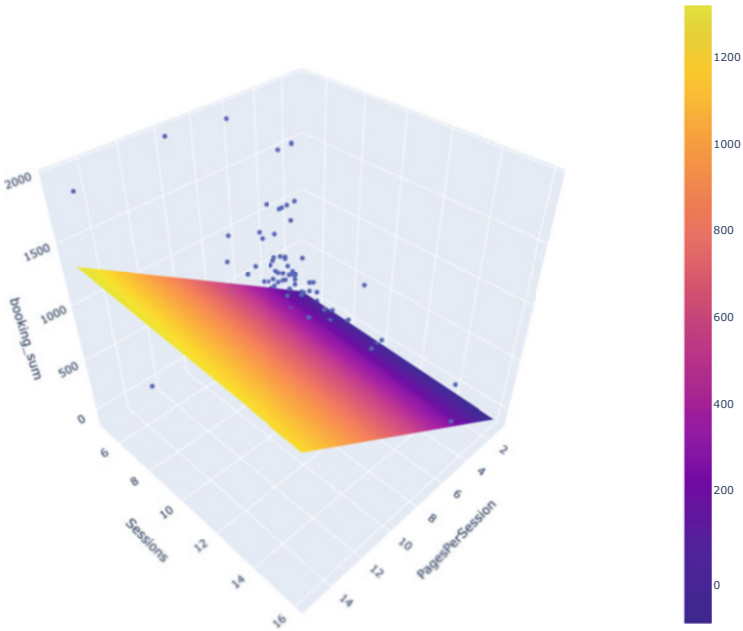


Fig. 7 3D scatter plot of linear regression with two features along with regression function

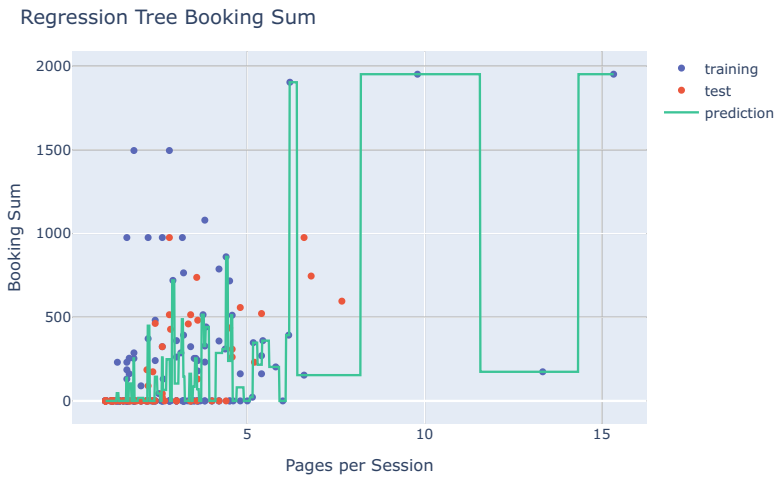


Fig. 8 Tree regressor along with training and test data

For gradient tree boosting (using the implementation of the XGBoost package; Chen & Guestrin, 2016), the results were even worse (with nine features, an  $R^2$  score of 0.22, an RMSE of 216.4, and an MAE of 115.7 were obtained based on the test set) (Fig. 9).





**Fig. 9** Scatter plot that contrasts prediction and true values for random forest (left) and gradient tree boosting (right); both models were trained with all nine features



**Fig. 10** One-dimensional regression functions obtained from support vector regression and different kernels

Next, we will try a support vector regression, first with the linear kernel and then also with non-linear kernels. In order to get an overall impression, Fig. 10 illustrates the support vector regression functions for different kernels: linear kernel, polynomial kernel with degree 4, and RBF kernel with different values of  $\gamma = \frac{1}{2\sigma^2}$ , namely 0.01, 1, and 10. All models were constructed using the  $\nu$ -SVR method with  $C = 1000$  and  $\nu = 0.2$ .

Support vector regression with all nine features and the RBF kernel plus an ad-hoc choice of  $C = 1000$ ,  $\gamma = 0.1$ , and  $\nu = 0.5$  led to the best overall results. We obtained an RMSE of 166.6, an MAE of 92.4, and an  $R^2$  score of 0.54, meaning 54% of the variance can be explained by the regression model. Although this is still not necessarily top-notch, it was the best we could achieve on this data set. We tried

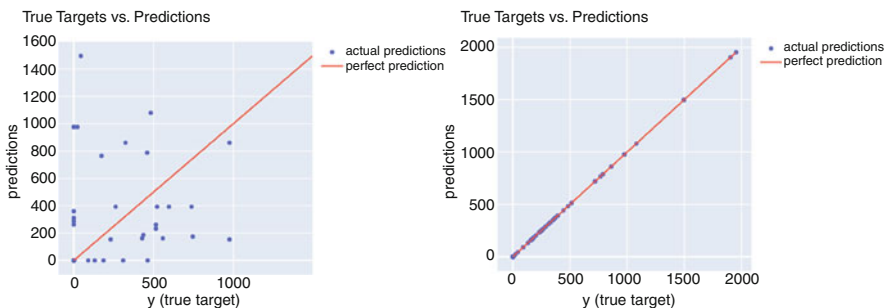
to improve this by applying a thorough hyperparameter optimization via a grid search, but it did not result in any improvements.

Naturally, note that we did indeed perform normalization of the input features when using support vector regression. This was imperative because support vector machines, whether linear or non-linear, are always sensitive to the scaling of features. Scikit-learn, in this sense, eases the use of such a pre-processing technique greatly; thanks to its pipelining mechanism, it permits the stacking of multiple steps in a pipeline, which can then be trained at once on the training sets and be employed on the test sets in an effortless manner.

This normalization was also necessary for the last method, artificial neural networks. We want to emphasize that the use of artificial neural networks in the accompanying Jupyter notebook is only for demonstration purposes. Artificial neural networks are currently the most powerful methods in machine learning and are mainly used for large data sets. The data set used for this use case is quite small; therefore, neural networks do not appear to be a very promising approach. Since the purpose of this example is merely to act as a teaser, we used the implementation from scikit-learn here as well (Pedregosa et al., 2011), even though more advanced implementations for neural networks are available in packages such as Tensorflow/Keras (<https://www.tensorflow.org/>; Abadi et al., 2015; Chollet, 2018) or PyTorch (<https://pytorch.org/>; Paszke et al., 2019).

Regardless of which neural network architecture we tried, we continuously ran into overfitting issues. This is best illustrated by the two scatter plots in Fig. 11; the left panel shows the results obtained on the test set, and the right panel shows the results for the training set. While the predictions are almost random on the test set (left), the fit is perfect for the training set (right). This is clearly a case of massive overfitting.

The results were very poor (negative  $R^2$  score and also very high RMSE and MAE values). Thus, we can conclude that artificial neural networks are simply not suitable for this regression task with such few training samples.



**Fig. 11** Scatter plot that contrasts true values and the predictions made with an artificial neural network

### 2.4.1 Summary of Results

All details and results can be found in the accompanying Jupyter notebook, which also contains the full source code and results, but also further references relating to this chapter and comments about implementation details. The general results can be summarized as follows:

Using one feature only allows us to visualize the regression function nicely, but it does not give very good results.

Most methods, at least the ones that give meaningful results, perform better on the full data set with all features.

Some methods fail, while others deliver good results. For this case study, the best results were obtained via random forest regression and traditional linear regression.

## 2.5 *Application of the Model*

After having trained a successful model, we can then put it into practice. This can be done by recording user interactions on the website, again with the tracking solution, and using the model to forecast the turnover potential in real-time during the user's visit.

The anonymous users are identified through a unique ID based on their digital fingerprint. If a visitor browses the website, the system first needs to check whether he/she has already visited the page at least five times. This was a prerequisite for the training of our model. Therefore, we may only use the model if the prerequisite is given and use it to forecast the user's potential turnover.

If a user meets the requirements, we can calculate a sales forecast based on his/her behavioral data. On this basis, marketing actions can be implemented on the site to exploit this forecast, for example, by displaying suitable action banners, by offering a voucher to users who register for the newsletter, and so on.

### **Service Section**

**Main Application Fields:** Any field in which numerical values should be assigned to objects. In tourism, typical use cases include, but are not limited to, the following:

- Forecasting sales, bookings, occupancy rates (globally)
- Forecasting turnover on an individual level (per customer; see use case above)
- Estimating cross-/upselling potential

(continued)

**Limitations and Pitfalls:** The methods presented in this chapter are used exclusively to identify regression models for which sufficient data are available. This data must consist of objects that are characterized by a sufficiently large and meaningful set of input features and their corresponding target values. Whether the amount of data and the set of features are sufficient for a decent regression performance cannot be answered a priori but, rather, needs to be evaluated empirically by actually training the regression models and evaluating the results.

The following pitfalls should be avoided (see also above):

- When pre-processing the data, biases in test sets must be avoided.
- In order to avoid under- or overfitting, a thorough hyperparameter optimization should be carried out for some of the methods (in particular, support vector machines and artificial neural networks).
- It is imperative to perform hyperparameter optimization using a validation set or by using cross validation on the training set. The test set should only be consulted when the model is final; otherwise, the selection of the hyperparameters is biased to the test set, which renders the test set useless.

**Similar Methods and Methods to Combine with:** Classification (see previous chapter) is a similar approach, with the only difference being that a category, rather than a numerical value, is predicted. In some applications, the user has a free choice of whether to use classification or regression. In our hotel example, for instance, we can predict the actual amount of bookings or the total price of booked items (→ regression), but we could also decide in favor of predicting whether a customer books/does not book (→ classification).

Regression methods can, or must, be combined with other methods, such as:

- Unsupervised methods for pre-processing data (e.g., dimensionality reduction)
- Methods for extracting features from unstructured data (e.g., images, text)

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter11-Regression>

## Further Readings and Other Sources

Linear regression: <https://medium.com/analytics-vidhya/linear-regression-in-machine-learning-eeee4dbc8bae>

Regression trees: <https://medium.com/@sonish.sivarajkumar/classification-and-regression-trees-f3e58be39f86>

Random forest: [https://medium.com/@harshdeepsingh\\_35448/understanding-random-forests-aa0cceedbbbb](https://medium.com/@harshdeepsingh_35448/understanding-random-forests-aa0cceedbbbb)

Gradient tree boosting: <https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d>

Support vector regression: <https://medium.com/analytics-vidhya/machine-learning-support-vector-regression-181aea35bedf>

Artificial neural networks: <https://medium.com/@purnasaigudikandula/a-beginner-intro-to-neural-networks-543267bda3c8>

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems.
- Abdi, H. (2007). Coefficients of correlation, alienation and determination. In N. J. Salkind (Ed.), *Encyclopedia of measurement and statistics*. Sage.
- Bodenhofer, U., & Stöckl, A. (2021). Classification. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications*. Springer.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (Eds.). (1984). *Classification and regression trees*. CRC Press.
- Chen, K.-Y., & Wang, C.-H. (2007). Support vector regression with genetic algorithms in forecasting tourism demand. *Tourism Management*, 28(1), 215–226. <https://doi.org/10.1016/j.tourman.2005.12.018>
- Chen, R., Liang, C.-Y., Hong, W.-C., & Gu, D.-X. (2015). Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. *Applied Soft Computing*, 26, 435–443. <https://doi.org/10.1016/j.asoc.2014.10.022>
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, & R. Rastogi (Eds.), *Proceedings of the 22nd ACM SIGKDD International Conference on knowledge discovery and data mining* (pp. 785–794). ACM. <https://doi.org/10.1145/2939672.2939785>
- Chollet, F. (2018). *Deep learning with python. Safari tech books online*. Manning Publications.
- Claveria, O., Monte, E., & Torra, S. (2015). Combination forecasts of tourism demand with machine learning models. *Applied Economics Letters*, 23(6), 1–4. <https://doi.org/10.1080/13504851.2015.1078441>
- Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., & Vapnik, V. N. (1996). Support vector regression machines. *Advances in Neural Information Processing Systems*, 9, 151–161.
- Glantz, S. A., & Slinker, B. K. (1990). *Primer of applied regression and analysis of variance*. McGraw-Hill.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction (second)*. Springer series in statistics. Springer.
- Hoerl, A. E. (1962). Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58, 54–59.
- Law, R., Li, G., Fong, D. K. C., & Han, X. (2019). Tourism demand forecasting: A deep learning approach. *Annals of Tourism Research*, 75, 410–423. <https://doi.org/10.1016/j.annals.2019.01.014>
- Livieris, I. E., Pintelas, E., Kotsilieris, T., Stavroyiannis, S., & Pintelas, P. (2019). Weight-constrained neural networks in forecasting tourist volumes: A case study. *Electronics*, 8(9), 1005. <https://doi.org/10.3390/electronics8091005>

- Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). Boosting algorithms as gradient descent. *Advances in Neural Information Processing Systems*, 12.
- McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the Python in Science Conference, Proceedings 9th Python in Science Conference* (pp. 56–61). SciPy. <https://doi.org/10.25080/Majora-92bf1922-00a>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., . . . Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85), 2825–2830.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels. Adaptive computation and machine learning*. MIT Press.
- Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199–222. <https://doi.org/10.1023/B:STCO.0000035301.49549.88>
- Stigler, S. M. (1993). *The history of statistics: The measurement of uncertainty before 1900*. Belknap Press.
- Stöckl, A., & Bodenhofer, U. (2021). Regression. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications*. Springer.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1), 267–288.
- Vapnik, V. N. (1998). *Statistical learning theory. Adaptive and learning systems*. Wiley Interscience.
- Xie, G., Qian, Y., & Wang, S. (2021). Forecasting Chinese cruise tourism demand with big data: An optimized machine learning approach. *Tourism Management*, 82, 104208. <https://doi.org/10.1016/j.tourman.2020.104208>
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2), 301–320.

# Hyperparameter Tuning



## The Art of Fine-Tuning Machine and Deep Learning Models to Improve Metric Results

Pier Paolo Ippolito

### Learning Objectives

- Explain the difference between model parameters and hyperparameters
- Motivate the importance of hyperparameter tuning
- Introduce and compare different possible methods for hyperparameter tuning
- Demonstrate how these different techniques can be applied in a tourism research case
- Provide examples of literature/studies in which hyperparameter tuning has been used in tourism

## 1 Introduction and Theoretical Foundations

### 1.1 Motivations

Hyperparameter tuning is a crucial step in the Data Science workflow (Fig. 1). If applied correctly, hyperparameter tuning can, in fact, turn an ineffective model into a model ready to be deployed to production and make real-world decisions. As machine learning models' performance can be highly dependent on initial conditions and potential applied constraints, different techniques will be presented in this chapter to attempt to limit this type of problem to the greatest possible extent.

Traditional machine learning models are characterized by two different types of parameters (Fig. 2):

---

P. P. Ippolito (✉)  
SAS Institute, London, UK

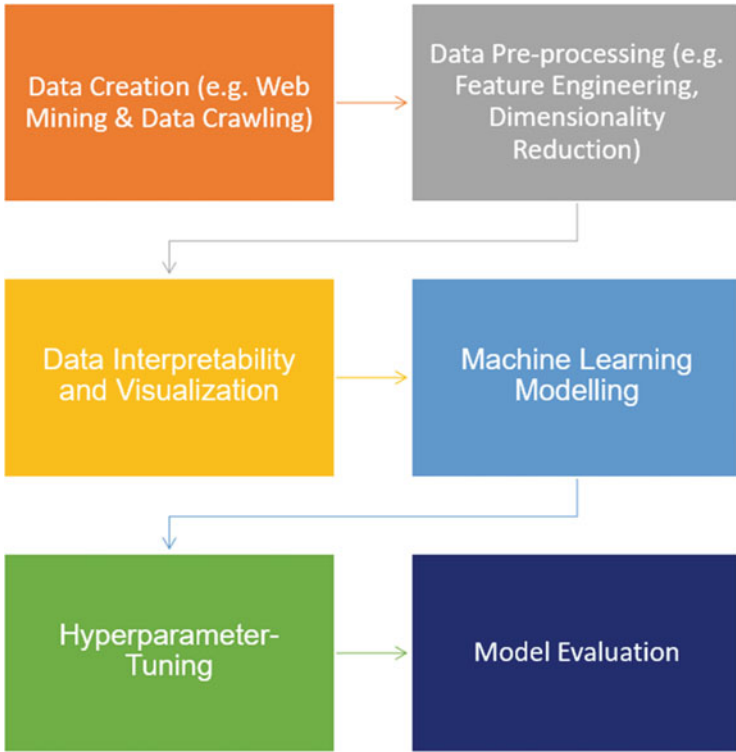


Fig. 1 Data Science workflow. Source: Author’s own illustration

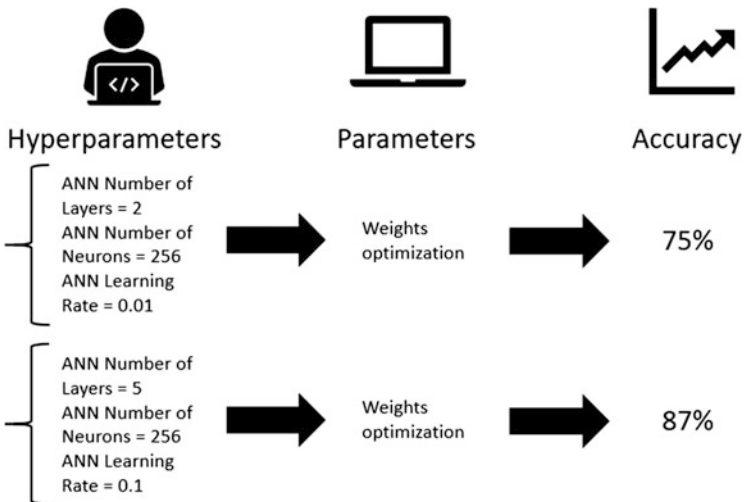
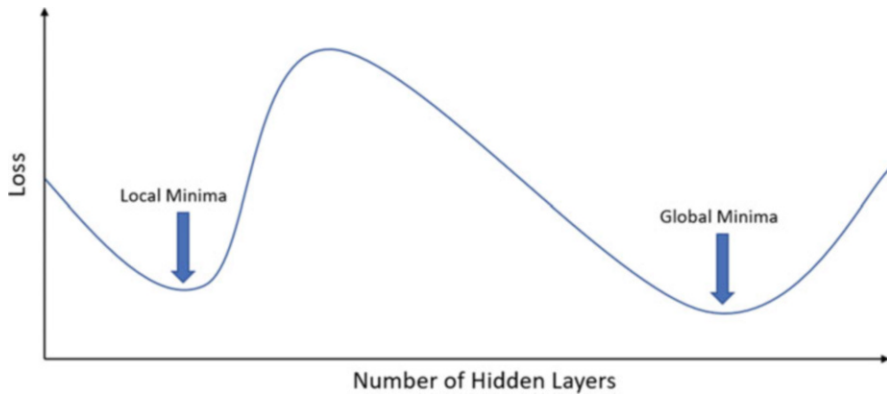


Fig. 2 Hyperparameters optimization workflow. Source: Author’s own illustration



**Table 1** Parameters vs. Hyperparameters

Parameter	Hyperparameter
Learnt during model training	Values defined before training
Internal to the model and saved after training	External to the model and not saved after training
Dependent on the training dataset	Independent from the training dataset

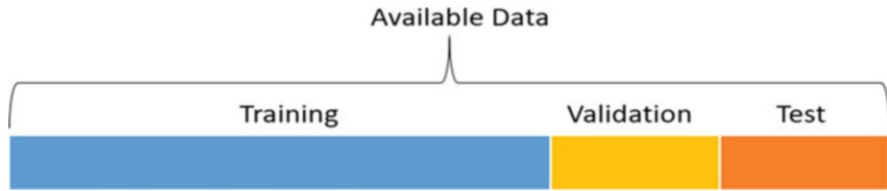


**Fig. 3** Example of a function search space – number of hidden layers vs. overall loss. Source: Author’s own illustration

- **Model Parameters:** Model parameters are the different variables learnt while training a model. These types of parameters can include, for example the weight values in Artificial Neural Networks (ANNs) and linear regression.
- **Hyperparameters:** Hyperparameters are all the different parameter values, which can be arbitrarily defined by a user before training a model. Therefore, these parameters can form various constraints on how a model should train. For example two hyperparameters in a Random Forest model are the number of estimators in total and the maximum allowed depth in each estimator. In a Support Vector Machine (SVM) algorithm, on the other hand, some examples of hyperparameters would be the type of kernel to use (e.g. linear, Gaussian, etc.) or the strength of the penalty parameter of the error term. Overall, having a deep understanding of how the different hyperparameters work is of the utmost importance when it comes to deciding their values.

In general, hyperparameters are used to refine the structure of a model, while model parameters determine how the input data is ingested and manipulated by the model to get the desired output (Table 1). Every machine learning model has hyperparameters, and finding a way to automatically determine the best values is one of the first steps towards Automated Machine Learning (AutoML).

Hyperparameter tuning can be considered an optimization problem. Each machine learning model is provided with a set of hyperparameters, and the objective is to find the best available combination of values to either maximize its accuracy or minimize its loss (moving along the search space of a function). Figure 3 shows a simple example clarifying this concept.



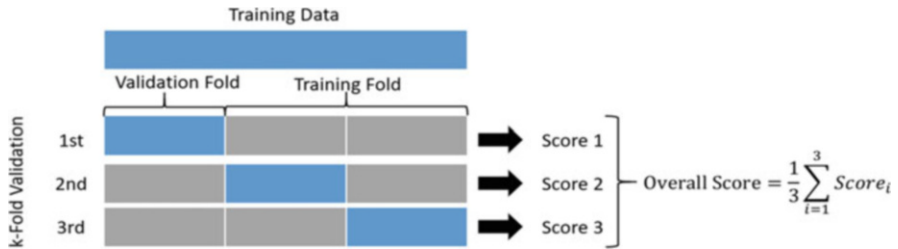
**Fig. 4** Dataset Partitions. Source: Author’s own illustration

Choosing the number of hidden layers is a common hyperparameter in ANNs, and finding the most appropriate value for this variable can lead to a reduction in the overall loss (see Neural and Mehlig (2019) for a thorough explanation regarding ANNs). One of the main problems when trying to solve optimization tasks is the possible presence of local minima in the search space. If a local minimum is reached, the optimization algorithm may get tricked into believing that the best possible solution has already been determined, when, in fact, there might be better solutions available further along in the search space (global minima). These same concepts hold true (but in higher dimensions) when taking multiple hyperparameters into consideration at the same time.

Hyperparameter tuning has been a prominent field of research since the early 1990s, and various studies have established that different hyperparameter values consistently perform best for different datasets (Kohavi & John, 1995). Therefore, the process of identifying the best possible combination of hyperparameters can be of critical importance when trying to improve a model’s overall performance (Melis et al., n.d.; Snoek et al., n.d.). However, hyperparameter tuning needs to be applied carefully in order to avoid false expectations (e.g. good performance during training but not the case when confronted with brand new data).

Machine learning models are generally trained on a limited amount of data; thus, this data can be viewed as a small sample of a much larger population. The final objective would then be to use this sample to make accurate predictions on all the unseen data. In order to achieve this, the available data is typically divided into training, validation, and test sets. By dividing the dataset into different partitions, it makes it possible to rigorously test whether the model actually performs better. Improvements in performance should be recorded for all three partitions to ensure that improved performance will also be registered once the model is incorporated in commercial and research applications (Fig. 4).

One of the main problems of modern machine learning is overfitting, a phenomenon which takes place when a model tries to learn too much from the training data (i.e. memorizing not just the main characteristics but also any added form of noise not representing the actual population). Overfitting can, therefore, lead to trained models failing to generalize the unseen data (e.g. good performance on the training data but poor performance on the test data). In this case, dividing a dataset into different partitions can aid in reducing the overall risk of overfitting and identify any form thereof.



**Fig. 5** k-Fold Cross-Validation. Source: Author’s own illustration

Additionally, to further test if a model is prone to overfitting, it is possible to apply a form of the Cross-Validation method in which k-Fold is the most commonly used form (Fig. 5). In k-Fold validation, the following steps are taken:

1. Divide the training set into N partitions.
2. Train the model on N-1 partitions and test against the left-over partition.
3. Repeat this previous step iteratively, and at each iteration, change the left-over partition.
4. Once the process has been repeated N times, the results obtained from each iteration are subsequently averaged to calculate the final score.

Two of the main advantages of using Cross-Validations are, first, that overfitting the training data is generally prevented, and, second, the best combination of hyperparameters is selected to provide optimal long-lasting results (once the model has been deployed).

## 1.2 Techniques

Hyperparameter tuning techniques are mainly divided into two different categories: (1) traditional approaches focusing solely on exploring different areas of the search space and (2) more complicated approaches hoping to speed up the process by focusing predominantly on the most promising combinations of hyperparameter values.

As part of this chapter, the following different approaches to hyperparameter tuning will be introduced:

1. Manual Search
2. Grid Search
3. Random Search
4. Bayesian Optimization
5. Genetic Algorithms

These approaches have also been extensively considered and compared against in publications such as Yang and Shami (2020) and Luo (2016).

### 1.2.1 Manual Search

The easiest approach to hyperparameter tuning is to try different values for a hyperparameter and see how they affect the results (e.g. variations in loss/accuracy). By repeating this process a few times, the original “guess” can be improved through trial and error. In most cases, the original guess will be dictated by the user’s judgement or experience. For example in some situations, it is perfectly reasonable to assume that increasing the batch size or base learning rate might help create more accurate models in a quicker manner.

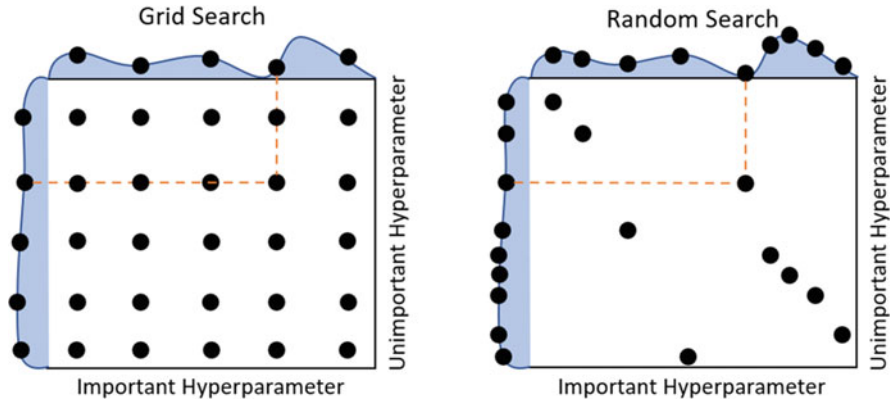
### 1.2.2 Grid Search

In the Grid Search approach, a grid of values is set up for each hyperparameter, and the model is then trained/tested against each possible combination of values. Grid Search is sometimes also referred to as “*full factorial design*” as it evaluates the Cartesian product for each set of values assigned to each hyperparameter (Montgomery, 2020). Therefore, this method allows for many possible combinations to be covered and a good portion of the available optimization search space to be investigated. One downside to this approach, however, is that it can be extremely expensive compared to other approaches (both in terms of time and the computing resources needed). As a result of the curse of dimensionality, this aspect can be fundamentally important when working with many different hyperparameters in a continuous, rather than discrete, space. For instance, when optimizing an Artificial Neural Network, trying seven different values for the number of hidden layers, three different learning rates, and two different activation functions would result in training 42 different models ( $7 \times 3 \times 2$ ) using Grid Search.

### 1.2.3 Random Search

In hopes of trying to overcome the limitations of Grid Search, Random Search was developed in which different possible hyperparameter combinations are randomly sampled from the search space. This approach has been demonstrated to be more efficient than traditional manual and grid search approaches when some hyperparameters have more weight than others (Bergstra et al., 2012; Hutter et al., n.d.). Hence, Random Search, when compared to Grid Search, can allow similar results to be achieved whilst considerably reducing the associated computational requirements. Nevertheless, one possible disadvantage of using Random Search is the fact that there is no control as to how the different values are selected, therefore making it more difficult to understand why some parameters might work better than others.

Figure 6 shows a simple two-dimensional example illustrating how Grid and Random Search work. In this case, two different hyperparameters are used. The first one seems to provide an almost flat surface for the search space; thus, varying this



**Fig. 6** Grid Search vs. Random Search 2D representation. Source: Author’s illustration adapted from Feurer and Hutter (2019)

hyperparameter does not seem to cause any major change in the overall loss (unimportant hyperparameter). On the other hand, the second hyperparameter has a much more complicated search space with local minima and maxima (important hyperparameter). Therefore, in order to significantly reduce the overall loss, it would be more profitable to focus on better optimizing the second hyperparameter rather than the first. Using Grid Search would allow the two search spaces to be covered evenly, yet using Random Search would help achieve similar loss results while consistently reducing the extent of the search.

#### 1.2.4 Bayesian Optimization

As described in Shahriari et al. (2016), Bayesian Optimization is a hyperparameter tuning technique created to take “*the human out of the loop*”. In fact, by using this approach, all the different hyperparameter values can be determined automatically without any human intervention. This technique has been used extensively in the past few years in order to achieve state-of-the-art performances in areas such as Image and Speech Recognition (Snoek et al., n.d.; Snoek et al., 2015; Dahl et al., n.d.). Furthermore, several studies have demonstrated how Bayesian Optimization can reliably provide better results than Random Search in many applications (Bergstra et al., 2013; Bergstra et al., n.d.).

Bayesian Optimization is part of a class of algorithms commonly referred to as Sequential Model-Based Optimization (Bossek et al., 2020). The key difference between this type of technique and Grid/Random Search is that information from past experiments are used to understand which values for the hyperparameters might be best to try and/or avoid (allowing to focus only on the most promising combinations). In this way, Bayesian Optimization, compared to Grid/Random Search, tends to converge faster to an optimal/sub-optimal solution and, in turn, pays less attention to areas of the search space that do not provide any added value.

This process is carried out by using the following steps:

1. Create a surrogate probabilistic model of the objective function by mapping hyperparameter values to a probability score with respect to the function being optimized. This surrogate model can be considered an approximation of the actual objective function. Apart from Gaussian Processes (Sui et al., n.d.), other algorithms, including Tree Parzen Estimator (Eggenesperger et al., n.d.) and Random Forest Regression, can be used.
2. Test hyperparameter values against the surrogate model and then test the most promising ones against the objective function. By using the surrogate model, testing hyperparameters against the objective function can be avoided (which is much more computationally expensive to do).
3. Based on the results obtained, update the surrogate model adequately.
4. Repeat steps 2 and 3 iteratively until the desired results are obtained or the time/number of iterations allowed runs out.

The surrogate model can be repeatedly updated once new observations are gathered, making use of Bayes Rule (“Bayesian Hyperparameter Optimization - A Primer on Weights & Biases”, n.d.):

$$P(\text{metric}|\text{hyperparameters set}) = \frac{P(\text{hyperparameters set}|\text{metric})P(\text{metric})}{P(\text{hyperparameters set})}$$

Here,  $P(\text{hyperparameters set})$  is the probability to reach this set of hyperparameters, and  $P(\text{metric})$  is the value of the metric in scalar.  $P(\text{metric}|\text{hyperparameters set})$  provides the probability that a combination of hyperparameters will minimize/maximize the chosen metric, while  $P(\text{hyperparameters set}|\text{metric})$  is the probability of the hyperparameters set if the metric is minimized/maximized.

Making use of the search space introduced in Figs. 3 and 7 illustrates how Bayesian Optimization works. In this example, the surrogate model is able to

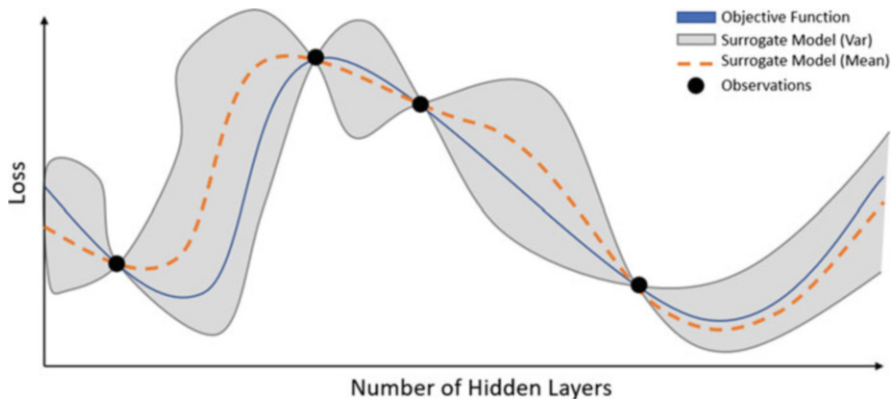


Fig. 7 Bayesian Optimization. Source: Author’s own illustration

approximate the objective function after four completed iterations. Gathering more observations thereafter would allow the approximation to become even more precise. Therefore, selecting hyperparameter values that can perform well on the surrogate model will most likely lead to good results on the objective function as well. However, it is important to note that Bayesian Optimization is considered one of the most difficult methods to implement and debug.

### 1.2.5 Genetic Algorithms

Lastly, Genetic Algorithms are a class of hyperparameter tuning techniques inspired by the Darwinian process of Natural Selection. For evolution to take place, four considerations need to be met:

- **Reproduction:** To perpetuate their species, organisms need to be able to generate offspring.
- **Heredity:** Generated offspring need to be able to inherit different characteristics from their parents.
- **Variation:** A population should be formed by individuals with unique characteristics (i.e. there needs to be some sort of variety between individuals).
- **Change in Fitness:** The reproductive success of individuals (fitness) is dependent on the differences between them and the other members of the population.

An evolutionary process via the Genetic Algorithms method can thus be exemplified in the following steps:

1. A population of  $N$  machine learning models is instantiated in which each of them has different unique values for their hyperparameters.
2. The performance of each model is calculated, and solely the top  $N/2$  performing models (the most promising) are kept.
3. In order to try to improve the population as a whole, other  $N/2$  models are created by implementing similar (but different) hyperparameters from the top performing models. In this way, a next generation of  $N$  models can be formed again.
4. Repeating this process would then allow the overall performance of the population of models to gradually increase iteration by iteration.

However, using Genetic Algorithms' different hyperparameters, such as the original population size, the proportion of the offspring that must change their hyperparameters each iteration, etc., still remains to be specified.

## 1.3 Summary

In order to briefly recapitulate the different approaches introduced thus far, a summary outlining the different advantages/disadvantages and available Python libraries is presented in Table 2.

**Table 2** Comparison of hyperparameter tuning techniques

Hyperparameter tuning technique	Advantages	Disadvantages	Python Libraries
Manual search	Simple, fast	Low overall performance	No additional libraries needed
Grid search	Large part of the search space can be explored	Computationally expensive	Scikit-learn model selection
Random search	More efficient than grid search	No control over the chosen values	Scikit-learn model selection
Bayesian optimization	Search gradually focuses on the most promising combinations	Most difficult technique to implement and debug	Hyperopt, Optuna, Scikit-optimize
Genetic algorithms	Inspired by the same principle of natural selection	Requires additional hyperparameters in order to set up the algorithm	TPOT (tree-based pipeline optimization tool)

## 2 Practical Demonstration

Being able to predict possible flight delays/cancellations can be of great help in improving customer satisfaction and overall travelling experiences. In order to try to solve this type of task, this section will take the “Flight Delays and Cancellations” Kaggle dataset (Department of Transportation, 2015) to create a model that is able to predict flight delays in the USA.

The “Flight Delays and Cancellations” Kaggle dataset is composed of three different datasets (`airlines.csv`, `airports.csv`, and `flights.csv`) containing respectively information about the registered airlines, airport specifics, and each flight statistic. These three datasets recorded by the U.S. Department of Transportation’s (DOT) Bureau of Transportation Statistics in 2015 contain all the information about major airlines’ domestic flights.

This demonstration will be carried out in Python using open source libraries such as Pandas, Numpy, Matplotlib, Scikit-learn, etc. In this section, only the key code components will be presented in order to explain the overall workflow and demonstrate the analysis results.

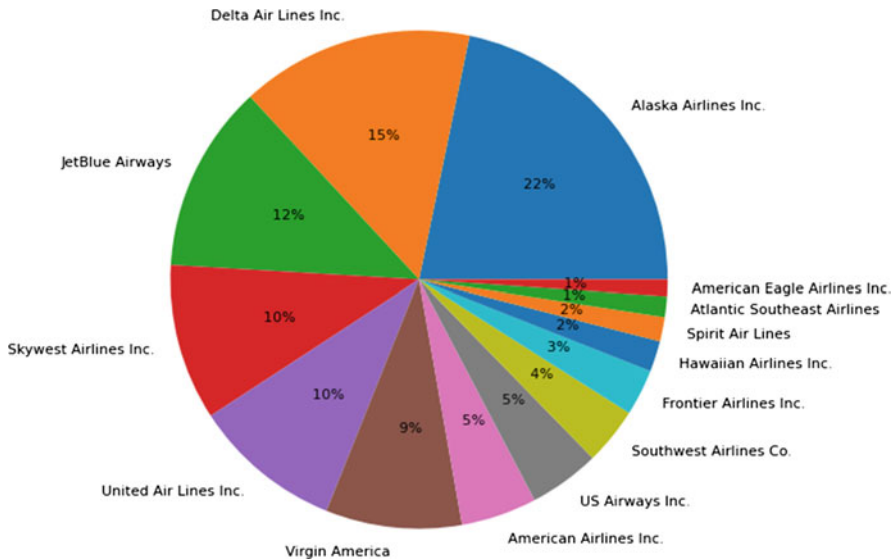
### 2.1 Data Preprocessing and Visualization

All in all, the three datasets originally contained 40 different columns. As the first step in processing the data, all the rows and columns containing missing values were deleted, and the three different datasets were joined together, forming a single dataset for creating a model. In order to connect the separate datasets, information regarding airline identifiers and origin/destination airports have been used in combinations with Inner Joins. Using Inner Joins, the rows from the two different datasets are sustained as long as there is a match in both datasets along the identifier



**Table 3** Dataset information

Dataset Columns	
Month	Scheduled Time
Day	Elapsed time
Day of week	Air time
Flight number	Distance
Tail number	Wheels on
Origin airport	Taxi in
Destination airport	Scheduled arrival
Scheduled departure	Arrival time
Departure time	Arrival delay
Departure delay	Diverted
Taxi out	Cancelled
Wheels off	Airline



**Fig. 8** Airlines considered in the provided dataset. Source: Author’s own illustration

column. Finally, duplicate and unnecessary columns were dropped (e.g. airports’ latitude and longitude), resulting in a processed dataset consisting of 24 columns (Table 3) and 5,222,000 rows.

The final objective of this exercise will be to predict the overall “Arrival Delay”. Referring to Table 3, columns such as planned departure time and origin and destination airports can be assumed to play an important role in building a successful model.

A summary of the distribution of the 14 different airlines considered in this dataset is available in Fig. 8.

## 2.2 Modelling

After preprocessing the dataset, the categorical variables (e.g. text data) were converted into a numeric format, and the features were standardized by removing the overall mean and scaling the unit variance. Feature standardization is a commonly used technique to ensure that each feature has a zero mean and unit variance. These characteristics can, in turn, help machine learning models train better by establishing that all the different features lie within the same range.

In order to predict flight delays, a Least Absolute Shrinkage Selector Operator (LASSO) Regression model with a train-test split of 70% and 30% each and a threefold cross-validation will be applied. This setup will then form a basis for the various tuning approaches, which were introduced earlier in the theoretical foundations' section, to be administered.

LASSO is a particular type of regression technique aiming to reduce the overall residual error. This is achieved by adding a constraint on the parameters so that the coefficient values of the variables, which barely provide any value to the model, are reduced towards zero. One of the key parameters of LASSO regression is, therefore, the weight assigned to the constraint parameter (**alpha**). The higher the value of alpha, the more weight the constraint in the model will have (if alpha is equal to zero, the result would be a traditional ordinary least square model).

Finally, the Mean Squared Error (MSE) will be used as a metric to assess the performance of the model:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Making use of Scikit-learn LASSO default parameters, an overall loss of 97.775 is registered. Scikit-learn, by default, makes use of an **alpha** value equal to 1, and a **maximum number of iterations** allowed equal to 1000. The maximum number of iterations hyperparameter represents, in this case, how many iterations the algorithm is allowed to use as an upper limit (in case of faster convergence, not all iterations need to be used).

### 2.2.1 Manual Search

Taking a Manual Search approach, it could be interesting to test if increasing the number of iterations (1500) and slightly reducing the alpha value (0.9) would lead to better outcomes. Reducing the weight assigned to the constraining parameter and allowing the model to train over a greater number of iterations could potentially make it easier for the model to converge to a lower score.

Making use of these parameters, an overall score of 96.853 is registered. Therefore, these first results demonstrate that better hyperparameter value combinations are indeed possible for this type of problem. In order to probe this hunch even further, more advanced techniques such as Grid and Random Search will now be examined.

### 2.2.2 Grid Search

Grid Search can be implemented easily into Python, making use of Scikit-learn GridSearchCV function. Hereby, merely a dictionary containing the ranges of values (to get the selected values and the loss function of choice from) is needed to designate which model to use (LASSO Regression). In this implementation, the negative mean squared error is specified instead of the mean squared error since, by default, Scikit-learn aims to maximize rather than minimize the objective function. Additionally, the **fit\_intercept** hyperparameter is introduced in order to test whether calculating the intercept for the model is the most suitable option. By trying to optimize and test against more of the different available hyperparameters, this could, in fact, lead to a potential reduction in the overall loss by enabling new possible combinations.

```

1. from sklearn.model_selection import GridSearchCV
2.
3. grid_search = {
4.     'alpha': list(np.linspace(0.7, 2.0, 5, dtype = float)),
5.     'fit_intercept': ['True', 'False'],
6.     'max_iter' : list(np.linspace(1000, 1800, 5, dtype = int)),
7. }
8.
9. clf = Lasso()
10. model = GridSearchCV(estimator = clf, param_grid = grid_search,
11.                      scoring='neg_mean_squared_error',
12.                      cv = 3, verbose= 5, n_jobs = -1)
13. model.fit(X_Train,Y_Train)
14.
15. predictions = model.best_estimator_.predict(X_Test)
16. mean_squared_error(Y_Test, predictions)

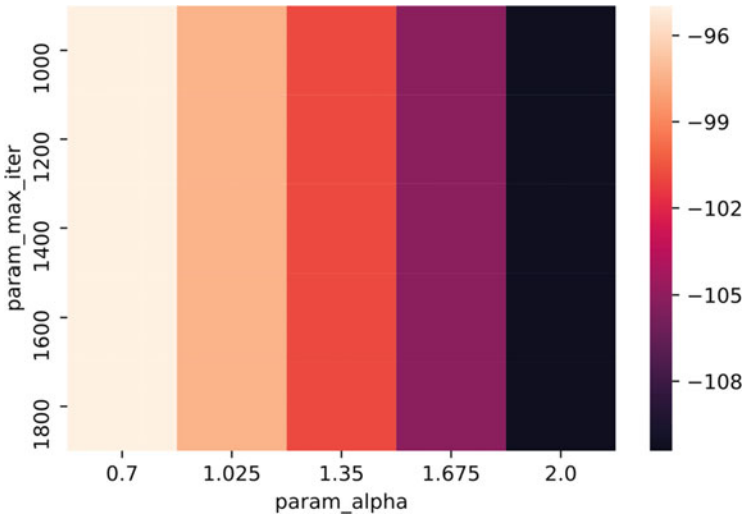
```

Once the model was trained/tested, a loss of 95.251 and the hyperparameter values in Table 4 were registered:

To conclude, using a heatmap, as shown in Fig. 9, additionally allows for a visualization of how changing different hyperparameters (alpha vs. maximum number of iterations allowed) affected the overall loss during the model training.

**Table 4** Grid Search identified hyperparameters

Alpha	Fit Intercept	Max Iterations Allowed
0.7	True	1000



**Fig. 9** Heatmap depicting overall loss due to hyperparameter changes. Source: Author’s own illustration

### 2.2.3 Random Search

Through the use of Grid Search, it was possible to reduce the overall loss once again. Now, the same procedure can be repeated, yet this time using Random Search instead. Focusing the search area around the best hyperparameter set identified thus far will aid in establishing if it is likely to reduce the overall loss even more.

```

1. from sklearn.model_selection import RandomizedSearchCV
2.
3. random_search = {
4.     'alpha': list(np.linspace(0.3, 1.5, 5, dtype = float)),
5.     'fit_intercept': ['True', 'False'],
6.     'max_iter': list(np.linspace(700, 1200, 5, dtype = int)),
7. }
8.
9. clf = Lasso()
10. model = RandomizedSearchCV(estimator = clf, param_distributions = random_search,
11.                            scoring='neg_mean_squared_error',
12.                            cv = 3, verbose= 5, random_state= 101, n_jobs = -1)
13. model.fit(X_Train,Y_Train)
14.
15. predictions = model.best_estimator_.predict(X_Test)
16. mean_squared_error(Y_Test, predictions)

```

In this case, a loss of 18.175 and the hyperparameter values in Table 5 were scored:

**Table 5** Random Search identified hyperparameters

Alpha	Fit Intercept	Max Iterations Allowed
0.3	True	825

As a result, Random Search demonstrated that it was possible to score a considerably lower loss compared to the other methods. At this point, only the last two techniques introduced in this chapter, Bayesian Optimization and Genetic Algorithms, remain to be tested.

## 2.2.4 Bayesian Optimization

In this example, the hyperopt library (Bergstra et al., 2013) will be used in order to implement the Bayesian Optimization routine. Alternative libraries, such as Optuna, are also available in Python (Akiba et al., 2019).

Hyperopt allows for an easy implementation of Bayesian Optimization by calling the `fmin()` function and specifying its three key parameters:

1. **Objective Function:** The chosen loss function to minimize.
2. **Domain Space:** The range of different hyperparameter values to test.
3. **Optimization Algorithm:** The search algorithm used to find the best hyperparameter values to use.

Once the best possible combination of hyperparameters is found, a standard LASSO model can be administered to test the results.

```

1. from hyperopt import hp, fmin, tpe, STATUS_OK, Trials
2. from sklearn.model_selection import cross_val_score
3.
4. space = {
5.     'alpha': hp.quniform('alpha', 0.3, 1.5, 5),
6.     'fit_intercept': hp.choice('fit_intercept', ['True', 'False']),
7.     'max_iter': hp.quniform('max_iter', 700, 1200, 5)
8. }
9.
10. def objective(space):
11.     model = Lasso(alpha = space['alpha'],
12.                  fit_intercept = space['fit_intercept'],
13.                  max_iter = space['max_iter'])
14.
15.     loss = cross_val_score(model, X_Train, Y_Train, cv = 3,
16.                           scoring='neg_mean_squared_error').mean()
17.     return {'loss': -loss, 'status': STATUS_OK }
18.
19. trials = Trials()
20. best = fmin(fn= objective,
21.            space= space,
22.            algo= tpe.suggest,
23.            max_evals = 30,
24.            trials= trials)
25.
26. fit_intercept = {0: 'True', 1: 'False'}
27.
28. lasso = Lasso(alpha = best['alpha'],
29.              fit_intercept = fit_intercept[best['fit_intercept']],
30.              max_iter = best['max_iter'])
31. scores = cross_validate(lasso, X, Y, cv=3, scoring='neg_mean_squared_error')
32. np.abs(np.mean(scores['test_score']))

```

**Table 6** Bayesian Optimization identified hyperparameters

Alpha	Fit Intercept	Max Iterations Allowed
0.0	True	1125

Using this approach, a loss of 0.00214 and the hyperparameter values in Table 6 were selected:

### 2.2.5 Genetic Algorithms

In this case, the TPOT auto machine learning library (Olson et al., 2016) will be the library of choice as it is based on the Scikit-learn library and can be used for either regression or classification tasks. For this purpose, a population size of 24 models, which were trained over five different generations, was specified.

```

1. from tpot import TPOTRegressor
2.
3. parameters = {
4.     'alpha': list(np.linspace(0.3, 1.5, 5, dtype = float)),
5.     'fit_intercept': ['True', 'False'],
6.     'max_iter' : list(np.linspace(700, 1200, 5, dtype = int)),
7. }
8.
9. tpot_regressor = TPOTRegressor(generations= 5, population_size= 24,
10.                               offspring_size= 12,
11.                               verbosity= 2, early_stop= 12,
12.                               config_dict=
13.                               {'sklearn.linear_model.Lasso': parameters},
14.                               cv = 3, scoring = 'neg_mean_squared_error')
15. tpot_regressor.fit(X_Train,Y_Train)
16.
17. res = np.abs(tpot_regressor.score(X_Test, Y_Test))
18. res

```

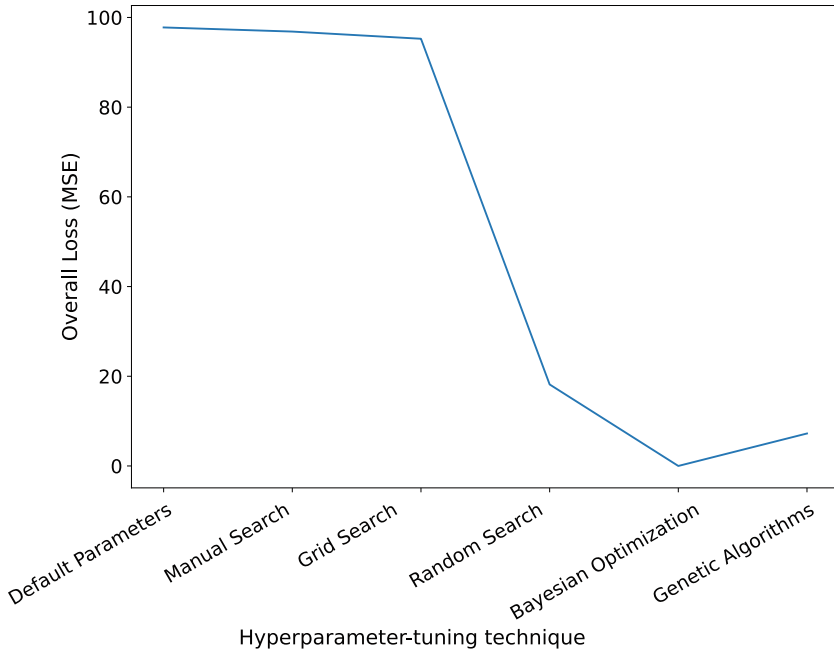
Finally, using this technique, a loss of 7.240 was achieved.

### 2.2.6 Conclusion

For this demonstration, the Bayesian Optimization outperformed the other methods regarding this type of task (see Table 7 and Fig. 10). Nonetheless, the results

**Table 7** Comparison of hyperparameter tuning techniques

Hyperparameter tuning technique	Overall Loss (MSE)
Default parameters	97.775
Manual search	96.853
Grid search	95.251
Random search	18.175
Bayesian optimization	0.00214
Genetic algorithms	7.240



**Fig. 10** Overall loss (MSE) vs. Hyperparameter tuning technique. Source: Author's own illustration

obtained were highly dependent on the chosen grid space and dataset used. Therefore, in different situations, various optimization techniques will perform better than others.

### 3 Research Case

The tourism sector has undoubtedly been one of the main drivers of economic growth in the past decades. As of 2018, the tourism sector on its own has contributed to about 10.4% of the worldwide gross domestic product (WEF, 2019). In order to improve the tourism industry further, one key area of interest has included forecasting tourism demand. As a matter of fact, accurate forecasting can be incredibly beneficial for organizations to optimize product pricing and staff capacity as well as for governments to plan for construction of new tourism infrastructures (Frechtling, 2001; Li et al., 2006).

Many different approaches, ranging from using traditional statistical methods to more advanced machine learning and deep learning techniques, have been proposed in the last few years trying to tackle this task. In Kulshrestha et al. (2020), for instance, a new approach on a Singaporean tourism dataset using a Bayesian

Bidirectional Long Short-Term Memory (BILSTM) network was introduced. This approach adopts the traditional Long Short-Term Memory neural network (Hochreiter & Schmidhuber, 1995) and expands its functionality by enabling it to take advantage of both forward and backward information (i.e. future and past). Thus, this type of setup makes the BILSTM network an optimal choice when it comes to working with data accompanying long-term time dependencies.

Accordingly, as part of this study, Bayesian Optimization was selected as the preferred approach for optimizing the model hyperparameters. In fact, Bayesian Optimization managed to achieve appreciable results against different evaluation metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|$$

In this case, hyperparameters such as the **learning rate**, **number of neurons**, **L2 regularization**, and **dropout probability** were considered. Large learning rates can lead to fast convergence towards a sub-optimal solution, while a small learning rate can cause a slow convergence towards any type of sub-optimal/optimal solution. Moreover, using few neurons can potentially lead to overfitting; on the other hand, using too many neurons might become computationally expensive (Guler et al., 2005). L2 regularization and dropout probability, however, can be used to try to prevent such overfitting (Phaisangittisagul, 2016).

As a result of this study, the Bayesian optimized BILSTM was able to outperform other models, such as traditional LSTMs, Support Vector Regression, and the Autoregressive Distributed Lag Model (Nkoro & Uko, 2016).

All in all, it can be said that the application of hyperparameter tuning in tourism data analysis is widely used nowadays and for good reason. Other common examples include, amongst others, hotel review sentiment analysis (Nguyen-Thanh & Tran, 2019) and air transportation direct share analysis and forecasting (Zheng et al., 2020).

### Service Section

**Main Application Fields:** Hyperparameter Optimization is a main step in the Data Science track, which can be applied in any ambit and for any type of machine learning model to improve overall performance. Mainstream libraries and software packages usually provide valuable baseline solutions that work

(continued)



well with many applications. If you are interested in building machine learning pipelines quickly, without necessarily aiming to obtain the best possible results, then Hyperparameter Optimization can be overlooked.

**Limitations and Pitfalls:** The overall process of Hyperparameter Optimization can be both computationally expensive (in terms of computing required resources) and time-consuming.

**Similar Methods and Methods to Combine with:** Hyperparameter Optimization is considered to be a first step towards automatic machine learning and relies on traditional mathematical optimization techniques as a foundation.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-12-Hyperparameter-Tuning>

## Further Readings and Other Sources

- Feurer, M., & Hutter, F. (2019). *Hyperparameter optimization. Automated machine learning* (pp. 3–33). Springer. [https://doi.org/10.1007/978-3-030-05318-5\\_1](https://doi.org/10.1007/978-3-030-05318-5_1)
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>

## References

- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A next-generation hyperparameter optimization framework*. Retrieved from <https://arxiv.org/pdf/1907.10902.pdf>
- Bayesian Hyperparameter Optimization - A Primer on Weights & Biases. (n.d.). [Wandb.com](http://www.Wandb.com). Retrieved November 8, 2020, from <https://www.wandb.com/articles/bayesian-hyperparameter-optimization-a-primer>
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (n.d.). *Algorithms for hyper-parameter optimization*. Retrieved from <https://papers.nips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf>
- Bergstra, J., Ca, J., & Ca, Y. (2012). Random search for hyper-parameter optimization Yoshua Bengio. *Journal of Machine Learning Research*, 13, 281–305. Retrieved from <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf>
- Bergstra, J., Yamins, D., & Cox, D. (2013). *Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures*. 28. Retrieved from <http://proceedings.mlr.press/v28/bergstra13.pdf>
- Bossek, J., Doerr, C., & Kerschke, P. (2020). Initial design strategies and their effects on sequential model-based optimization. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. <https://doi.org/10.1145/3377930.3390155>.
- Dahl, G., Sainath, T., & Hinton, G. (n.d.). *Improving deep neural networks for LVCSR using rectified linear units and dropout*. Retrieved from [http://www.cs.utoronto.ca/~gdahl/papers/reluDropoutBN\\_icassp2013.pdf](http://www.cs.utoronto.ca/~gdahl/papers/reluDropoutBN_icassp2013.pdf)

- Department of Transportation. (2015). *2015 Flight delays and cancellations*. Retrieved from Kaggle.com website: <https://www.kaggle.com/usdot/flight-delays>
- Eggenesperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., & Leyton-Brown, K. (n.d.). *Towards an empirical foundation for assessing Bayesian optimization of Hyperparameters*. Retrieved from <https://www.cs.ubc.ca/~hoos/Publ/EggEtAl13.pdf>
- Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. In *Automated machine learning* (pp. 3–33). Springer. [https://doi.org/10.1007/978-3-030-05318-5\\_1](https://doi.org/10.1007/978-3-030-05318-5_1)
- Frechtling, D. C. (2001). *Forecasting tourism demand: Methods and strategies*. Butterworth-Heinemann.
- Guler, N., Ubeyli, E., & Guler, I. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications*, 29(3), 506–514. <https://doi.org/10.1016/j.eswa.2005.04.011>
- Hochreiter, S., & Schmidhuber, J. (1995). *Long short term memory*. München Inst. Für Informatik.
- Hutter, F., Hoos, H., Leyton-Brown, K., & Ca, K. (n.d.). *An efficient approach for assessing Hyperparameter importance*. Retrieved August 3, 2020, from <http://proceedings.mlr.press/v32/hutter14.pdf>
- Kohavi, R., & John, G. H. (1995, January 1). *Automatic parameter selection by minimizing estimated error* (A. Prieditis & S. Russell, Eds.). Retrieved November 8, 2020, from ScienceDirect website: <https://www.sciencedirect.com/science/article/pii/B9781558603776500451>
- Kulshrestha, A., Krishnaswamy, V., & Sharma, M. (2020). Bayesian BILSTM approach for tourism demand forecasting. *Annals of Tourism Research*, 83, 102925. <https://doi.org/10.1016/j.annals.2020.102925>
- Li, G., Wong, K. K. F., Song, H., & Witt, S. F. (2006). Tourism demand forecasting: A time varying parameter error correction model. *Journal of Travel Research*, 45(2), 175–185. <https://doi.org/10.1177/0047287506291596>
- Luo, G. (2016). A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1), 1–16. <https://doi.org/10.1007/s13721-016-0125-6>
- Melis, G., Dyer, C., & Blunsom, P. (n.d.). *On the state of the art of evaluation in neural language models*. Retrieved from <https://arxiv.org/pdf/1707.05589.pdf>.
- Montgomery, D. C. (2020). *Design and analysis of experiments*. Wiley.
- Neural, A., & Mehlig, N. (2019). *Lecture notes*. Retrieved from <https://arxiv.org/pdf/1901.05639.pdf>
- Nguyen-Thanh, T., & Tran, G. T. C. (2019). Vietnamese sentiment analysis for hotel review based on overfitting training and ensemble learning. In *Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019*. <https://doi.org/10.1145/3368926.3369675>.
- Nkoro, E., & Uko, A. (2016). Autoregressive Distributed Lag (ARDL) cointegration technique: Application and interpretation. *Journal of Statistical and Econometric Methods*, 5(4), 1792–6939. Retrieved from [http://www.scienpress.com/Upload/JSEM/Vol%205\\_4\\_3.pdf](http://www.scienpress.com/Upload/JSEM/Vol%205_4_3.pdf)
- Olson, R. S., Urbanowicz, R. J., Andrews, P. C., Lavender, N. A., Kidd, L. C., & Moore, J. H. (2016). Automating biomedical data science through tree-based pipeline optimization. In *Applications of evolutionary computation* (pp. 123–137). Springer. [https://doi.org/10.1007/978-3-319-31204-0\\_9](https://doi.org/10.1007/978-3-319-31204-0_9)
- Phaisangittisagul, E. (2016). An analysis of the regularization between L2 and dropout in single hidden layer neural network. In *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*. <https://doi.org/10.1109/isms.2016.14>.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175. <https://doi.org/10.1109/jproc.2015.2494218>

- Snoek, J., Larochelle, H., & Adams, R. (n.d.). *Practical Bayesian optimization of machine learning algorithms*. Retrieved from <https://papers.nips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf>
- Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., . . . Adams, R. (2015). Scalable Bayesian optimization using deep neural networks Prabhat PRABHAT@LBL.GOV. 37. Retrieved from <http://proceedings.mlr.press/v37/snoek15.pdf>
- Sui, Y., Zhuang, V., Burdick, J., & Yue, Y. (n.d.). *Stagewise safe Bayesian optimization with Gaussian processes*. Retrieved from <https://arxiv.org/pdf/1806.07555.pdf>
- WEF. (2019). *The travel & tourism competitiveness report 2019*. Retrieved from [http://www3.weforum.org/docs/WEF\\_TTCR\\_2019.pdf](http://www3.weforum.org/docs/WEF_TTCR_2019.pdf)
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295–316. <https://doi.org/10.1016/j.neucom.2020.07.061>
- Zheng, X., Liu, C.-M., & Wei, P. (2020, February 1). *Air transportation direct share analysis and forecast*. Retrieved December 6, 2020, from Journal of Advanced Transportation website: <https://www.hindawi.com/journals/jat/2020/8924095/>

# Model Evaluation



## How to Accurately Evaluate Predictive Models

Ajda Pretnar Žagar and Janez Demšar

### Learning Objectives

- Describe what model evaluation is and why it is important
- Explain what each evaluation score means and how to apply it in practice
- Show how to accurately evaluate predictive models accurately
- Demonstrate a typical model evaluation workflow on a case study

## 1 Introduction

Model evaluation is key to understanding how useful a model is in real life. It estimates how well the model will perform on new data, with different scores estimating different aspects of the model. Suppose that one would like to create a model for predicting the long-term profitability of small hotels based on their type, location, size, and other properties. As another example, one may have some well-defined types of tourists, and the goal is to classify a person based on her travelling history. To approach these problems using predictive modelling, one first needs to collect an appropriate data set: a collection of data instances, described by the available features (*attributes* in machine learning or *independent variables* in statistics), and the associated outcome (*class*, *label*, or *dependent variable*) for each data instance. The goal is to create a model capable of predicting the outcome based on the given features. If the outcome is numeric (e.g. hotel profitability), this is a regression problem (see Chapter ‘Regression’). If it is categorical (e.g. type of tourist), it is known as classification (see Chapter ‘Classification’).

---

A. Pretnar Žagar (✉) · J. Demšar  
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia  
e-mail: [ajda.pretnar@fri.uni-lj.si](mailto:ajda.pretnar@fri.uni-lj.si); [janez.demsar@fri.uni-lj.si](mailto:janez.demsar@fri.uni-lj.si)

The next step is to choose the type of model and the algorithm for fitting it (or, in machine learning parlance, for *training* or *learning*). The choice will depend upon the problem (regression or classification), its complexity, the size of the collected data set, and the number and types of features. There are many models and learning algorithms to choose from, but no generally applicable rules for selecting the modelling technique based on the properties of the problem and the data. In actual practice, it is best to try several approaches and select the one that results in the optimal model.

Models can be evaluated based on several criteria. Subjectively, models that are easy to understand, simple to use in practice, and cheap, in terms of the data that need to be collected, are preferred. Otherwise, in many cases, simply opting for models traditionally used in a particular domain is a good solution. In this chapter, we will focus on an objective assessment of model quality, that is, different scores that measure the expected behaviour of the model when used in practice. Classification and regression problems use different sets of scores; therefore, we will start the chapter with one section for each type of problem. We will then expose the problem of overfitting the model, which is usually reflected in good scores on training data but bad performance on new data. We shall dedicate a section to detecting and avoiding these problems and, thereafter, conclude with a practical example of predicting cancellations of hotel bookings.

## 2 Performance of Classification Models

Most classification models do not directly predict classes but, rather, compute a score reflecting the probability that the given data instance belongs to the target class. Alternatively, in problems with non-binary classes (i.e. more than two class labels), the model computes a score for each possible class. In some models, such as logistic regression, these scores directly model probabilities. In others, there is a monotone relation between the score and the probability, with higher scores corresponding to higher probabilities. In such cases, methods like Platt scaling (Platt, 2000) can be used to translate scores into probabilities. Therefore, one can assume that the model outputs the probability (or probabilities) that a particular instance belongs to the target class (or to each of the classes).

To make predictions, thresholds are placed on class probabilities. In the case of non-binary classification problem, the class with the highest probability is typically chosen. Binary classification, on the other hand, is more common and more interesting for the performance assessment. Here, a data instance is classified as positive if the predicted probability of belonging to the positive (or target) class equals or exceeds the threshold. The threshold is chosen so that the model yields the highest possible performance in terms of some desired metric (Hernandez-Orallo et al., 2012). It may often be beneficial to set a threshold different from 0.5, in other words, to predict the data instance as positive although its probability is very low (consider the detection of terrorist attacks) or to require a very high probability for

classifying an instance as positive (consider screening candidates for a risky surgery) (Elkan, 2001).

Performance scores can be separated into two groups: (1) those that measure the performance of the model at a certain operating point, that is, with some fixed (usually optimized) threshold, and (2) those that measure the performance of the model before setting such threshold, or a general performance over all possible operating points (Drummond & Holte, 2006).

### 2.1 Performance at Fixed Operating Conditions

By fixing the probability threshold, the model does not yield probabilities but predictions. These can be summarized in the form of a confusion matrix, which serves as a basis for the computation of various scores. An example of such a matrix is shown in Table 1, showing a synthetic data set on whether or not a person purchased a museum ticket. There are 20 data instances in which 11 cases indicate that a ticket was not purchased and 9 cases reveal the person did purchase one. The classifier predicted that in 15 cases, the person did not buy the ticket, and in 5, the person did.

The confusion matrix shows correct class labels in rows and predicted class labels in columns. Each cell shows the number of data instances for the actual and the predicted class. Diagonal cells correspond to correctly predicted data instances; for example, nine instances from the negative class were correctly classified as negative (true negative). Off-diagonal cells correspond to misclassified instances, for example two positive instances that should have been recognized as negative (false positive).

Binary classification will be mostly considered. To distinguish between being classified as positive and being actually positive, the former instance will be referred to as *positive* and the latter as *condition positive* (i.e. actually having the condition).<sup>1</sup> Positive instances are called *true positives* (TP), if they are also condition positive, or else *false positives* (FP). Similarly, *true negatives* are defined as instances that are correctly predicted as negative (TN), and *false negatives* (FN) are instances that are predicted as negative although they have the observed condition.

**Table 1** Confusion matrix

		Predicted		$\Sigma$
		No	Yes	
Actual	No	9	2	<b>11</b>
	Yes	6	3	<b>9</b>
$\Sigma$		<b>15</b>	<b>5</b>	<b>20</b>

<sup>1</sup>In the context of binary classification, positive means belonging to the class label 1, whatever that may signify. It can be having a disease, purchasing a product, booking a hotel, etc. Positive class is arbitrary and can be reversed, i.e. not having a disease, not purchasing a product.

### 2.1.1 Classification Accuracy

Classification accuracy (CA) reports the percentage of correctly classified instances. In our museum example from Table 1, classification accuracy would be  $(9 + 3)/20 = 0.6$ , or the number of correctly predicted instances divided by the total number of instances. The formula for classification accuracy is as follows,

$$CA = \frac{TP + TN}{N}$$

with TP as the number of true positive instances, TN the number of true negative instances, and N the number of all instances in the model.

Classification accuracy is the first score that comes to mind because it seems to match our intuitive understanding of ‘correctness’ of the model. Another advantage is that it is well-defined for non-binary classification problems as well. However, it also has significant drawbacks. The first is the lack of absolute scale. Consider a rare disease with only 1% of condition positive instances. While classification accuracy of 97% seems excellent, in this context, a model can reach 99% accuracy by simply classifying all instances as negative. The second problem is that classification accuracy is rarely the score that one wishes to optimize. In the case of rare diseases, the goal may be to detect possible occurrences. Therefore, one is more willing to accept a considerable number of false positives to detect as many condition positive instances as possible. Even without larger imbalances in class distribution, different types of errors may have different costs. In those cases, checking the performance scores that measure the probabilities of making different kinds of errors is necessary.

### 2.1.2 Recall (Sensitivity)

Consider a travel agent who wishes to optimize the conversion rate (people booking his/her services) so that the non-converting visitors can be offered an incentive to convert (i.e. a discount). Say 45% of the people who browse the agent’s website do not purchase any services. If the agent optimizes classification accuracy only and looks for an 80% CA, there is still a chance of misidentifying two-fifths of the non-converting visitors. Therefore, not only does the agent have to be correct over a certain percentage of the time, but he/she should also focus more on the website visitors who may not purchase any services. This class will be denoted as the target, that is, the positive class.

In such an example, the aim is to optimize the percentage of all instances from the target class that were detected. The corresponding score is called *recall*, or, in some areas such as medicine, *sensitivity* (Fig. 1). This is computed as the number of true positive instances divided by the number of all instances in the target class (true positives and false negatives).

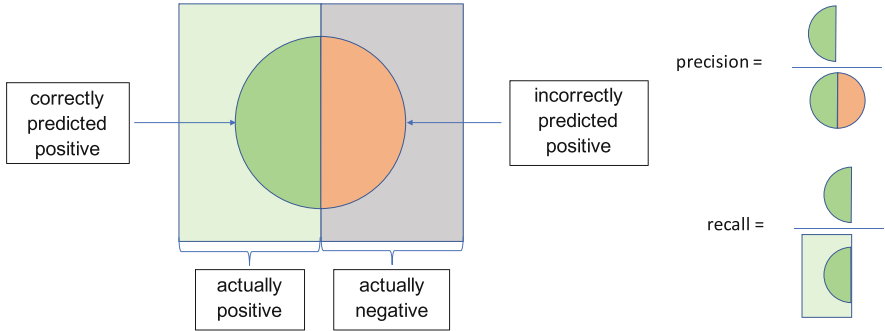


Fig. 1 Illustration of precision and recall ('Precision and recall', n.d.)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

In the confusion matrix shown earlier in Table 1, only 3 out of 9 instances in the target class were classified as positive; hence, the recall (or sensitivity) is  $3/9 \approx 0.33$ .

### 2.1.3 Precision

The travel agent, however, cannot offer discounts to everyone visiting the website as this would cost her/him revenue. For instance, what if the visitor books anyway, although the model predicted this would not be the case, and the agent offered the discount? The agent also has to consider how many visitors predicted as positive are indeed in the target class. This score is called *precision* and is computed by dividing the true positives by all positives (both true and false):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

In the above confusion matrix, 3 out of 5 positive instances are in the target class; hence, precision equals  $3/5 = 0.6$ .

### 2.1.4 F1

In essence, the agent needs to know how well the model identifies non-converting visitors among all visitors and, at the same time, how well the model identifies non-converting visitors among all those predicted as non-converting. To compute the mean of the two values, the harmonic mean is used, which is more suitable for



computing average rates than the common arithmetic mean. This combination of the two measures is called *F1*.

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Thus, the harmonic mean of the above precision (0.6) and recall (0.33) is approximately 0.46.

### 2.1.5 Specificity

A common pair of counter-balancing measures, recall and precision, have been presented. These are used to identify certain types of instances within a large pool of data, for example retrieving documents of interest from a large library. The task in many other fields is to distinguish between instances of two kinds, such as finding people infected by a specific disease vs. those who are healthy. In this context, one speaks of sensitivity (synonymous with recall) and specificity. *Specificity* is defined as the ratio of true negative instances to all instances that are not in the target class and are indeed negative.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

As per our example, specificity would equal  $9/11 \approx 0.82$ .

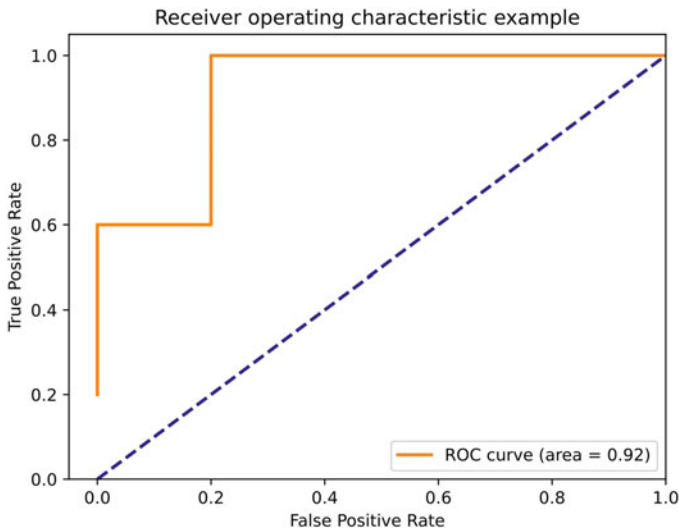
## 2.2 ROC Curves, P-R Curves

Confusion matrices are based on positive and negative predictions. The models usually predict probabilities (or a score that can be turned into probabilities), and actual predictions only arise after setting a probability threshold. The threshold does not need to be at  $p = 0.5$ ; a better threshold may be chosen according to some utility function. Different thresholds yield different confusion matrices and, in turn, different values for the scores described above. For example, in the context of tourism, a hotel manager might want to decrease the threshold for detecting fraudulent bookings if the hotel has recently received a high volume of such transactions. Alternatively, he/she might increase the threshold when the number of such transactions is low in order to avoid burdening the hotel employees with additional checks.

To facilitate the choice of the threshold, it is useful to depict a pair of scores, such as precision and recall or sensitivity and specificity, in the form of a parametric curve that shows the relation between them over the whole range of possible thresholds. This approach will be demonstrated by using the ROC curve as an example. Taking the previous travel agent data set, the model gives the probabilities shown in Table 2.

**Table 2** Evaluation results of decision tree classifier on the travel agent data set. The samples are sorted by the probability of class label yes (i.e. the client books the service)

Booked	p(booking)	Sensitivity	1 – Specificity
Yes	1	0.2	0
Yes	1	0.4	0
Yes	0.83	0.6	0
No	0.77	0.6	0.2
Yes	0.75	0.8	0.2
Yes	0.51	1	0.2
No	0.49	1	0.4
No	0.33	1	0.6
No	0.25	1	0.8
No	0	1	1



**Fig. 2** Simple ROC curve

*Booked* is the true class label, and *p(booking)* depicts the probability of a person booking a service as predicted by the model. Most importantly, the table is sorted by decreasing probability.

A curve in Fig. 2 shows the false positive rate (1 – specificity) on the x-axis and the true positive rate (sensitivity) on the y-axis. Every point on this curve corresponds to a different possible threshold. Starting with a threshold of 1, the model would classify the first two samples (in a sorted table!) as positive, and they are indeed positive. Therefore, the sensitivity is 0.4 (2 ‘yes’ out of a total of 5), whereas specificity is 0 (0 ‘no’ out of a total of 5). The next possible threshold is 0.83 with one positive sample. There are 3 ‘yes’ out of a total of 5 and still 0 ‘no’ out of a total of 5. At threshold 0.77, the first incorrectly predicted sample can be found. Sensitivity then remains the same, but the false positive rate is 0.2 (1 ‘no’ out of a total of

5). We continue plotting the points representing the relationship between sensitivity and specificity until we get the curve of all the samples.

The curve plotted above is called a receiver-operating characteristic (ROC) curve (Fawcett, 2006). This is a simplistic example of the data from Table 2. Actual curves are smooth with more points to choose from; see Fig. 7 for a realistic curve. Note, once again, that the curve represents the relationship between true positive and false positive cases. If the sensitivity (true positive rate) goes up, so does the false positive rate, and the specificity, which is  $1 - \text{false positive rate}$ , goes down.

How would a ROC curve of a random classifier look like?

How would a ROC curve of a worse-than-random classifier look like?

How would a perfect ROC curve look like? Remember, we try to maximize the area under the curve.

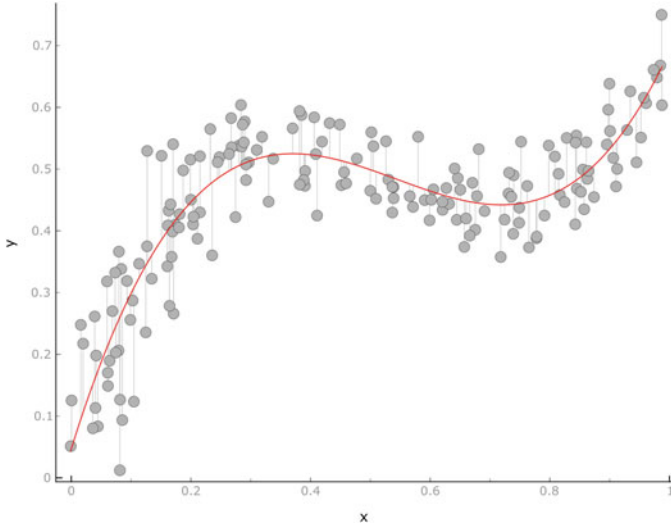
As mentioned above, specificity and sensitivity both rely on the threshold at which positive class labels are assigned. The curve allows us to choose an optimal operating condition, that is, the optimal combination of sensitivity and specificity that can be achieved with this model.

To express the overall performance of a model in regards to the relation depicted by this curve, the area under the curve, or *AUC*, can be computed. This score has several interesting properties. *AUC* represents the probability that, when choosing between a negative and positive data instance, the model will assign a higher probability of being positive to the instance that is indeed positive. Hence, the *AUC* measures the model's ranking abilities: if the data instances are ranked by their probability of being positive, models that produce a better ranking will have higher *AUC* scores.

For a model that makes random guesses, the probability to correctly identify the positive data instance in such a pair is 0.5. Hence, an *AUC* of 0.5 represents the worst possible model (any model worse than that can be improved by simply flipping its predictions). In ROC space, this corresponds to the diagonal curve. As such, the measure does not depend on prior class distributions. Even if one class contains a vast majority of cases, random guessing will still result in an *AUC* of 0.5. An *AUC* between 0.7 and 0.8 can be considered reasonably good, while anything above is excellent. One shortcoming of ROC curves and *AUC*, however, is that they are difficult to generalize to non-binary problems.

### 3 Regression

Unlike classification, regression predicts a numeric value, say, the price of a house. The goal of regression models is to find the function that best describes (fits) the data. Figure 3 shows the data with a fitted model, where the fitted curve represents the



**Fig. 3** Polynomial regression function for a simple synthetic data set

function of the predictive model. The vertical bars represent the error of the model (the distance of the true value to the predicted value). Here, one does not operate with probabilities but, rather, only with the differences between the predicted and the actual value.

### 3.1 Evaluation Scores

Regressors are typically evaluated with the following scores: MSE, RMSE, MAE, and  $R^2$ . Just as with classification, each score covers a different aspect of the model.

#### 3.1.1 Mean Square Error (MSE)

Mean square error measures the average of squared errors or deviations (also known as residuals). Deviations are the difference between the true and the predicted values. Say the true revenue per hotel room is 123,572 €, and the predicted value is 124,210 €. The difference is 638 €, meaning the square error is  $638^2 = 407,044$ . The score is the average of all squared deviations, with a smaller score signifying a better model.

### 3.1.2 Root Mean Square Error (RMSE)

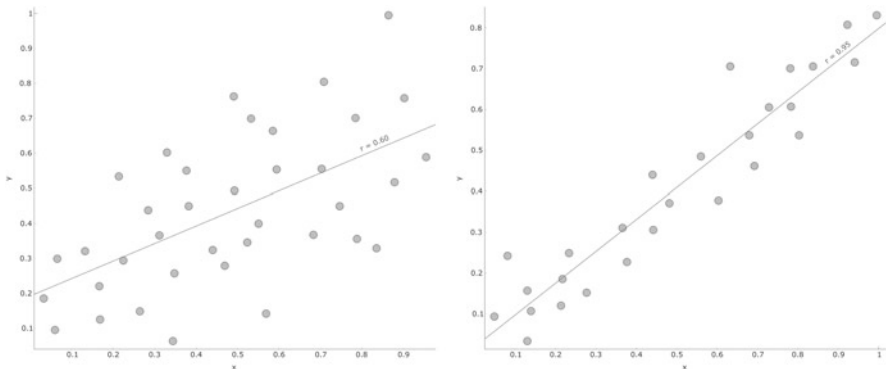
Mean square error is measured in squared units of quantity being measured; in the above case, in square euros. To put it on the same scale as the quantity, the square root of it is taken. The corresponding score is called the root mean square error, or RMSE.

### 3.1.3 Mean Absolute Error (MAE)

MAE is the average of the sum of absolute errors. The difference between MAE and RMSE is that RMSE is strongly affected by outliers, a small number of very bad predictions; the computed RMSE score can be almost entirely dependent on outliers because smaller squared terms do not contribute much to the overall sum. MAE does not exhibit this effect.

### 3.1.4 Coefficient of Determination ( $R^2$ )

In regression analysis, the coefficient of determination is one minus the proportion of the variance in the dependent variable that is predictable from the independent variable (Fig. 4). When evaluating the model, it is computed as the ratio between the variance (error) of predictions (also called *residual variance*) and the variance of the actual target variable (the *total variance*). The proportion is subtracted from one; hence, higher scores represent better models.



**Fig. 4** Low  $R^2$  score on the left – data points are scattered far apart from the regression line. High  $R^2$  score on the right – data points lie close to the regression line

## 4 Overfitting

Predictive modelling takes the data and identifies significant patterns that correlate the variables with the target variable. The scores presented above can measure the tightness of this fit, but the real question is regarding how the model will perform on new data. The model might achieve high scores by simply remembering all the samples and capturing every specificity, including the outliers, without identifying the underlying patterns in the data. Such a model would perform very well on the data to which it was fit but poorly on new data. This effect is called overfitting.

The key question is whether a model has learned something useful, i.e. did it find patterns that reflect the real world. Therefore, predictive modelling aims to train a model that would generalize well (Vapnik, 2000: 123; Kawaguchi et al., 2017). To increase the likelihood that the model will be useful, model evaluation has to test the model on an unseen data set. To achieve this, the model is first trained on one data set (training data) and then tested on another unseen data set (test data).

### 4.1 *Random Sampling*

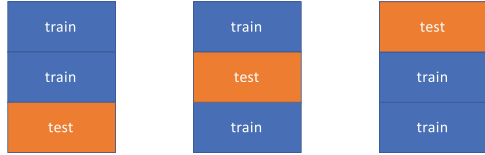
To split the data set into training and test sets, one can, for example take 70% of the data to train the model and save the remaining 30% for testing. The scores would be reported for the test data only, describing the model's performance on an unseen data set.

While this method is fast and simple, it suffers from two shortcomings. One is the possibility of having a different class distribution in training and testing data. This can be solved by stratified sampling. The other dilemma is its large variance and, consequently, low reliability because the test data can contain predominantly simple or predominantly difficult cases. The latter can be solved by repeated random sampling and reporting the average performance score. A single run of random sampling, though, is still useful in cases where the abundance of data allows for the creation of a large test set, guaranteeing better statistical validity.

### 4.2 *Cross-Validation*

Multifold cross-validation is a form of repeated sampling in which every data instance is used for testing exactly once. The data is split into several subsets, usually 5 or 10; Fig. 5 illustrates a threefold cross-validation where the data is split into three parts. In each pass, one subset is held out for testing, while the others are used for training. In the end, the average performance across all test sets is reported. Increasing the number of subsets decreases the variance yet increases the computation time.

**Fig. 5** Example of a threefold cross-validation

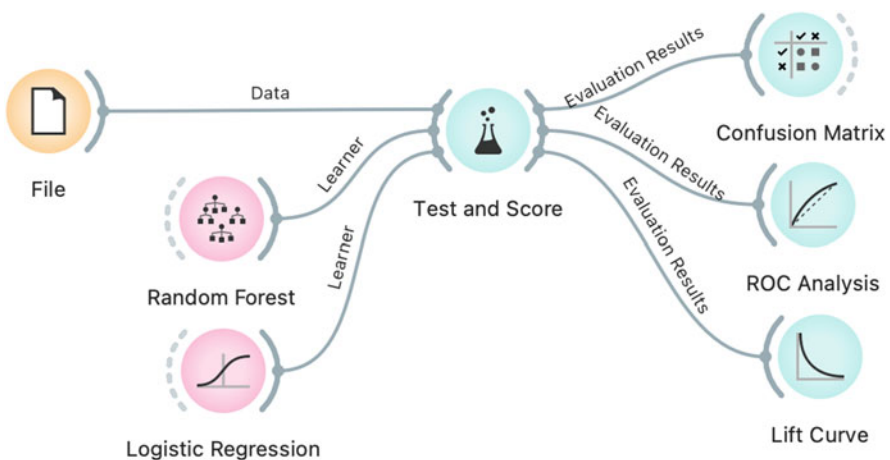


### 4.3 Leave-One-Out

All methods with multiple rounds of sampling and fitting produce a different model with each pass. Accordingly, the final model is an average of all the trained models and does not correspond to any particular model. A method that partially overcomes this is called leave-one-out (LOO) in which only a single instance is used for testing in each round. Thus, leave-one-out is equivalent to n-fold cross-validation, with n being the number of data instances. Due to the almost complete overlap of the training data, all derived models are supposed to be very similar. Such an evaluation is highly reliable yet very time consuming and only applicable to smaller data sets.

## 5 Practical Demonstration

In every predictive modelling situation (Fig. 6), one must first determine the type of problem – classification or regression – based on the variable one wishes to predict (target variable). Then, using one of the evaluation techniques, say cross-validation, one tests several models. The choice of the model(s) also depends on the type of problem.



**Fig. 6** Predictive modelling workflow

Evaluation procedure returns several evaluation scores, for instance, AUC, classification accuracy, precision, recall, and F1. They determine the quality of the model. However, scores are not the only way to observe the model’s performance.

### 5.1 Confusion Matrix

The confusion matrix has already been mentioned previously. It is a simple way to inspect what type of mistakes our model is likely to make. A high number of false negatives could indicate that the model has a high percentage of negative instances, and the algorithm for modelling does not handle it well.

### 5.2 ROC Curve

A ROC curve shows the model’s performance at different thresholds, which ultimately helps in determining the optimal threshold for each model. The cost of each type of mistake can also be adjusted; if the cost of a false positive is much higher than the cost of a false negative, the model will have to prioritize the true positive rate, shifting the ROC curve. In Fig. 7, the green curve has a better performance than the orange curve as its true positive rate is higher and false positive rate lower.

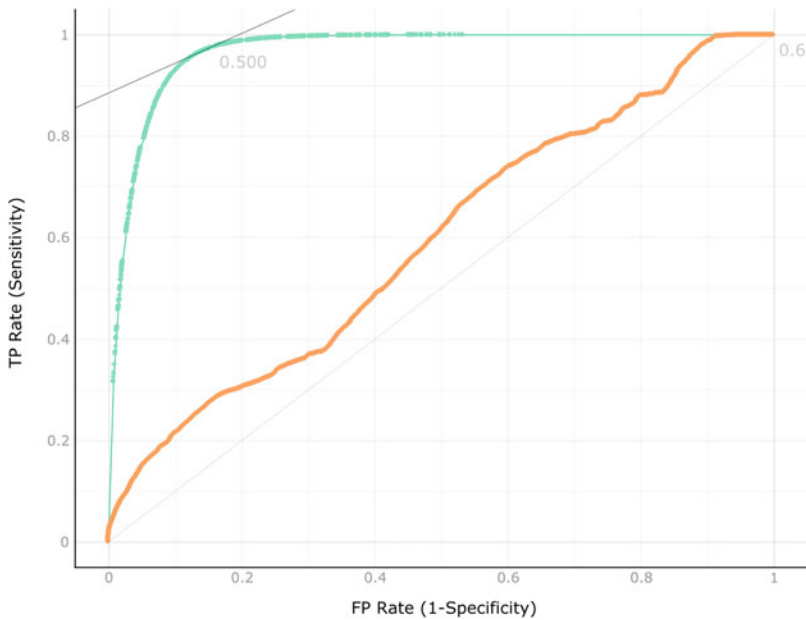
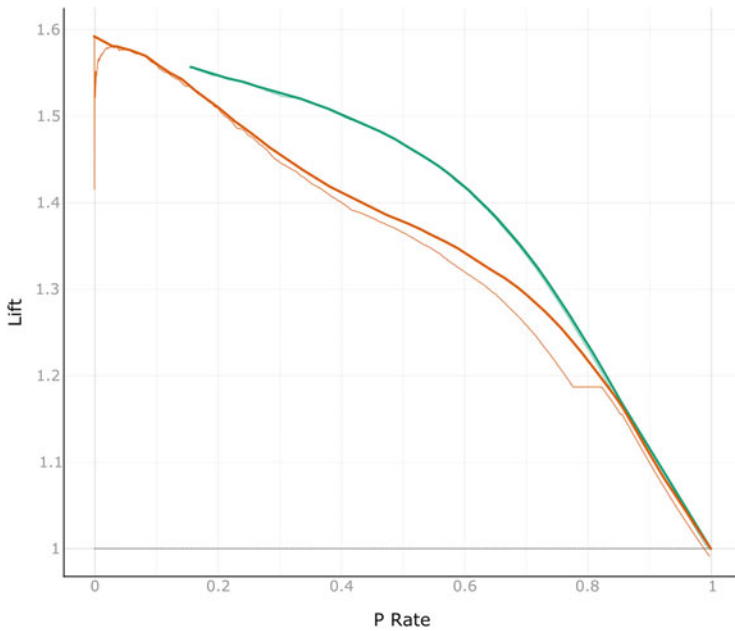


Fig. 7 ROC curve



### 5.3 Lift Curve

A lift curve shows how well the model identifies true positive instances, that is, if the sample with higher probabilities also has a large proportion of true positive instances. The curve shows the proportion of the samples from sorted probabilities on the x-axis, while the y-axis shows the lift, a ratio of true positives in the sample to the true positives in the data set. The curve is read from left to right; the higher the curve, the higher the lift, which means there is a high ratio of true positives in the sorted sample. In Fig. 8, the green curve shows the results with better performance. It starts later as the first couple of samples have the same probability. The orange curve shows an initial dip, which is a result of some negative samples having a high probability of being positive.



**Fig. 8** Lift curve plot

## 5.4 Data Over- or Undersampling

Lastly, a brief note on data oversampling. Sometimes the distribution of the target variable is highly imbalanced, with one (or some) class label(s) overwhelming the other(s). Common machine learning algorithms that optimize classification accuracy tend to do so at the cost of missing/not detecting instances of minority classes. It is a common temptation to over-represent instances of the minority class by oversampling or duplicating them in the data or by removing some instances of the majority class. These off-the-shelf approaches are usually unreliable. Undersampling reduces the model's ability to fit the structure of the data, and oversampling via duplication may, depending on the type of algorithm, often have no effect at all. In all cases, this leads to skewed models. A better approach is to manipulate the probability threshold or to use algorithms in order to detect rare events and outliers (Liu et al., 2008; Breunig et al., 2000; Carreño et al., 2020).

## 6 Research Case

To present a typical model evaluation workflow, we will use a hotel bookings data set. The data was collected by Antonio et al. (2019a) and is available on Kaggle<sup>2</sup> and GitHub.<sup>3</sup> This data set contains booking information for a city hotel as well as a resort hotel and includes information such as when the booking was made, the duration of the stay, the number of adults, children, and/or babies, the number of available parking spaces, etc. (Table 3). There are 29 features, 1 meta attribute, and 119,390 instances.

First, let us organize and explore the data. We will use the Orange data mining toolbox for this demonstration (Demšar et al., 2013). Set *is\_cancelled* as the target variable. Our aim will be to predict whether or not the booking will be cancelled. Looking at the data set, *reservation\_status* provides almost the same insights as the target variable, therefore, we will remove it from the data set. In the File widget, double-click the role of the variable to change it.

The most important thing in model evaluation is to never test with the same data on which the model was trained. Before training the model, we have to split the data into training data and test data using Data Sampler. We will use 70% of the data for training and 30% of the data for testing.

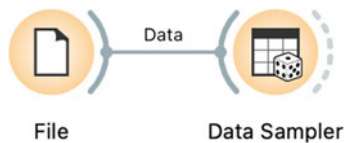
---

<sup>2</sup><https://www.kaggle.com/jessemostipak/hotel-booking-demand>

<sup>3</sup><https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-02-11/readme.md>

**Table 3** Description of variables for the hotel bookings data set

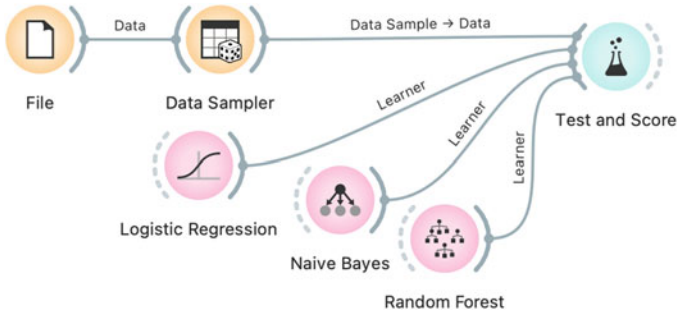
variable	meaning
hotel	City hotel or resort hotel
is_canceled	Whether a booking is cancelled (1) or not (0)
lead_time	Days elapsed between booking and arrival date
arrival_date_year	Year of arrival
arrival_date_month	Month of arrival
arrival_date_week_number	Week of arrival
arrival_date_day_of_month	Day of arrival
stays_in_weekend_nights	Number of booked weekend nights
stays_in_week_nights	Number of booked week nights
adults	Number of adults
children	Number of children
babies	Number of babies
meal	Type of meal booked
country	Country of origin
market_segment	Origin of booking (TA = travel agents, TO = tour operators)
distribution_channel	Booking distribution channel
is_repeated_guest	Repeated guest (1) or not (0)
previous_cancellations	Number of previous cancellations
previous_bookings_not_canceled	Number of previous bookings not cancelled
reserved_room_type	Code of room type reserved
assigned_room_type	Code of assigned room
booking_changes	Number of changes to the booking
deposit_type	Type of deposit made
agent	ID of travel agent
company	ID of the company that made the booking
days_in_waiting_list	Days elapsed between booking and confirmation
customer_type	Type of booking (group or not group)
adr	Average daily rate
required_car_parking_spaces	Number of parking spaces required
total_of_special_requests	Number of special requests
reservation_status	Last status of the reservation
reservation_status_date	Date at which the last status was set



Connect Test and Score to the Data Sampler widget. Test and Score is the primary model evaluation widget, which performs cross-validation and reports the scores.



Test and Score widget needs two things: the data we just provided and the learner (s) to train the model. Let us use a few popular classifiers: logistic regression, naïve Bayes, and random forest.



Test and Score used fivefold cross-validation to compute model scores. Looking at the scores, the best performing model is the random forest, as its AUC score is very high at 0.96.

**Test and Score**

**Sampling**

- Cross validation
  - Number of folds: 5
  - Stratified
  - Cross validation by feature
- Random sampling
  - Repeat train/test: 10
  - Training set size: 66 %
  - Stratified
  - Leave one out
  - Test on train data
  - Test on test data

**Target Class**

(Average over classes)

**Model Comparison**

Area under ROC curve

Negligible difference: 0.1

**Evaluation Results**

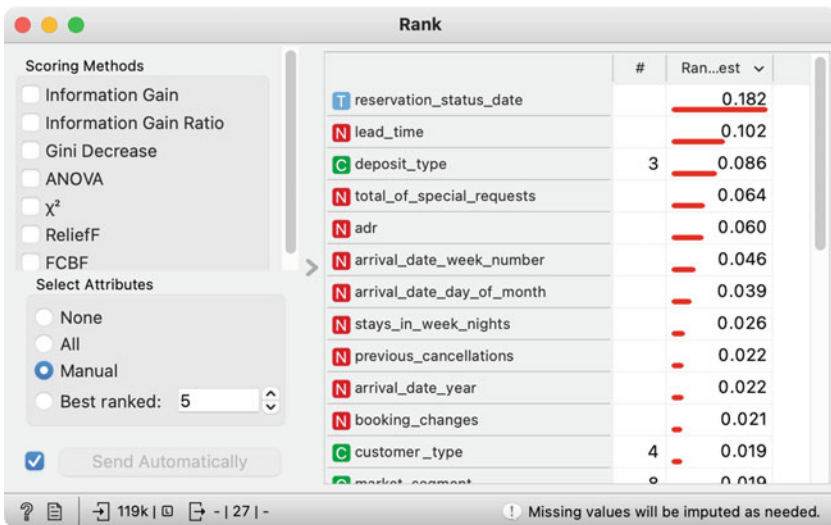
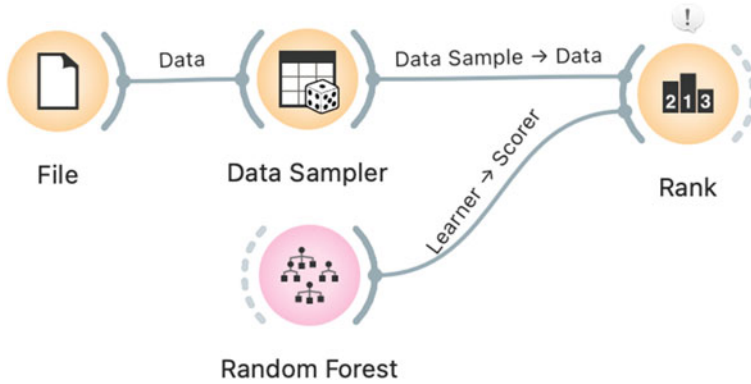
Model	AUC	CA	F1	Precision	Recall
Random Forest	0.959	0.916	0.915	0.918	0.916
Naive Bayes	0.816	0.762	0.755	0.758	0.762
Logistic Regression	0.594	0.628	0.485	0.395	0.628

**Model Comparison by AUC**

	Random Forest	Naive Bayes	Logistic Regres...
Random Forest		1.000	1.000
Naive Bayes	0.000		1.000
Logistic Regression	0.000	0.000	

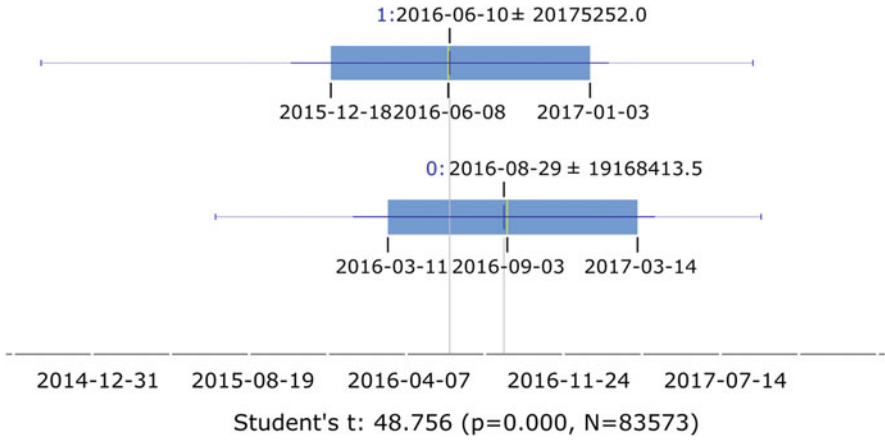
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.

Unlike logistic regression and naïve Bayes, which can be inspected with nomogram (Možina et al., 2004), random forest is more difficult to interpret. To understand the reasoning behind random forest, we will use feature ranking to see which variables are informative in the context of this model.



The date when the reservation was made is very important for distinguishing between bookings that were cancelled and those that were not. Looking at the box plot, split by cancellations, we see that bookings that were not cancelled (0) were made at a later date than those that were (Fig. 9).

However, for predicting future instances, knowing that there were more cancellations before 2017 will not help us much. Thus, let us remove the *reservation\_status\_date* variable and re-run cross-validation.



**Fig. 9** Box plot of reservation\_status\_date, split to cancelled (0) and not cancelled (1) groups. Comparing their means, it looks like there were more cancellations before mid-2016 than after

Model	AUC	CA	F1	Precision	Recall
Random Forest	0.917	0.853	0.851	0.853	0.853
Naive Bayes	0.816	0.757	0.752	0.753	0.757
Logistic Regression	0.862	0.811	0.804	0.813	0.811

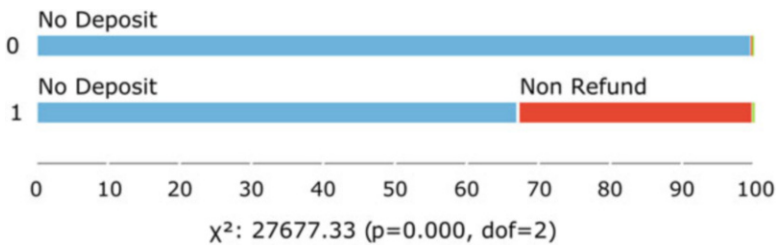
Model accuracy remains high, with an AUC score of 0.917 for the random forest model. A quick look at the feature ranking shows that the order of the important variables remained the same after removing the reservation date. Lead time and deposit type are the most important variables for the random forest, but, unfortunately, we cannot conclude the relationship between their values and the target value. In other words, it is difficult to determine whether a long lead time and a certain type of deposit are more or less likely to result in a cancellation.

	#	Ran...est
<b>N</b> lead_time		<u>0.120</u>
<b>C</b> deposit_type	3	<u>0.102</u>
<b>N</b> adr		<u>0.086</u>
<b>N</b> total_of_special_requests		<u>0.061</u>
<b>N</b> arrival_date_day_of_month		<u>0.058</u>
<b>N</b> arrival_date_week_number		<u>0.047</u>

Finally, we must evaluate the model on unseen, test data. This will tell us how well the model generalizes and, consequently, how useful it is when it comes to practical application. We will send the remaining 30% of the data to Test and Score and use the *Test on test data* option. The model still reports high accuracies, namely, 0.918 AUC and 0.855 recall, meaning it will be able to identify, with a high degree of certainty, which bookings are likely to result in a cancellation.

Model	AUC	CA	F1	Precision	Recall
Random Forest	0.921	0.859	0.857	0.858	0.859

Let us take another look at the box plot to see how *lead\_time*, *deposit\_type*, and *adr* relate to cancellations.



Longer lead time, high ADR, and a high number of previous cancellations all make sense. But what about deposit type? It is counter-intuitive to have more cancellations with non-refundable booking than the other way around. To uncover the mystery, a more profound knowledge of the tourism industry is required. As Antonio et al. (2019b) argue, this is due to certain bookings that are made through OTA using false or invalid credit card details issued for visa requests. This not only shows how vital domain knowledge is but also how necessary it is to understand the data set.

**Service Section**

**Main Application Fields:** Testing and evaluating classification and regression predictive models.

**Limitations and Pitfalls:** Model evaluation must always be performed on separate training and test data sets. Otherwise, it is easy to overfit the model. Such a model would generalize poorly and would not work on new data. Even despite training the model on the training data and testing on the test data, it is still easy to overfit, for example with feature selection. If one selects informative features before training the model, this is yet another type of overfitting. Finally, one can overfit by tweaking the model parameters without validating the final model on another validation data set.

(continued)

**Similar Methods and Methods to Combine with:** Model evaluation typically follows any kind of model training, whether it be classification (logistic regression, naïve Bayes, random forest, etc.) or regression (linear regression, SVM, random forest, etc.). Model evaluation can be domain-specific in which different domains use different model evaluation scores. For text mining, for example, Cohen’s Kappa (Cohen, 1960) is frequently used to determine inter-rater reliability for manually labelled data. For time-series analysis, on the other hand, model evaluation is slightly different. The two typical learners are VAR (vector autoregression) and ARIMA (autoregressive integrated moving average), which would be evaluated as regression models with some additional scores (e.g. a prediction of change in the direction of the series).

**Code:** The Orange workflow is available at: <https://github.com/DataScience-in-Tourism/Chapter-13-Model-Evaluation-Overfitting>

## Further Readings and Other Sources

- Flach, P. A., & Kull, M. (2015). Precision-recall-gain curves: PR analysis done right. *Advances in Neural Information Processing Systems*, 838–846. Available at: <https://papers.nips.cc/paper/2015/file/33e8075e9970de0cfea955afd4644bb2-Paper.pdf>
- Hastie, T., Tibshirani, R., & Friedman, J. (2017). *Elements of statistical learning*. Chap. 7, pp. 219–260. Available at: [https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII\\_print12\\_toc.pdf](https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12_toc.pdf).
- Orange video tutorials.: <https://www.youtube.com/c/OrangeDataMining>

## References

- Antonio, N., de Almeida, A., & Nunes, L. (2019a). Hotel booking demand datasets. *Data in Brief*, 22, 41–49. <https://doi.org/10.1016/j.dib.2018.11.126>
- Antonio, N., de Almeida, A., & Nunes, L. (2019b). Big data in hotel revenue management: Exploring cancellation drivers to gain insights into booking cancellation behavior. *Cornell Hospitality Quarterly*, 60(4), 298–319. <https://doi.org/10.1177/1938965519851466>
- Breunig, M. M., Kriegel, H.-P., & Ng, R. (2000). LOF: Identifying density-based local outliers. *Proceedings of ACM SIGMOD Record*, 29(2), 93–104. <https://doi.org/10.1145/335191.335388>
- Carreño, A., Inza, I., & Lozano, J. A. (2020). Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artificial Intelligence Review*, 53, 3575–3594. <https://link.springer.com/article/10.1007/s10462-019-09771-y>
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46. <https://doi.org/10.1177/001316446002000104>
- Demšar, J., Curk, T., Erjavec, A., et al. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14(Aug), 2349–2353. <http://jmlr.org/papers/v14/demсар13a.html>



- Drummond, C., & Holte, R. C. (2006). Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65, 95–130. <https://link.springer.com/article/10.1007/s10994-006-8199-5>
- Elkan, C. (2001). The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 2001*, 973–978. <http://web.cs.iastate.edu/~hnavar/elkan.pdf>
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. <https://doi.org/10.1016/j.patrec.2005.10.010>
- Hernandez-Orallo, J., Flach, P., & Ferri, C. (2012). A unified view of performance metrics: Translating threshold choice into expected classification loss. *Journal of Machine Learning Research*, 13, 2813–2869. <https://www.jmlr.org/papers/volume13/hernandez-orallo12a/hernandez-orallo12a.pdf>
- Kawaguchi, K., Pack Kaelbling, L., & Bengio, Y. (2017). *Generalization in deep learning*. arXiv preprint. <https://arxiv.org/abs/1710.05468>
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *Eighth IEEE International Conference on Data Mining 2008*. <https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf>
- Možina, M., Demšar, J., Kattan, M., & Zupan, B. (2004). Nomograms for visualization of naive Bayesian classifier. In *European conference on principles of data mining and knowledge discovery* (pp. 337–348). Springer. [https://doi.org/10.1007/978-3-540-30116-5\\_32](https://doi.org/10.1007/978-3-540-30116-5_32)
- Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers* (pp. 61–74). MIT Press.
- Precision and recall. (n.d.). In *Wikipedia*. Retrieved 5 February, 2021, from [https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall).
- Vapnik, V. (2000). The nature of statistical learning theory. *Springer Science & Business Media*. <https://doi.org/10.1007/978-1-4757-3264-1>

# Interpretability of Machine Learning Models



## How Can One Explain Machine Learning Models?

Urszula Czerwinska

### Learning Objectives

- Understand why model interpretation is useful
- Demonstrate how to explain a machine learning model
- Learn to ask pertinent questions about the model

## 1 Introduction and Theoretical Foundations

### 1.1 Introduction to Explainability

Having functional data pipelines and powerful models is not sufficient anymore. Moreover, society in general is impacted by AI decisions, and more questions are being raised in regard to how algorithms work and what their predictions are based on. In practice, one is often faced with the problem: how can one convince the executing committee to change a marketing strategy if the only thing that can be argued is the algorithm's prediction? Different inquiries may concern the robustness and the reliability of the algorithm; how can it be assumed to be unbiased? Others may ask: how can I trust the results without understanding its logic?

Some essential terms and short definitions employed throughout this chapter must be introduced first. To start with, *explainability* is the possibility of explaining the prediction of an algorithm from a technical point of view. *Interpretability*, per contra, can be defined as the ability to explain or provide meaning in terms that

---

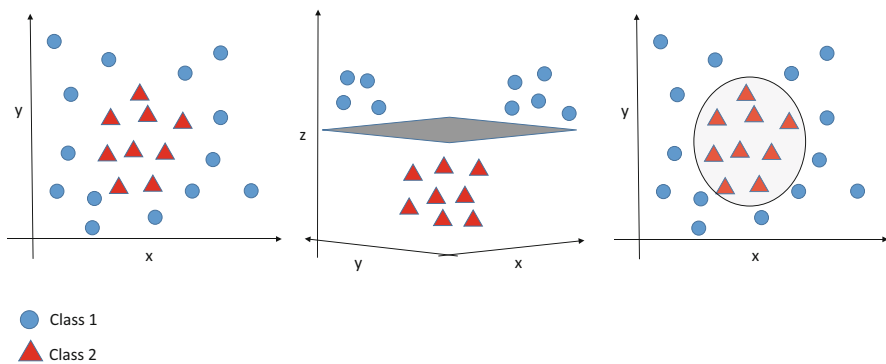
U. Czerwinska (✉)  
Paris, France  
e-mail: [urszula.czerwinska@cri-paris.org](mailto:urszula.czerwinska@cri-paris.org)

are simple and understandable to a general audience. Finally, *transparency* is a property of a model that is self-explanatory on its own.

In this chapter, the explanation of a machine learning model can be divided into two types: local and global. *Local explanations* refer to explanations of a particular prediction. For instance, local explanations answer the question: why has the model predicted this particular value for this customer or entry? *Global explanations*, on the other hand, explain the general decision-making of the model. The main attributes that can be obtained in the case of using global explanations are considered the important features for the model. Moreover, the ensemble of the general conclusions of the model result in the internal rules.

## 1.2 Why Are Some Models Uninterpretable?

While linear dependence between factors is understandable to the human mind, more complex nonlinear data transformations may pose a challenge. Some models, such as Supported Vector Machines (SVM) (Vapnik & Golowich, 1997), became very popular in the 1990s. In the case of binary classification, this algorithm, in its nonlinear version, projects the data in a nonlinear space by searching for the optimal transformation to easily separate points using a plane (Fig. 1). In the original space, the data is not linearly separable (left image in Fig. 1); thus, the transformations are applied to the original feature space (middle image). For instance, an age variable could be scaled and translated into a different scale, including negative values (i.e.,  $-3$  to  $5$ ). Therefore, the transformations make the transformed features inconceivable and hard to understand. However, they are necessary for the model because they enable the separation of the two classes (right image). Most complex models, such as Neural Networks, perform a vast number of mathematical transformations, making them hard to trace back and even harder to interpret.

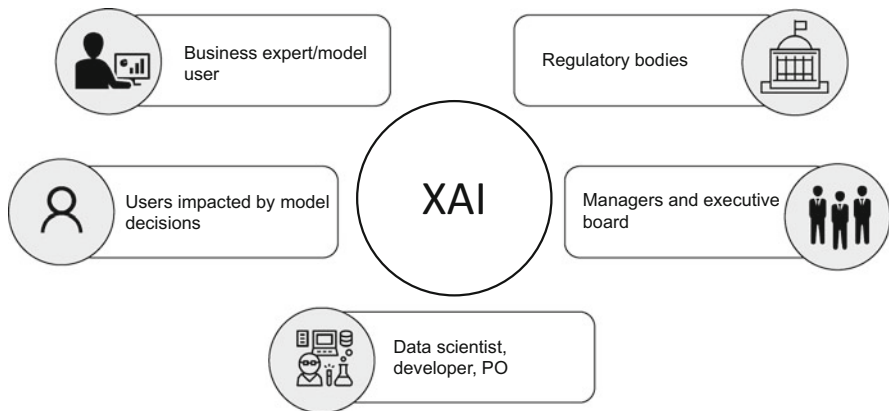


**Fig. 1** The SVM algorithm projects data into a space where the problem is separable and can make predictions in that space. Source: Author's illustration

### 1.3 For Whom Can Explainability Be Useful or Necessary?

Various profiles interested in explainability or interpretability have been identified in Fig. 2 (Barredo Arrieta et al., 2019). Firstly, a business expert (a model user) can use explainability to increase his trust in the model by better understanding the causality of the prediction. Regulatory agencies require model explanations in order to certify compliance with the legislation, or they use explainability to inspect the accuracy of the model and lack of bias. In addition, managers and executive boards use interpretability software to assess regulatory compliance and understand enterprise AI applications. Users impacted by model decisions may also request access to the model explanations so as to understand their situation or to question model decisions. Finally, data scientists, developers, and project owners use explainability tools to improve product performance, identify new features, or explain predictions to their superiors.

To make explainability accessible to people with moderate technical skills, data scientists and developers should be comfortable with explainability tools. As explainability methods and software programs continue to undergo active research, an understanding of numerous interdisciplinary concepts is required (Adadi & Berrada, 2018). It also demands and calls for substantial developmental work in order to adapt these tools to real-life projects (Sokol & Flach, 2020). In many cases, the skill of interpreting models remains more an art than a craft (Gilpin et al., 2018; Hall, 2018).



**Fig. 2** Different profiles are directly concerned with a models’ explainability. Source: Barredo Arrieta et al. (2019)

## ***1.4 Why Should One Care About the Interpretability of ML Systems?***

### **1.4.1 Providing Trust**

First, interpretability certifies the decision's impartiality, which means that no unwanted bias is playing a role in the algorithmic decisions (Doshi-Velez & Kim, 2017). It also protects the system from being "hacked" by manipulating input values (Tao et al., 2018) and provides understanding and verification regarding if valid data plays an essential role in the computed outcome (Kaufman et al., 2011). An example of data leakage, meaning that some features in a training set provide a direct proxy for forecasting, can be used to illustrate explainability. One could imagine that a rental bike company needs to predict the number of rental bikes necessary at every time of the year. The training data consists of two types of bikes. For training, one excludes the total number of bike rentals from the historical data but forgot to delete the number of the different types of bike rentals. The trained model performance is outstanding. However, one can realize, through an explainability approach, that the prediction is based solely on the number of the two types of bikes. It then becomes obvious that data leakage was introduced, and the model will not be able to function in real-life conditions. For complex datasets with thousands of variables, it can be challenging to find the source of the leakage; however, thanks to the model explainability, this can be identified more efficiently.

### **1.4.2 Complying to Regulations**

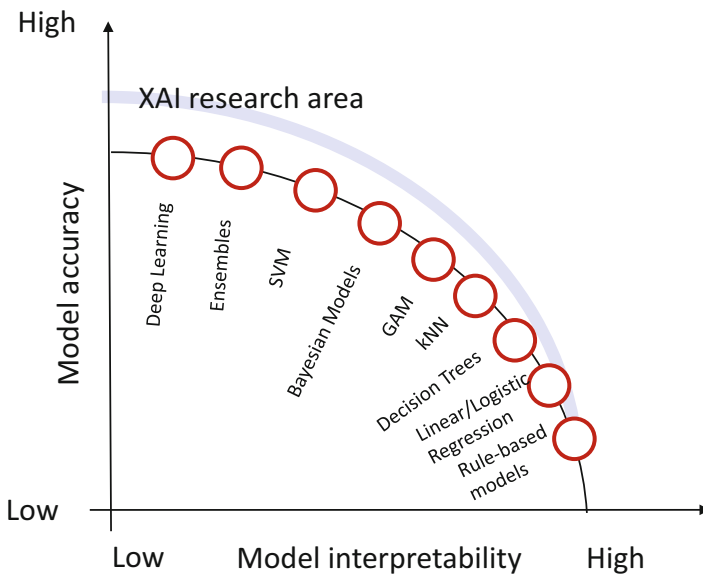
With the growing impact of AI-based systems, regulatory institutions put more constraints on how these technologies can be used. A European Commission White Paper on AI expresses that "the opacity of systems based on algorithms could be addressed through transparency requirements" (European Commission, 2020, p.15) Besides, the General Data Protection Regulation (GDPR) law mentions a "right to explanation" without specifying the application conditions. Therefore, it is not clear how and in which cases an explanation can be demanded and which types of companies or institutions are obliged to provide one (European Commission, 2020). In the USA, the "right to explanation" is part of an ongoing debate. The Federal Trade Commission's Bureau of Consumer Protection studies how consumers use algorithms by researching algorithmic transparency and by funding external research efforts on the topic (Noyes, 2015). Moreover, authorities may request proof that the algorithm was trained on unbiased data. In the domains of recruitment and banking, specific laws, such as the US Civil Rights Act, Equal Credit Opportunity Act, Fair Housing Act (US), or the aforementioned GDPR, Article 22 (EU), have been formulated to control AI algorithms and prohibit discrimination and bias (Barocas & Selbst, 2018; O'Neil, 2017). In some cases, an ML approach should not be preferred because of lack of clear interpretability and, therefore, potential risk of

an increase in biases. In that situation, a rule-based deterministic system would be advisable, even at the cost of the performance (Rudin, 2019).

### 1.4.3 Understanding Predictions

In most cases, when very restrictive regulations do not apply, one would put innovation and performance as the priority. Best-case scenario, a satisfying trade-off between accuracy and explainability can be found. The relationship between model interpretability and accuracy for different ML strategies can be depicted as seen in Fig. 3. To find the best way to approach equilibrium between performance and explainability, a new field has emerged in which Explainable Artificial Intelligence (XAI) aims to produce robust and explainable algorithms. In general, the best performing and the least complex model possible is preferred.

An interpretable model can provide an understanding of why the model predicted the given outcome. More specifically, local explanations can answer questions about a particular prediction. For instance, a rental agency may use an ML model to put a price on their available rentals based on the rental characteristics, the actual demand, and the time of the year. The model could predict a very high price for a quite unpopular rental. The renting company (the user of the model) would like to obtain a “logical” explanation for this event. Having modeled local explanations would allow for scrutinization of the “logic” of the model. Based on the explanations, it could



**Fig. 3** The relationship between interpretability and accuracy of a model. Source: Barredo Arrieta et al. (2019)

then be decided if the predicted price is justified by the data or if it is just a technical artifact.

#### 1.4.4 Creating Better Models

Interpretability is an essential concept in AI and ML research as it ensures if the model works correctly. It is difficult for a very complex yet well-performing model to identify bias and make sure that the model truly predicts what it was intended to. Ribeiro et al. (2016), authors of one of the explainability frameworks, show a striking example in their scientific publication in which an image classifier was trained to differentiate between a husky and a wolf. In one of their figures, it is clearly shown, thanks to their explainability method, that a husky dog is classified as a wolf because of the snow in the background and not because of the actual characteristics of the animal. This striking example is a warning that even performant machine learning models, marked with high accuracy, might be fallacious. For research purposes, both global and local explanations are valuable sources of information, and researchers can use explainability to improve their models. The model weak points, observed through particular examples and general model behavior, jeopardize the whole system. However, owing to explainability, they can be dealt with and the overall model quality can be improved.

### 1.5 Explainability Frameworks

Recently, explainability has emerged as a real demand for ML practitioners and users. Different strategies have been developed; some of which dive deep into the models' mechanics, while some use other strategies to mock models' behavior and, in turn, explain the predictions.

#### 1.5.1 Model Agnostic Strategy

One popular strategy is to train a *surrogate*, or *shadow model*. This additional model explains the predictive model, as proposed by Bastani et al. (2017) and TeamHG-Memex (2016). A shadow model can approximate or simulate the true model's decisions with a simple framework based on the training data. According to Ribeiro et al. (2016), application of the surrogate model can be mathematically valid when used locally.

Another strategy is called the *perturbation strategy*. It was introduced by Breiman (2001) for random forest and extended by Fisher et al. (2019). In the scope of perturbation strategies, changes (a.k.a. perturbations) are introduced to inputs. They are simulated according to mathematical standards, and their impact on the output is recorded and evaluated. This process is iterated until a local model

can be built based on the created examples. Lundberg et al. (2017) proposed a perturbation strategy based on game theory, guaranteeing mathematically valid perturbations that are uniformly distributed among feature values of the instance. They proposed several perturbation approaches, of which one is truly model agnostic, and developed more time-efficient explanations of when the structure of the model can be employed (Lundberg et al., 2020). Both the perturbation and surrogate strategies are popular techniques as they do not require access to the model structure. In theory, they can be used with available service models, for instance, through an API.

Finally, the strategy of *example-based explanations* is considered very “human-friendly” and straightforward as it explains a data instance using contrastive examples (Aamodt & Plaza, 1994). In opposition to the strategies discussed above, here, specific data examples are used to explain the model. This approach is popular for text and image data explanations since these examples can speak for themselves.

The details of the selected frameworks will be discussed subsequently, coded in the preferred programming language choice for data science practitioners, Python (Gregory Piatetsky, 2019). Please note that the described software may find equivalents in other programming languages or be embedded in services proposed by cloud platforms.

### 1.5.2 LIME

Local Interpretable Model-agnostic Explanation (LIME), introduced by Ribeiro et al. (2016), can be used to interpret most of the existing models with tabular, image, or text data, using slightly different algorithms for each data type. The general concept remains the same independent of the data format. LIME takes the predictive model and the prediction for which the explanation is wanted as inputs. The LIME framework produces local perturbations of a feature, or simulations of true instances. Then, the LIME algorithm uses the predictive model to predict the outcome based on the simulated data input. It performs those perturbations numerous times to create a local landscape of predictions around the selected instance. Subsequently, a second, simple model (the surrogate model) is fitted on the simulated examples of the local landscape. This surrogate model is naturally interpretable and is used to explain the original model prediction. Nonetheless, it can only be trusted in a very local environment. The linear model answers how a change in a specific feature impacts the outcome. In the case of predicting rental values, as described above, it would indicate which vacancy characteristic contributes the most to the predicted price.

As mentioned, this approach works with practically any model involving tabular data, text, and images. The only difference is in the perturbation strategy, which must be adapted to the datatype. The universality of the LIME framework is what greatly contributes to its popularity. Nevertheless, LIME is not free of its drawbacks. It has been criticized in the research community because it suffers from labels and data shift. Explanations may depend on the choice of hyperparameters (lime algorithm parameters used in the advanced model), resulting in similar points obtaining



different explanations. LIME framework's biggest challenge is to define what "similar" means (Laugel et al., 2018). The hyperparameter setting that defines the neighborhood of interest should be chosen carefully for tabular data. If any features are correlated, the LIME results may be incorrect. As the perturbations depend on the sampling procedure, the results may change between the two executions of the algorithm.

### 1.5.3 ELI5

ELI5 (TeamHG-Memex, 2016) is another popular library that uses LIME for local explanations and Mean Decrease Accuracy (MDA) for global explanations (also called Permutation Feature Importance). The main idea of MDA is to replace each feature with a random set of numbers and measure the impact on the model performance. As this approach is model agnostic, it can be used with simple and complex models alike. In contrary to LIME, it requires access to the true labels of data instances. Consequently, it cannot be used for non-labeled or fictive data instances. Besides, it shares LIME's weakness of being sensitive to random simulations. Another technical downside is that the Python implementation is limited to working with the sci-kit learn library (Pedregosa et al., 2011) for the global explanations. The ELI5 library provides off-the-shelf powerful visualization tools that can be useful for even unexperienced data analysts and data scientists. This is undoubtedly one of the strongest advantages of this library. It can be seen as an additional layer for the LIME framework or as a stand-alone tool for a quick diagnostic of the predictive algorithm.

### 1.5.4 Anchors

An extension of local surrogate models, such as LIME, was proposed by Ribeiro et al. (2018a, 2018b). Anchors aim to solve one of LIME's biggest problems, the neighborhood definition. They explain predictions for an instance of any classification model by finding a region in which the prediction does not change, or in which it is "anchored". This solution is model agnostic as it only needs data and the predictive model function. The approach deploys reinforcement learning techniques instead of fitting surrogate models, making it more efficient and less prone to underfitting.

Anchors are very user-friendly as they formalize the regions in short "if, then" statements. For example: "The house is worth more than 500k USD if the number of bedrooms is superior to 3 and has a garden". These statements are completed based on precision and coverage information where coverage is the percentage of perturbations to which the rule applies, and precision is the percentage of the rule's accuracy. Therefore, anchors answer the question of which feature values are essentially responsible for the predicted outcome. Questions such as why client A

received suggested accommodation X or which features of monument B were decided to be placed in the “must-see place of city Z” category can be asked.

Anchors can sometimes be too specific and, therefore, only apply to a limited number of cases. To avoid this problem, discretization is employed, and values are grouped by an interval. However, the user needs to pick the right discretization parameters. The application of anchors to images is questionable because it is unclear how to define coverage (i.e., the superpixel of one image does not apply directly to another). Yet, on the practical side, the authors of anchors provided their implementation as an open-source code (Ribeiro et al., 2018a, 2018b), applicable to tabular and text data. A popular XAI library called “Alibi” (Klaise et al., 2019) proposed a slightly changed implementation of their solution called Scoped Rules (Anchors). This software extends the algorithm’s use to image data and provides a unified interface for many other explicability algorithms.

### 1.5.5 Counterfactuals

Another user-friendly approach, which asks the exact opposite question than the aforementioned anchors, are counterfactuals. Counterfactuals reflect the strategy of example-based explanations. They can answer the question: “Which feature values should I modify to change the prediction of the model?”, and their use can be justified in multiple scenarios. For instance, a holiday rental is worth 30 euros per night; how should the features be optimized in order to be able to rent it for 50 euros per night? Several features cannot be changed easily, such as square meter area, room number, or city area, but the decor, check-in hours, electronic equipment, or heat isolation could be optimized to reach the desired price. To compute the counterfactuals of a model, the predictive function and a data instance are necessary. Therefore, they are simple to apply even if the model’s exact calculations are not accessible or the data is sensitive. Usually, for one instance, several counterfactuals will be computed; yet, there is no guarantee that they will turn out to be practical or convenient.

The first implementation of counterfactuals was proposed by Wachter et al. (2017). Their aim was to minimize the loss between the desired outcome and randomly drawn data instances. However, this approach does not work well with categorical features with too many categories, and the unrealistic feature combinations are not penalized. This algorithm is a part of the Alibi Python library, mentioned in the previous section. It can be applied to tabular data and image classification. Another popular implementation by Dandl et al. (2020) takes advantage of the NSGA-II algorithm (Deb et al., 2002), optimizing the quality of four criteria simultaneously, as described by Molnar (2019). These criteria can be formulated as follows. First, the counterfactual falls into the desired prediction scenario. Second, the generated counterfactual should be as close to the original data instance as possible. Third, it generates multiple variable counterfactuals (possible paths to reach the researched objective), and, lastly, the values of the features should be realistic. Apart from this, the Alibi library proposed yet a different implementation

based on the publication by Van Looveren and Klaise (2019), claiming to be an efficient way to compute counterfactuals.

### 1.5.6 SHAP

One of the most popular explainability frameworks recognized in the scientific community is Shapley Additive exPlanations (SHAP) by Lundberg et al. (2017, 2020). In short, Shapley's values calculate the importance of a feature by comparing what a model predicts with and without this specific feature. However, since the order in which a model "sees" the features can affect its predictions, this is done in all possible ways so that the features are compared fairly. This approach is inspired by game theory and has solid mathematical foundations. The most used implementation of Shapley's values, the Python SHAP library, has different SHAP algorithms for different predictive algorithm types. For example, *LinearExplainer* for linear models and *TreeExplainer* for tree-based models (XGBoost, LightGBM, CatBoost, RandomForest), while *DeepExplainer* and *GradientExplainer* are optimized for neural networks. There is also a model agnostic model (KernelSHAP), but it is not the most efficient one.

To see how the calculations behind the library are achieved in detail, resources, such as the SHAP documentation and publications by the authors of the library (Lundberg & Lee, 2018; Lundberg et al., 2017, 2020), are available online. There are also numerous publications from the data science community explaining SHAP details and applying them to different scenarios (see *Additional resources*).

### 1.5.7 Deep Learning

Libraries dedicated to explaining solely artificial neural network models also exist. One of the most well-known explanatory models is DeepLift (Shrikumar et al., 2017), but there are many other methods discussed in the literature. Arras et al. (2017) proposed a method called Layer-wise Relevance Propagation (LRP) showing promising results for text classification, while Olah et al. (2017) proposed a method for artificial neural network image processing. The latter technique shows the visualization of neural network layer weights within the context of the original image, helping to understand which part of the image has the most dominant influence. On the other hand, Tan et al. (2018) use Generative Adversary Networks (GAN) to train a simpler white-box model with a performance similar to the performant complex model. This application is inspired by a general model distillation idea aiming to create simpler, more compact models with high performance. A selection of libraries compatible with programming frameworks (e.g., TensorFlow or PyTorch) have also been released. For instance, Captum (Kokhlikyan et al., 2020) groups algorithms related to the PyTorch framework. Likewise, the `tf_explain` (Meudec, 2020) interface groups the ones relevant for TensorFlow and Keras models.

### 1.5.8 Cloud Platforms

Some cloud platforms propose integrated interpretability algorithms, usually in a user-friendly way, optimized and accompanied by visualizations. Several ones can be named: Dataiku,<sup>1</sup> which proposes all-in-one ML platforms with a specified module facilitating the use of XAI frameworks such as SHAP. Azure Machine Learning<sup>2</sup> proposes a Python library pre-installed in the Azure ML working environment, unifying the interfaces of several XAI methods discussed in this chapter. Google Cloud Explainable AI<sup>3</sup> proposes original methods to explain and evaluate model robustness, while H<sub>2</sub>O, a popular autonomous AI platform, developed their own set of tools called MLI.<sup>4</sup> IBM Watson Explainable AI<sup>5</sup> proposes a module dedicated to the XAI as well. Some companies further propose explainability as a service. Among those, DataRobot<sup>6</sup> has a plethora of “ready-to-use” ML models and guarantees a possibility to explain them, while another company, Craft-ai,<sup>7</sup> automates the process of data preparation and ML model training. It proposes only explicable models and explains them along with a prediction. The explainability-as-a-service platforms require a minimal understanding of machine learning but no coding skills.

## 1.6 Fairness and Adversarial Attacks

Bias detection, security, and explicability are separate topics, but they remain linked. Model explainability can be used for bias detection, and the vulnerabilities of the model can be used to hack it. Explainability exposes those weakest points; thus, many toolboxes aiming to “audit” models are considered model explicability frameworks as well.

One Python toolbox, FairML, introduced by Julius (2016), assembles models that investigate the fairness of an ML model. The perturbation strategy inspects the collinearity issues in the dataset it verifies if one feature does not dominate decisions made by the model. Another framework focused on auditing models is BlackBoxAuditing (Adler et al., 2016a, 2016b; Feldman et al., 2015). This Python library detects the indirect influence of features and the relationship between the features in the dataset. It is model agnostic and does not need to access the model or retrain it.

---

<sup>1</sup><https://blog.dataiku.com/white-box-vs-black-box-models-balancing-interpretability-and-accuracy>

<sup>2</sup><https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability>

<sup>3</sup><https://cloud.google.com/explainable-ai>

<sup>4</sup><http://docs.h2o.ai/driverless-ai/latest-stable/docs/userguide/interpreting.html>

<sup>5</sup><https://www.ibm.com/watson/explainable-ai>

<sup>6</sup><https://www.datarobot.com/platform/trusted-ai/>

<sup>7</sup><https://www.craft.ai/>

Sokol et al. (2019) published a FAT-fairness toolbox incorporating a wide plethora of algorithms that help to evaluate fairness, accountability, and transparency using one common interface. This is a remarkable effort provided to the community in order to make research easier to apply to a production-like environment. Moreover, AI Fairness 360 toolbox (Bellamy et al., 2018) is dedicated solely to evaluating the fairness of an algorithm by grouping many previously published algorithms. Many similar frameworks, such as FairTest (Tramer et al., 2015) or FairLearn (Agarwal et al., 2018), have also been published in recent years.

Finally, in regard to examining the security of high-interest ML models in fields closely related to explainability, Nicolae et al. (2018) introduced the Adversarial Robustness Toolbox (ART), which “enable[s] developers and researchers to evaluate, defend, certify and verify Machine Learning models and applications against the adversarial threats of Evasion, Poisoning, Extraction, and Inference”.<sup>8</sup>With explainability techniques, it is possible to better understand the model’s manner of working and, therefore, to create the best adversarial attack techniques (Gilmer et al., 2018). Along with the attack techniques, defense techniques are often proposed (Tramèr et al., 2020), and thanks to the research in this field, one can better protect the model against malicious usage. The adversarial attacks can also target the explainability frameworks themselves, as demonstrated by Ghorbani et al. (2019) and Slack et al. (2020).

## 2 Practical Demonstration

Numerous applications of explainability frameworks can be used. Thus, in this hands-on-section, a publicly accessible dataset of booking records (“Hotel Booking Demand | Kaggle”, 2020), originally published in Antonio et al. (2019), will be used to build a model that predicts booking cancellations. In this chapter, we will focus on the added value of explicability frameworks.

### Outline

1. Data description
2. Data preparation
3. Classification model
4. Explicability
  - (a) SHAP—global interpretation
  - (b) SHAP—local interpretation
  - (c) LIME
5. Conclusions

---

<sup>8</sup><https://adversarial-robustness-toolbox.readthedocs.io/en/stable/>

### 2.1 Data Description

In the downloaded dataset, there are 32 columns and 119,390 rows. A full data dictionary and the cleaning process done on the original dataset can be found on Mock and Bichat’s (2020) GitHub page. The dataset contains both categorical and continuous data as well as text fields. It was processed so as to facilitate the ML task described in the following section.

### 2.2 Data Preparation

The dataset was cleaned, in which no entries containing missing values were kept, resulting in 118,902 entries (Fig. 4). Only 24 columns having a significant correlation with the target variable “is\_canceled” were kept (Table 1). The categorical variables: “meal”, “market\_segment”, “distribution\_channel”, “reserved\_room\_type”, “assigned\_room\_type”, and “customer\_type” were one-hot encoded for the ML model. The string variables or categorical variables, such as “country” or “hotel”, were coded into numbers, and new variables were created, such as “is\_family”, which expresses if adults were traveling with children. Most of the feature names are self-explanatory. There is information about booking type, date, length, customer type, and further information about customer previous bookings, for instance, “lead\_time” (the time between booking and arrival) and “adr” (Average Daily Rate: dividing the sum of all lodging transactions by the total number of

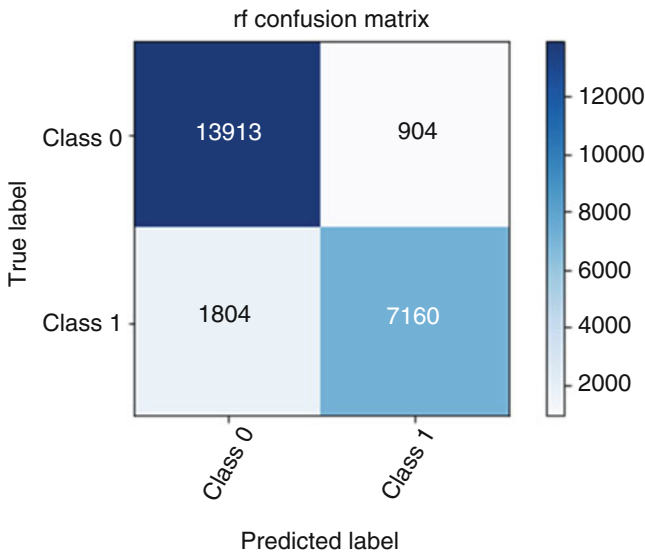


Fig. 4 Confusion matrix. Source: Author’s illustration generated with Python code

**Table 1** List of columns in the processed dataset

Column name	Python data type
hotel	int64
lead_time	int64
arrival_date_year	int64
arrival_date_day_of_month	int64
stays_in_week_nights	int64
country	int64
is_repeated_guest	int64
previous_cancellations	int64
previous_bookings_not_canceled	int64
booking_changes	int64
days_in_waiting_list	int64
adr	float64
required_car_parking_spaces	int64
total_of_special_requests	int64
is_family	int64
total_customer	float64
deposit_given	int64
total_nights	int64
assigned_room_type	object
reserved_room_type	object
customer_type	object
distribution_channel	object
market_segment	object
meal	object
<b>is_cancelled</b>	<b>int64</b>

staying nights). In total, there were 79,306 canceled and 39,596 non-canceled entries.

### 2.3 Classification Model

Data were then split into training (80%) and testing (20%) sets. The target variable “is\_canceled” was excluded from the data and saved as a y variable. A “RandomForestClassifier()” from the sci-kit learn library was trained with default parameters and saved in the “rf\_model” variable. Class 0 was defined as “not canceled” and Class 1 as “canceled” (Tables 2 and 3).

The trained model “rf\_model” has a satisfying accuracy of 0.89 and an F1 score of 0.84. Moreover, 904 instances were wrongly classified as class 1, and 1804 instances were wrongly classified as class 0 (Fig. 4). As learned from other chapters, such as the Hyperparameter-Tuning and Supervised ML models, training models can be much more complicated than demonstrated here, and they can always be

**Table 2** Confusion matrix without normalization

	precision	recall	f1-score	support
Not cancelled	0.89	0.94	0.91	14817
Cancelled	0.89	0.80	0.84	8964
Accuracy			<b>0.89</b>	23781
Macro avg	0.89	0.87	0.88	23781
Weighted avg	0.89	0.89	0.88	23781

**Table 3** Model performance

Cohen Kappa score:	0.75
ROC AUC score:	0.95
Default recall score:	0.80
<b>f1:</b>	<b>0.84</b>

optimized. As the focus of this chapter concentrates on explainability, a simple model has been used.

## 2.4 Explicability

### 2.4.1 SHAP: Global Interpretation

First, the pre-installed SHAP library is imported using the “import” statement.

```
import shap
```

SHAP uses a part of the dataset called “background” to define reference points for the graphics. Only 1% of the training set (952 entries) will be selected for the background set, preserving the proportion of canceled and non-canceled entries. Selected data will be saved in the “x\_background” variable.

Subsequently, the explainer is built with the command “shap.TreeExplainer()”. The trained model named “rf\_model” and the “x\_background” data are the inputs, and the parameter “model\_output” is set to “probability”. The units of the predicted values will be expressed as a probability ranging from 0 to 1. This choice enables the comparison of different models as the probability brings values to a common scale. The other parameter, “feature\_perturbation”, is set to “interventional” which signifies that the explainer model is computed using the background data (a representative sample of the training set).

The Shapley values are then computed with a method known as “explainer.shap\_values”.

```
explainer = shap.TreeExplainer(rf_model,x_background,
model_output="probability", feature_perturbation="interventional")
shap_values = explainer.shap_values(x_background)
```



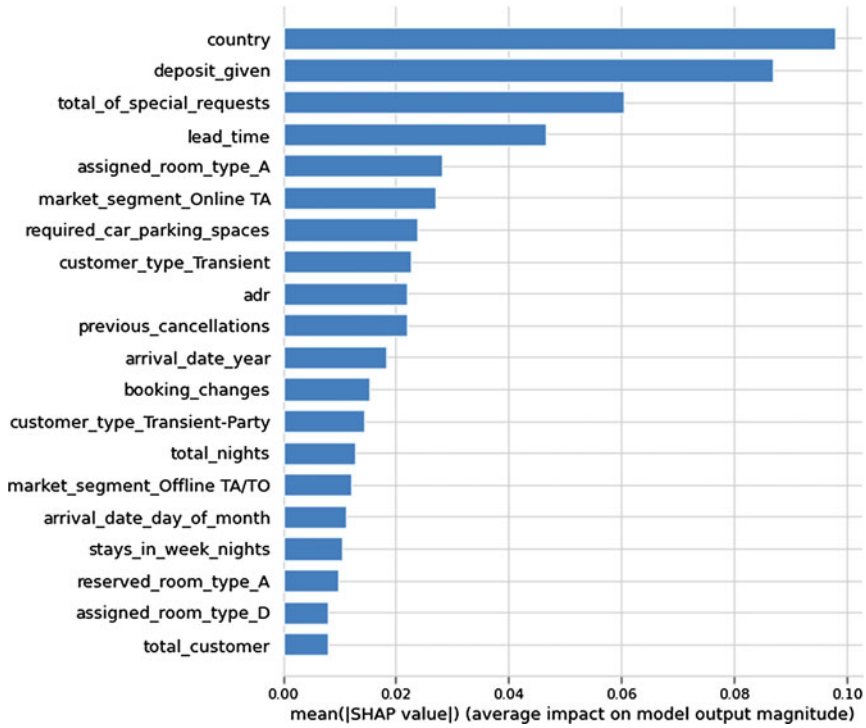


Fig. 5 The impact of variables on the model output computed with Shapley values. Source: Author’s illustration generated with Python code

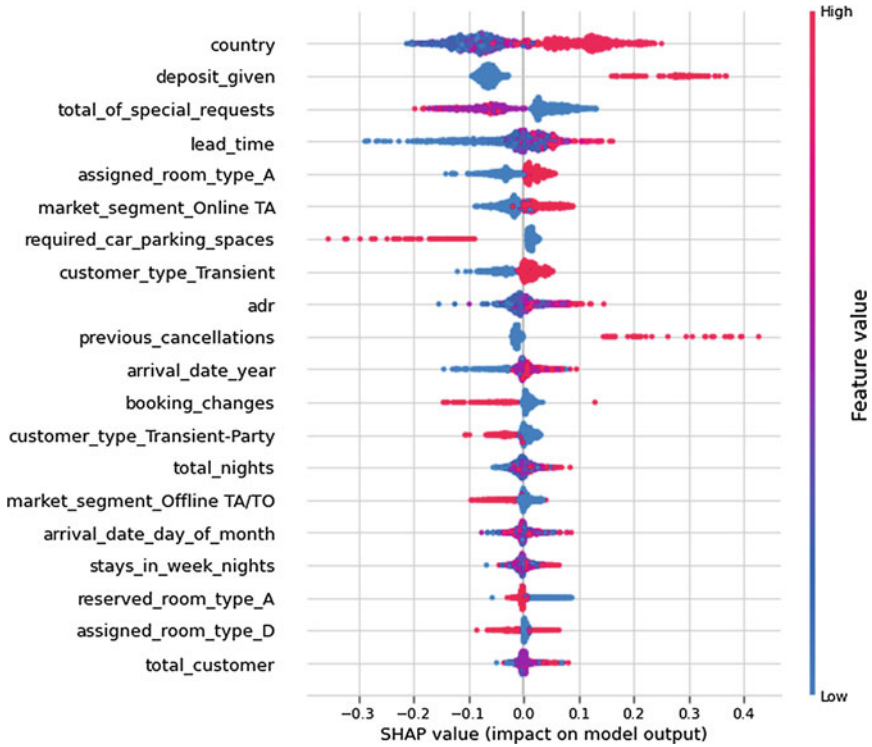
The selected explainer is a TreeExplainer because the Random Forest model belongs to the type of tree models.

Visualizations are one of SHAP library’s strengths. By calling “shap.summary\_plot” with arguments, (1) Shapley values for class 1, (2) feature names (the column names), and (3) type of the plot, the plot with the importance of features is drawn.

Using the parameter plot\_type = “bar”, the global (aggregated) importance of variables can be obtained (Fig. 5). It can be noticed that visitors’ country of origin has an essential impact on the model output, and the deposit paid is an important feature as well. However, it is not known how these features affect the model.

```
shap.summary_plot(shap_values[1], feature_names=x_background.columns, plot_type='bar', max_display=None)
```

More detailed information on how each variable affects the prediction can be obtained through a summary plot by choosing the default option, “dot” plot (Fig. 6).

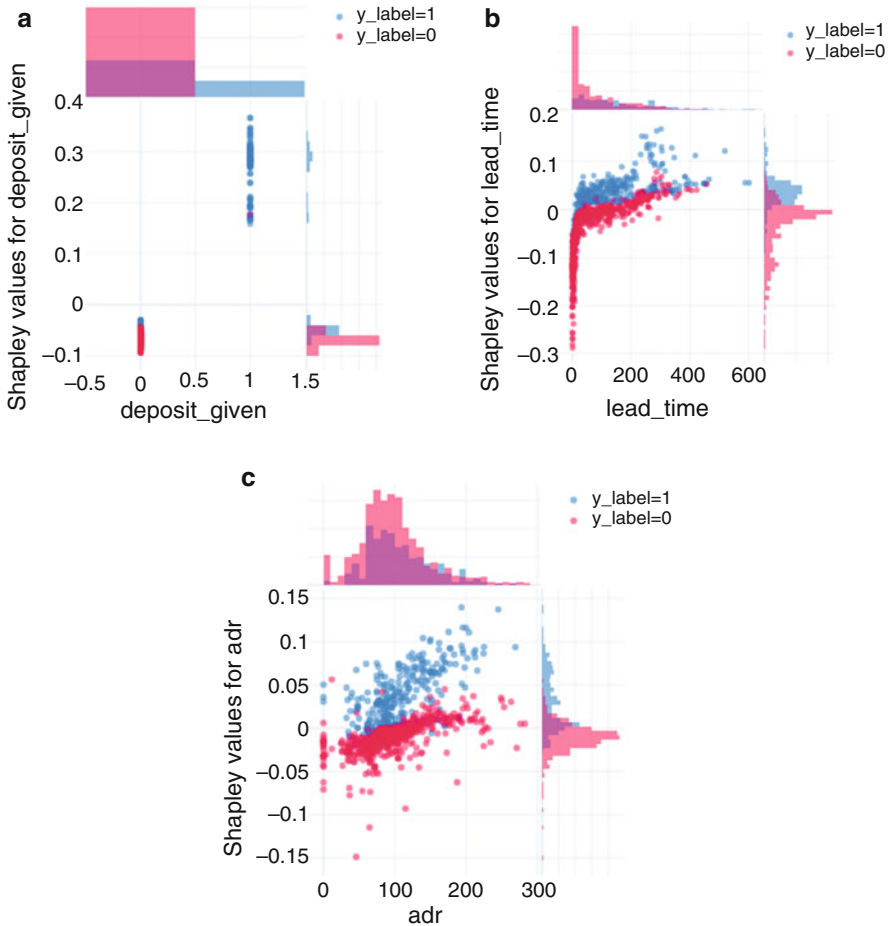


**Fig. 6** Summary plot with the importance of features and the type of impact. Source: Author’s illustration generated with Python code

```
shap.summary_plot(shap_values[1], x_background,  
feature_names=x_background.columns, max_display=None)
```

As mentioned in the data preparation section, each of the country codes was encoded as an integer. Therefore, it does not make much sense to interpret the values of these features directly. If the deposit were to be given (value = 1; pink), it would have a positive impact on the model predicting a cancellation, although this is quite counterintuitive. The no special requests feature also has a positive impact on the model predicting a cancellation. Likewise, the longer the booking until arrival period is, the higher the chance of the booking being canceled, and bookings with room type A or online bookings have a higher cancellation rate as well.

Dependency plots can be used to understand a relationship between variable values, Shapley values, and the predicted class (Fig. 7). In this tutorial, a custom dependency plot is used by calling the “dependence\_plot\_classes” function. It shows the points in the 2D space of x: feature values and y: Shapley values as well as the bars that show the count of the predicted label for each point in the scatter plot. Each



**Fig. 7** The relationship between the feature value (x-axis), Shapley values (y-axis), and predicted value (bars). (a) deposit\_given (b) lead\_time (c) adr. Source: Author’s illustration generated with Python code

feature can be represented in this way by replacing “deposit\_given” with the selected column name.

```
dependence_plot_classes("deposit_given",x_background, df_display, rf_model.predict(x_background), shap_values[1])
```

From Fig. 7a, it can be concluded that when the deposit was given (feature value = 1), all the model predictions were equal to 1 (blue). This gives this feature a crucial predictive value. Similarly, Fig. 7b indicates that, for the “lead\_time” variable, the relationship with the prediction is not clear. For lower values, most of

the predictions are “not canceled”, and over time the “canceled” status appears. This feature most likely plays a more important role when interacting with another feature. Finally, Fig. 7c shows the variable “adr”, which averages the total spending over the number of nights. It can be noted that positive Shapley values are mostly related to class 1. However, there is no clear relationship with the “adr” feature value. Therefore, one can suppose that, similar to “lead\_time”, this variable plays a role in an interaction effect.

### 2.4.2 SHAP: Local Interpretation

Local explanations allow for an explanation analysis of each instance. The SHAP force plot depicts the contribution of each feature to the final prediction (Fig. 8a and b). For a prediction of class 0 (0.02 probability of becoming class 1), most of the features contribute positively to class 0. In contrast, features “market segment online TA” push the prediction towards class 1. The waterfall plot (Fig. 8c and d) ranks the features by importance from top to bottom in which the length of the bars is proportional to the feature contribution, and the color indicates if the impact is positive or negative. From the waterfall plot, for an instance of class 1 (Fig. 8d),



**Fig. 8** Shap local explanation of an instance in the form of force plot and waterfall plot. (a) force plot class 0 (b) force plot class 1 (c) waterfall plot class 0 (d) waterfall plot class 1. Source: Author’s illustration generated with Python code

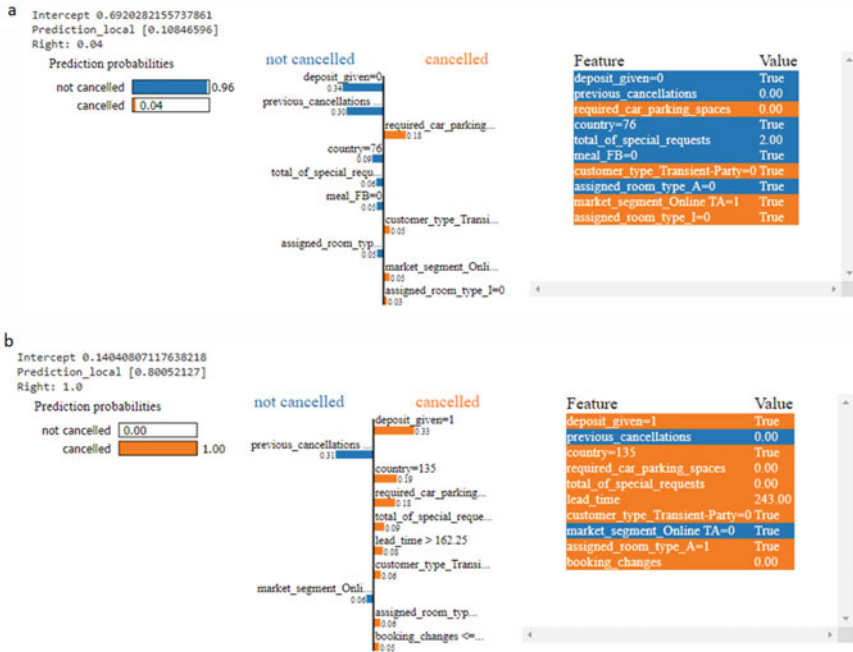
one can read that the country is an important decisive feature. The customers have two special requests, which is a marker of “noncancellation”. Likewise, the “adr” feature is relatively low and also contributed to the “not canceled” prediction. As depicted in the force plot, “market segment online TA”, “non-reserved room type A”, and “Resort Hotel” hotel-type would push the prediction towards “canceled”, but they are outweighed by “negative” impact variables. For an instance predicted correctly to be in “canceled” class, most of the variables push the prediction towards class 1. “Deposit\_given”, “country”, “no special requests”, “lead\_time”, and “assigned\_room\_type\_A” equal one. Contrarily, only “market segment online TA” contributes to the “non-canceled” prediction.

### 2.4.3 Lime

Another framework, LIME, can be used to compare the explanations. However, the data format needs to be adapted for this purpose in which it is preferred to use the pipeline feature from sci-kit learn. LIME can then decode the categorical coded values and show human-readable labels. To simplify and make the comparison more straightforward, here the coded features are conserved, and the explanation will be shown using the same instances as in the SHAP example. Once the library and data is imported, and since we are dealing with tabular data, an explainer is built using “LimeTabularExplainer”. The minimal inputs are the original data formatted as a numerical matrix (NumPy Python package can transform the matrix) with the class names, the feature column names, the indexes of categorical features, and the names of categorical features. The parameter `kernel_width` defines the size of the local prediction. The bigger the `kernel_width`, the more data points the function will take to generalize. The explanation can be computed along with a visualization when calling the `explain_instance` method. The required inputs are data instances and the prediction function (here: “`model_rf.predict_proba()`”).

According to LIME, the most important feature for the instance 1 is “deposit\_given” set to 0. The country is only the fourth most important feature (Fig. 9a), and no “previous cancellations” was the second most important feature. No demand for car parking space contributed to the opposite prediction. Even though the contribution weight has changed when compared with SHAP, the direction of the impact remains the same.

For the selected instance from class 1 (canceled), “deposit given” is the most important feature according to the explanation (Fig. 9b). Country is placed as the third most crucial feature followed by no required car parking space and “lead time”. Moreover, no “previous cancellations” would push the prediction into “non-canceled”.



**Fig. 9** Lime explanation of an instance from (a) class 0 (non-cancelled) and (b) class 1 (cancelled). Source: Author's illustration generated with Python code

```
import lime
import lime.lime_tabular

predict_fn = lambda x: rf_model.predict_proba(x).astype(float)
explainer_lime=lime.lime_tabular.LimeTabularExplainer
(X_background.to_numpy(),class_names=['not cancelled',
'cancelled'], feature_names = X_train.columns,
categorical_features=categorical_columns_index,
categorical_names=categorical_features, kernel_width=3)

i = 1
exp = explainer_lime.explain_instance(X_background.iloc[i].values,
predict_fn)
i = 43
exp = explainer_lime.explain_instance(X_background.iloc[i].values,
predict_fn)
```

## 2.5 Conclusions

It can be concluded that different techniques give rise to different interpretation scenarios. However, they can indicate with consistency which feature values have a substantial impact on the model output. SHAP has the most solid mathematical foundations and uses an explainer that is adapted to tree models, while LIME is model agnostic, and, therefore, the explanations are more prone to be biased by the linear surrogate explainer. Explanations indicate the features that are fundamental to the model without placing too much importance on the specific weights and order.

It remains to be answered whether these explanations that were found can be used in professional applications. It can be hypothesized that this model was created to serve a hotel consortium to better manage their vacancies, but what would be a useful interpretation? First, an indication that if a deposit was given, then the booking would be more certain seems to be wrong. It would be interesting to investigate whether or not the data was collected correctly and whether this conclusion is a result of a labeling error. Then, it seems that some nationalities tend to cancel their holidays more than others. This could be due to the paid holiday system or just random facts, but if the trend is consistent over the years, this information could be used to overbook some places with nationalities of “high cancellation risk”. It would also be advisable to trust in bookings that have special requests as it indicates that the client invests in the stay. To make a booking more certain, the time leading up to a stay could be shortened, for example, by accepting bookings only 100 days in advance. Furthermore, one could also observe from the data that some other information, such as the number of interactions with hotel staff and the context thereof, could be crucial when it comes to preventing or predicting cancellations.

## 3 Research-Case

The explicability frameworks have been employed to solve applied research questions. In the article, “Cascaded Machine Learning Model for Efficient Hotel Recommendations from Air Travel Bookings”, Thomas et al. (2019) used LIME’s feature importance values to improve their recommendation model. The goal of their work was to optimize the hotel recommendation engine and maximize the conversion rate. What features make a hotel desirable? Can one propose a list of hotels tailored to a selected traveler/tourist? Usually, previous customer booking history is used to find the most suitable place. However, what can be done if such data is not available? The authors of the article proposed to build a model based on the historical data of the flight travel information and the hotel features predicting the conversion rate (click-through rate) of a hotel. The best performance was achieved with a random forest model.

Next, the conversion rate was predicted for a list of hotels that would be presented to a traveler. In the article, this is what is called a “session model”. The session model

combines the flight details, aggregates of the hotel features, and aggregates of the individual hotel conversion probabilities. The best performing model for this task was the GBM model. The next goal was to build a list of hotels that would have the highest probability of high conversion rates. This is what authors call a “session builder”. Finally, the “session model’s” most important features are used to make a selection of the hotels. The used GBM model was a black-box, and the classification task was highly imbalanced (most of the hotels do not get clicks). Therefore, the authors selected LIME as an explainability framework to compute feature importance.

As stated in this chapter, when global interpretation is concerned, SHAP would be the preferred technique because LIME can only provide a local model for selected instances. However, the authors cite a good reason for using a local model and aggregate the local values to build global features. First, in highly imbalanced problems, the dominant class may bias the feature importance. Secondly, the authors used only the feature importance of positive instances to perform hotel selection.

LIME identified the most important features in which the hotel conversion model features appeared on the top of this list: the standard deviation, maximum, and average individual hotel conversion probabilities. Some other important features such as the country where the booking was made, the flight class of service, the destination city, and arrival and departure times of the flight could not be used to manipulate the results of the session builder because they were part of the recommendation context. Features extracted from prices (the difference between the average price and the minimum and the lowest price ratio to the average price) were also considered important to the LIME model but ranked lower than many hotel conversion probability features. Therefore, a simple rule based on the most important feature, the standard deviation of hotel conversion, was used. The new hotel list was built by replacing one of the hotels, whose conversion rate was closest to the mean, with another, whose conversion rate was highest, in order to increase the standard deviation. The full pipeline is depicted in Fig. 10.

This method resulted in a significant improvement in the conversion rate for the session model. The authors then compared it with the “brute force method” (i.e., changing hotels at random till the conversion rate improves). The LIME-based method performed close to the “brute force” one, especially for smaller cities, but worked on average 2.8 times faster and even 21 times faster for a single prediction. An evolution of this pipeline would be to perform LIME analysis in real time and dynamically select the best features that would improve the conversion rate.

The work conducted by Thomas et al. (2019) proves that a more comprehensive application of the explainability frameworks could improve, among others, recommendation systems in the tourism industry. It could strengthen the predictions and bring more robustness to the used ML models. Thus, more research must be undertaken in order to show the plethora of possibilities within the tourism industry.



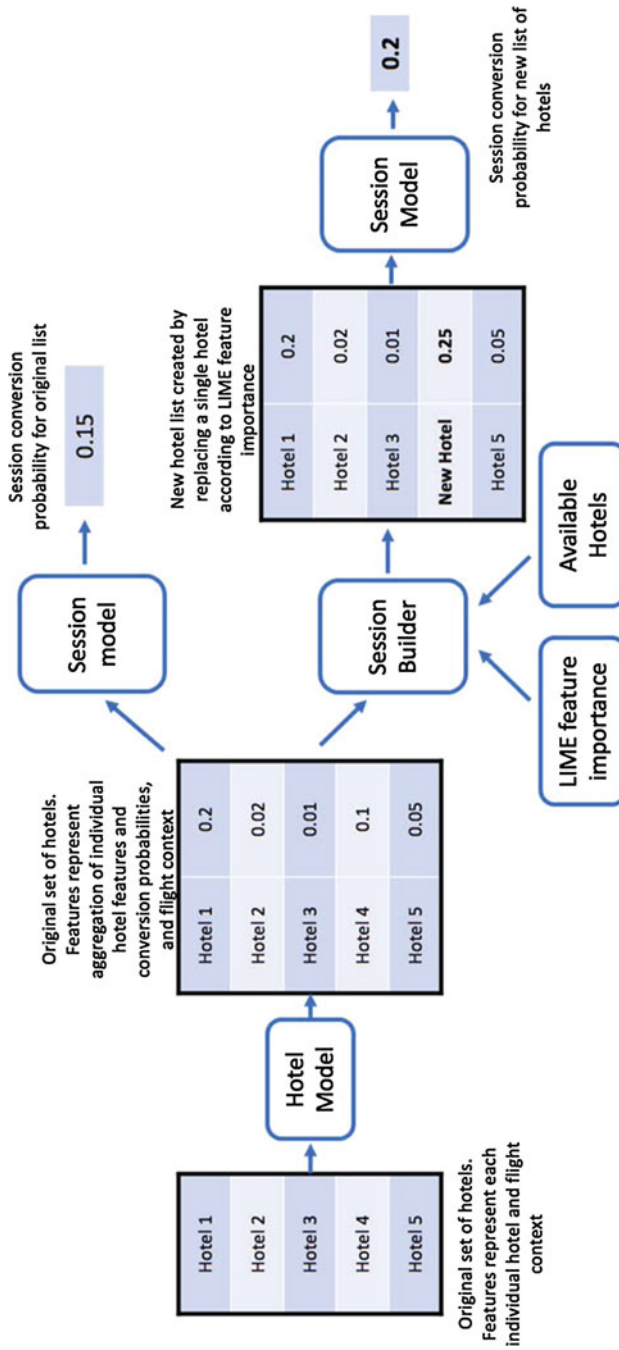


Fig. 10 Schematic representation of the hotel recommendation pipeline. Source: Thomas et al. (2019)

**Service Section**

**Main Application Fields:** Interpretability, or XAI, is a technique that allows for a better understanding of AI models, ensuring that they can be trusted and are not biased. The XAI field is relatively new and dynamic; therefore, several distinct techniques support the notion of explainability. The best known libraries for classical machine learning are SHAP and LIME and DeepLift for deep learning. Explainability techniques are most often applied so as to identify the most important features for a model or a particular prediction.

**Limitations and Pitfalls:** In theory, explainability can be applied to all types of black-box models. However, it is not always trivial to make the results of explainability understandable for a human. Moreover, using explainability frameworks often demands adaptations of the data structure or changes in the model. Some explainability techniques also call for access to the model structure or training data. This is not always possible in real-life conditions, especially since access to non-transformed data can be a sensible or difficult matter. Also, in practice, a combination of machine learning models can be used instead of a single model, and there is no consensus on how to construct an explanation for a stacked models pipeline.

**Similar Methods and Methods to Combine with:** Explainability is always combined with machine learning frameworks. For Python users, that would be sci-kit learn, TensorFlow, or PyTorch. Similar techniques, such as bias detection, model adversary control, or data quality control, can be used or combined with the explainability frameworks as well.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-14-Data-Interpretability-of-ML-Models>

**Acknowledgments** The author would like to thank Peter Nylor for an informal review of the manuscript and advice on using the English language.

**Further Readings and Other Sources**

An excellent presentation of these methods can be found in the Cloudera white paper <https://ff06-2020.fastforwardlabs.com/>

About LIME

<https://towardsdatascience.com/whats-wrong-with-lime-86b335f34612>

Comparing Robustness of LIME and SHAP

<https://arxiv.org/abs/1806.08049>

More about SHAP library

<https://medium.com/swlh/push-the-limits-of-explainability-an-ultimate-guide-to-shap-library-a110af566a02>

<https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30>

<https://medium.com/@stanleyg1/a-detailed-walk-through-of-shap-example-for-interpretable-machine-learning-d265c693ac22>

<https://medium.com/@gabrieltseng/interpreting-complex-models-with-shap-values-1c187db6ec83>

Description of different XAI tools and platforms

<https://medium.com/analytics-vidhya/explainable-ai-the-next-level-c6b4dadcd240>

AI toolbox 360

<https://github.com/Trusted-AI/AIX360>

## References

- Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1), 39–59.
- Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052>
- Adler, P., Falk, C., Friedler, S. A., Rybeck, G., Scheidegger, C., Smith, B., & Venkatasubramanian, S. (2016a). *Auditing black-box models for indirect influenc.*
- Adler, P., Falk, C., Friedler, S. A., Rybeck, G., Scheidegger, C., Smith, B., & Venkatasubramanian, S. (2016b). GitHub - algofairness/BlackBoxAuditing: Research code for auditing and exploring black box machine-learning models. Retrieved January 13, 2021, from Github website.: <https://github.com/algofairness/BlackBoxAuditing>
- Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., & Wallach, H. M. (2018). A reductions approach to fair classification. *CoRR*, abs/1803.0. Retrieved from <http://arxiv.org/abs/1803.02453>
- Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel booking demand datasets. *Data in Brief*, 22, 41–49. <https://doi.org/10.1016/j.dib.2018.11.126>
- Arras, L., Montavon, G., Müller, K. R., & Samek, W. (2017). Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 159–168. Retrieved from <https://github.com/jjweil/>
- Barocas, S., & Selbst, A. D. (2018). Big data’s disparate impact. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2477899>
- Barredo Arrieta, A., Díaz-Rodríguez, N., Ser, J. Del, Bennetot, A., Tabik, S., Barbado, A., . . . , Herrera, F. (2019). *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI*.
- Bastani, O., Kim, C., & Bastani, H. (2017). Interpretability via model extraction. *ArXiv*, abs/1706.0.
- Bellamy, R. K. E., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., . . . Zhang, Y. (2018). *{AI Fairness} 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias*. Retrieved from <https://arxiv.org/abs/1810.01943>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Dandl, S., Molnar, C., Binder, M., & Bischl, B. (2020). Multi-objective counterfactual explanations. In T. Bäck, M. Preuss, A. Deutz, H. Wang, C. Doerr, M. Emmerich, & H. Trautmann

- (Eds.), *Parallel problem solving from nature -- PPSN XVI* (pp. 448–469). Springer International Publishing.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. <https://doi.org/10.1109/4235.996017>
- Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning*. European Commission. (2020). *On artificial intelligence—a European approach to excellence and trust white paper on artificial intelligence a European approach to excellence and trust*. Retrieved from [https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020\\_en.pdf](https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf)
- Feldman, M., Friedler, S., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). *Certifying and removing disparate impact*.
- Fisher, A., Rudin, C., & Dominici, F. (2019). *All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously*.
- Ghorbani, A., Abid, A., & Zou, J. (2019). Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 3681–3688. <https://doi.org/10.1609/aaai.v33i01.33013681>
- Gilmer, J., Adams, R. P., Goodfellow, I. J., Andersen, D. G., & Dahl, G. E. (2018). Motivating the rules of the game for adversarial example research. *CoRR*, abs/1807.0. Retrieved from <http://arxiv.org/abs/1807.06732>
- Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. *2018 IEEE fifth International Conference on Data Science and Advanced Analytics (DSAA)*, 80–89. <https://doi.org/10.1109/DSAA.2018.00018>
- Gregory Piatetsky. (2019, May). *Python leads the 11 top data science, machine learning platforms: Trends and analysis*. Retrieved January 31, 2021, from KDnuggets website: <https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html>
- Hall, P. (2018). On the art and science of machine learning explanations. *ArXiv*, abs/1810.0.
- Hotel booking demand | Kaggle. (2020). Retrieved January 15, 2021, from Kaggle website: <https://www.kaggle.com/jessemostipak/hotel-booking-demand>
- Julius, A. (2016). *FairML: Auditing black-box predictive models*. Retrieved January 13, 2021, from GitHub website: <https://github.com/adebayoj/fairml>
- Kaufman, S., Rosset, S., & Perlich, C. (2011). Leakage in data mining: Formulation, detection, and avoidance. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 6, 556–563. <https://doi.org/10.1145/2020408.2020496>
- Klaise, J., Van Looveren, A., Vacanti, G., & Coca, A. (2019). Alibi: Algorithms for monitoring and explaining machine learning models. URL <https://github.com/SeldonIO/alibi>. Retrieved from <https://github.com/SeldonIO/alibi>
- Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., ... Reblitz-Richardson, O. (2020). *Captum: A unified and generic model interpretability library for PyTorch*.
- Laugel, T., Renard, X., Lesot, M.-J., Marsala, C., & Detryniecki, M. (2018). Defining locality for surrogates in post-hoc interpretability. *Workshop on Human Interpretability for Machine Learning (WHI) – International Conference on Machine Learning (ICML)*. Stockholm, Sweden. Retrieved from <https://hal.sorbonne-universite.fr/hal-01905924>
- Lundberg, S. M., Allen, P. G., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (Vol. 30). Retrieved from <https://github.com/slundberg/shap>
- Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., ... Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67. <https://doi.org/10.1038/s42256-019-0138-9>

- Lundberg S. M., & Lee, S.-I. (2018). *A game theoretic approach to explain the output of any machine learning model*. Retrieved January 13, 2021, from Github website: <https://github.com/slundberg/shap>
- Meudec, R. (2020). GitHub – sicara/tf-explain: Interpretability Methods for tf.keras models with Tensorflow 2.x. Retrieved January 13, 2021, from Github website: <https://github.com/sicara/tf-explain>
- Mock, T., & Bichat, A. (2020). tidyuesday/readme.md at master · rfordatascience/tidyuesday · GitHub. Retrieved January 15, 2021, from Github website: <https://github.com/rfordatascience/tidyuesday/blob/master/data/2020/2020-02-11/readme.md>
- Molnar, C. (2019). *Interpretable machine learning*.
- Nicolae, M.-I., Sinn, M., Minh, T. N., Rawat, A., Wistuba, M., Zantedeschi, V., ... Edwards, B. (2018). Adversarial Robustness Toolbox v0.2.2. *CoRR, abs/1807.0*. Retrieved from <http://arxiv.org/abs/1807.01069>
- Noyes, K. (2015). *The FTC is worried about algorithmic transparency, and you should be too* | *PCWorld*. Retrieved January 13, 2021, from Computer World website: <https://www.pcworld.com/article/2908372/the-ftc-is-worried-about-algorithmic-transparency-and-you-should-be-too.html>
- O’Neil, C. (2017). *Weapons of math destruction: how big data increases inequality and threatens democracy*. Penguin Books.
- Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*. <https://doi.org/10.23915/distill.00007>
- Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. In *Journal of Machine Learning Research* (Vol. 12). Retrieved from <http://scikit-learn.sourceforge.net>.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13–17-August-2016* (pp. 1135–1144). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939778>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2018a). Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32). Retrieved from [www.aaai.org](http://www.aaai.org)
- Ribeiro, T. M., Singh, S., & Guestrin, C. (2018b). Code for “High-precision model-agnostic explanations” paper. Retrieved January 31, 2021, from GitHub website: <https://github.com/marcotcr/anchor>
- Rudin, C. (2019). *Stop explaining black box machine learning models for high stakes decisions and use interpretable models Instead*.
- Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. *34th International Conference on Machine Learning, ICML 2017*, 7, 4844–4866. Retrieved from <http://arxiv.org/abs/1704.02685>
- Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (2020). Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 180–186*. Association for Computing Machinery. <https://doi.org/10.1145/3375627.3375830>
- Sokol, K., & Flach, P. (2020). One explanation does not fit all. *KI-Künstliche Intelligenz*, 1–16.
- Sokol, K., Santos-Rodríguez, R., & Flach, P. A. (2019). {FAT} Forensics: {A} Python toolbox for algorithmic fairness, accountability and transparency. *CoRR, abs/1909.0*. Retrieved from <http://arxiv.org/abs/1909.05167>
- Tan, S., Caruana, R., Hooker, G., & Gordo, A. (2018). Transparent model distillation. *ArXiv*.
- Tao, G., Ma, S., Liu, Y., & Zhang, X. (2018). Attacks meet interpretability: Attribute-steered detection of adversarial samples. *Advances in Neural Information Processing Systems, 31*, 7728–7739.

- TeamHG-Memex. (2016). eli5: A library for debugging/inspecting machine learning classifiers and explaining their predictions. Retrieved January 13, 2021, from Github website: <https://github.com/TeamHG-Memex/eli5>
- Thomas, E., Ferrer, A. G., Lardeux, B., Boudia, M., Haas-Frangii, C., & Agost, R. A. (2019). *Cascaded machine learning model for efficient hotel recommendations from air travel bookings*.
- Tramer, F., Atlidakis, V., Geambasu, R., Hsu, D., Hubaux, J.-P., Humbert, M., . . . Lin, H. (2015). FairTest: Discovering unwarranted associations in data-driven applications. *ArXiv Preprint ArXiv, 1510*, 02377.
- Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2020). *Ensemble adversarial training: Attacks and defenses*.
- Looveren, A. Van, & Klaise, J. (2019). Interpretable counterfactual explanations guided by prototypes. *CoRR, abs/1907.0*. Retrieved from <http://arxiv.org/abs/1907.02584>
- Vapnik, V., & Golowich, S. E. (1997). *Support vector method for function approximation, regression estimation, and signal processing*.
- Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law and Technology, 31*, 841. Retrieved from <http://arxiv.org/abs/1711.00399>

**Part III**  
**Natural Language Processing**

# Natural Language Processing (NLP): An Introduction



## Making Sense of Textual Data

Roman Egger and Enes Gokce

### Learning Objectives

- Illustrate the foundations of Natural Language Processing
- Appreciate where NLP can be applied in tourism research
- Demonstrate how to pre-process texts
- Explain the logic behind each data-cleaning and pre-processing step

## 1 Introduction and Theoretical Foundations

Within both academia and the tourism industry, it is vital to make proper use of textual data. According to a commonly reported statistic, 80% of the data produced every day consists of text (Wennker, 2020). This number brings about numerous challenges as text data is fundamentally unstructured and must be processed appropriately before it can be used for further analysis. The statement “information is the lifeblood of tourism” (Poon, 1993) has never been more appropriate than it is today, especially due to social media channels contributing greatly to the rising importance of user-generated content (UGC) (Conti & Lexhagen, 2020; Aicher et al., 2016). The analysis of customer reviews and posts from social media channels such as Twitter, Facebook, or Instagram has opened up previously unimagined opportunities for companies to better understand customer wishes, needs, and feelings, ultimately

---

R. Egger (✉)

Salzburg University of Applied Sciences, Innovation and Management in Tourism, Urstein (Puch), Salzburg, Austria

e-mail: [Roman.egger@fh-salzburg.ac.at](mailto:Roman.egger@fh-salzburg.ac.at)

E. Gokce

Pennsylvania State University, Pennsylvania, USA



helping businesses to improve their services accordingly (Aicher et al., 2016; Egger, 2010).

The history of text analysis can be traced back to the twelfth century, when the first biblical concordances were written by monks (Ignatow & Mihalcea, 2017). It has also been reported that the first qualitative text analysis was performed in Sweden in the seventeenth century. Thereafter, systematic text analysis experienced a rapid upswing in the twentieth century as numerous methodological approaches for text analysis and associated procedures relating to the qualitative interpretation of texts and documents were developed in the social sciences (Bussière, 2018). More recently, a change towards the digital analysis of texts has been observed (Rockwell, 2003), and the enormous amount of data makes digital processing and analysis inevitable. Yet, text, unlike numerical data, poses particular challenges for computer-aided analyses. Take the following description of a dish on a restaurant menu as an example:

*Italian dish made of stacked layers of thin flat pasta, alternating with fillings such as ragù and other vegetables, cheese, and seasonings and spices such as garlic, oregano and basil, topped with melted grated mozzarella cheese.*

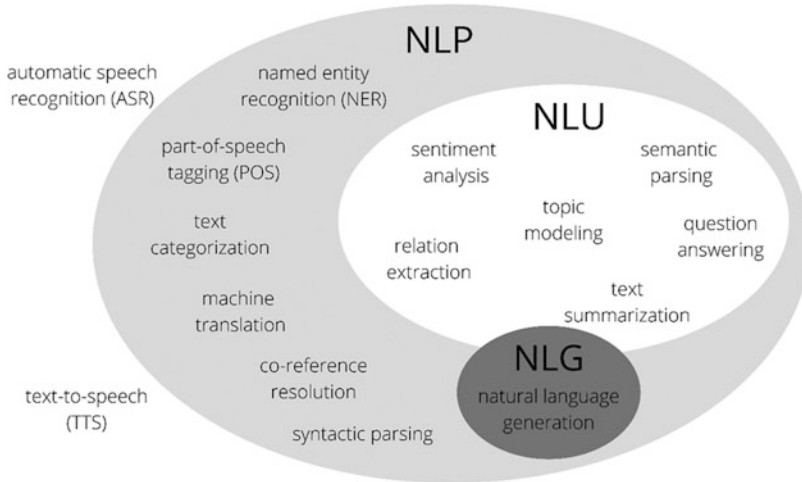
The thought of lasagna should instantly come to mind; for a computer, however, the context is difficult to grasp, and/or making a connection to the term *lasagna* is challenging.

Driven by advances in machine learning, Natural Language Processing (NLP) is now a well-established field in artificial intelligence, linguistics, computer science, and information technology. NLP uses machines to read, understand, and make sense of human language in order to streamline data mining, data analysis, and business operations (Han et al., 2016). Alongside the improvements in NLP, there has been a shift from traditional manual coding to the automation of data collection, data cleaning, and statistical analysis (Li et al., 2019).

NLP's general task is to make a computer understand written language in the form of words, sentences, or paragraphs. Thus, the overall aim of NLP is for machines and computers to be able to successfully accomplish tasks concerning natural language. Natural language refers to any human language that was developed through natural circumstances and follows a specific syntactic and semantic system (Sarkar, 2019). Hapke et al. (2019) defines NLP as:

*an area of research in computer science and artificial intelligence (AI) concerned with processing natural languages such as English or Mandarin. This processing generally involves translating natural language into data (numbers) that a computer can use to learn about the world. And this understanding of the world is sometimes used to generate natural language text that reflects that understanding (p. 4).*

Natural Language Processing (NLP) and Natural Language Understanding (NLU) are two closely related concepts in which confusion often arises between the demarcations of the two terms as well as misinterpretations of other subfields of artificial intelligence. NLU first came about in the 1960s, approximately ten years after NLP, out of the need to understand increasingly complex language input. While NLP covers all areas of communication between humans and computers, from input to processing to response, NLU seeks to understand content (MacCartney, 2014).



**Fig. 1** Terminology relating to NLP, NLU, and NLG. Source: adapted from MacCartney (2014)

NLP is thus an overarching concept, and Natural Language Understanding (NLU) and Natural Language Generation (NLG) are sub-disciplines thereof. For the sake of simplicity, however, in this chapter, we will only refer to NLP, which includes NLU and NLG (Fig. 1).

NLP is considered to be very multifaceted and has already been integrated into various everyday situations. Machine translation, such as Google Translate or DeepL, allows machines to successfully translate natural language from one language to another (Sarkar, 2019), while text prediction (i.e. auto-correction) can aid in quickly solving typos and grammatical errors (Naik, 2020). Text summarisation, on the other hand, takes a text or collection of texts and expels a reduced summary based on the most prominent keywords, phrases, or sentences that appear. If, however, a large corpus of texts is presented, topic modelling would be a more potent way to extract and summarise essential concepts and themes (Kao & Poteet, 2007). Lastly, when it comes to texts with subjective content (e.g. feedback surveys, reviews, etc.), sentiment analysis or opinion mining would be considered the best fit (Ignatow & Mihalcea, 2017).

## 2 Text Analysis in Tourism

With the use of social media and user-generated content, a flood of unstructured data in the form of text, images, videos, and audio has become the dominant source of data in tourism research (Xiang, 2018). In order to be able to analyse the vast amounts of user-generated content available, automated processes that prepare unstructured texts for further processing and interpretation are needed. As such, in

the field of NLP applications, topic modelling (compare Chapter “Topic Modeling”) and sentiment analysis (compare Chapter “Sentiment Analysis”) seem to be the most common techniques implemented by tourism researchers (Yu & Egger, 2021; Alaei et al., 2017; Munezero et al., 2014; Guerreiro & Rita, 2020; Li et al., 2020; Chang et al., 2020). Sentiment analyses make it possible to capture tourists’ feelings based on a given text (Markopoulos et al., 2015). By using TripAdvisor, Expedia, and Yelp datasets, Xiang et al. (2017), for example, identified the semantic features and sentiment scores of online reviews. Their study also highlights the importance of social media analytics in tourism and hospitality as insights from text data can promote smart tourism development (Li et al., 2019). Another recent study applied text mining techniques to reveal the antecedents of tourism recommendations on the Yelp platform (Guerreiro & Rita, 2020). Supported by previous literature, Han et al. (2016) reinforce the suggestion that tourism researchers should go beyond numerical ratings and interpret quantitative results with text data to reveal tourists’ true feelings. In topic modelling (e.g. Latent Dirichlet Allocation), on the other hand, different topics are extracted from an existing text. In this way, by structuring them thematically and preparing them for further analysis, an overview of the text can be obtained. Topic modelling methods are often also combined with sentiment analysis via looking at the sentiments within the extracted topics (Calheiros et al., 2017). The results generated by most topic modelling algorithms must be interpreted by individual identified keywords, which is often a difficult task (Hannigan et al., 2019).

Recent literature has introduced Deep Learning approaches to NLP as a way to minimise existing limitations in text analysis (Chang et al., 2020; Ma et al., 2018). For example, a study by Chang et al. (2020) processed tourist reviews and comments by integrating visual analytics and Deep Learning-based NLP to assist hoteliers with strategic planning. Furthermore, Named Entity Recognition (NER) (Chantrapornchai & Tunsakul, 2019) and co-reference resolution (García-Pablos et al., 2016) have gradually gained popularity within the tourism domain as well. The former is based on information extraction in which predefined codes are automatically assigned to text elements, while the latter identifies expressions that refer to the same entity in natural language (Hannigan et al., 2019). Examples of such entities include the names of hotels, restaurants, people, countries, cities, or destinations (García-Pablos et al., 2016). More details on NER will be presented below. Beyond the field of tourism marketing and management, the use of NLP can also help to better understand socio-cultural aspects in the field of tourism. For instance, Li et al. (2020) examined online comments to conceptualise the impact of global racism on tourism experiences.

In addition, the possibilities of text summarisation have been explored further in the context of tourism. Tsai, Chen, Hu, & Chen (2020) presented approaches on how to create summaries of online hotel reviews, while Yang et al. (2020), for example, dealt with question-answering systems and built such a system based on a tourism knowledge graph. Knowledge-based reasoning systems are often used for recommendation systems in which these techniques use domain-specific knowledge to rate items and predict how useful they are for a specific user (Ricci, 2020).

Table 1 provides an overview of selected articles applying a text analytics approach within the tourism domain.

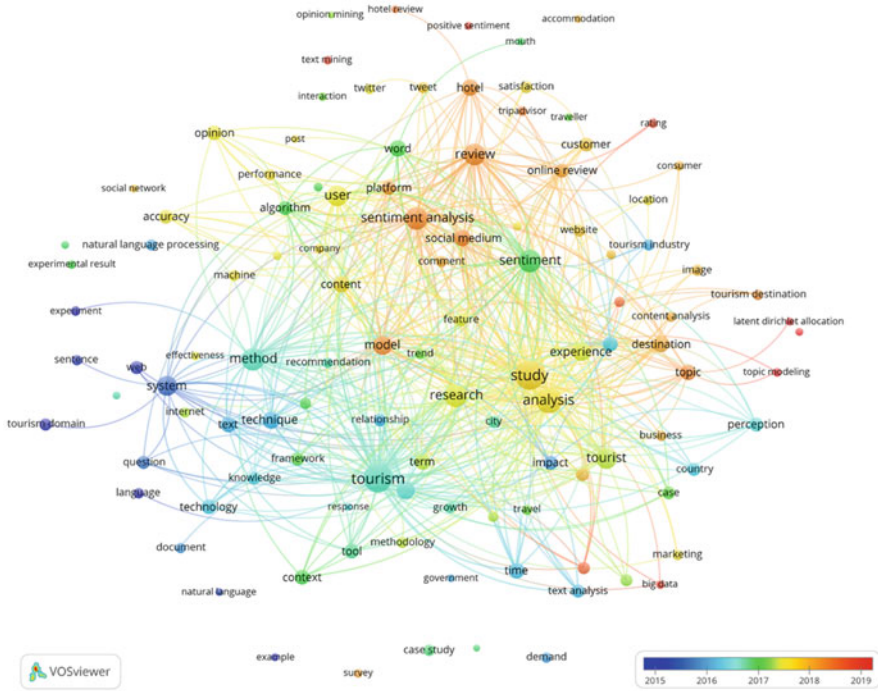
**Table 1** Text analysis in tourism

Name	Year	Title	Objectives	Methodology	Software
Yu & Egger	2021	Tourist experiences at overcrowded attractions: A text analytics approach	To explore the perception and feelings of tourists when visiting overcrowded attractions	Topic modelling; sentiment analysis	Python; Orange
egger & Yu	2021	Identifying hidden semantic structures in Instagram data: a topic modelling comparison	Evaluating the effectiveness of different topic modelling algorithms	Topic modelling	Python
Chang, Ku, & Chen	2020	Using deep learning and visual analytics to explore hotel reviews and responses	To analyse hotel reviews and responses collected on TripAdvisor and to identify response strategies	Visual analytics; deep learning-based natural language processing	Selenium; TripCollective; geocoding API; tableau
Chen, Wang, Zhu, & Lian	2020	Will you miss me if I am leaving? Unexpected market withdrawal of Norwegian Joy and customer satisfaction	To investigate how customer satisfaction changes as a response to a cruise's market-withdrawal decision	Sentiment analysis	Baidu Senta system
Li, Li, Law, & Paradies	2020	Racism in tourism reviews	To validate the impact of racial discrimination on tourists' experience	Semantic analysis; sentiment analysis	Python
Andreu, Bigne, Amaro, & Palomo	2020	Airbnb research: An analysis in tourism and hospitality journals	To examine Airbnb research using bibliometric methods	Stemming; n-grams detection	R
Guerreiro & Rita	2020	How to predict explicit recommendations in online reviews using text mining and sentiment analysis	To explore what may drive reviewers to make direct endorsements in text	Text mining; sentiment analysis	SPSS modeler text analytics
Zhang, Yang, Zhang, & Zhang	2020	Designing tourist experiences amidst air pollution: A spatial analytical approach using social media	To propose a spatial analytical framework from geotagged social media data in Beijing for a better understanding of tourist experiences	Sentiment analysis; deep learning classifiers; econometric analysis	Linguistic inquiry and word count (LIWC) program

(continued)

Table 1 (continued)

Name	Year	Title	Objectives	Methodology	Software
Tsai, Chen, Hu, & Chen	2020	Improving text summarisation of online hotel reviews with review helpfulness and sentiment	To evaluate how review helpfulness and hotel features improve the results of hotel review summarisation	Segmentation; stemming; part of speech	Stanford CoreNLP
Chantrapornchai & Tunsakul	2019	Information extraction based on named entity for tourism corpus	To extract information in order to help with ontology data acquisition within the tourism domain	Named entity recognition	Python; spaCy
Deng, Liu, Dai, & Li	2019	Different cultures, different photos: A comparison of Shanghai's pictorial destination image between east and west	To propose a new method to compare destination image differences among inbound tourists	Part of speech; sentiment analysis	Python; TextBlob
Ban, Joung, & Kim	2019	The text mining approach to understand seat comfort experience of airline passengers through online review	To explore the seat comfort experience of airline passengers via online reviews	Text mining; semantic network analysis	Python; NetDraw
Yadav & Roychoudhury	2019	Effect of trip mode on opinion about hotel aspects: A social media analysis approach	To uncover the aspects of hotels discussed by people from online reviews	Aspect-based sentiment analysis; part of speech	R
Fazzolari & Petrocchi	2018	A study on online travel reviews through intelligent data analysis	To assist providers in adapting their services to help customers improve their decision processes based on review data	Tokenisation; language-specific and domain-specific stop words; stemming	Python; Scikit-learn
Antonio, de Almeida, Nunes, Batista, & Ribeiro	2018	Hotel online reviews: Creating a multi-source aggregated index	To obtain a prediction model for hotel review ratings	Sentiment analysis	C#; R



**Fig. 2** NLP development in tourism

Figure 2 shows the development of Natural Language Processing in the field of tourism based on 356 identified articles. Relevant terms such as “Text Analysis”, “Natural Language Processing”, “Topic Modeling”, “Sentiment Analysis”, “Text Summarization”, etc., were used in combination with “Tourism” to obtain articles from Web of Science and Scopus. One can clearly see that sentiment analysis has been one of the most frequently applied techniques since 2018. Contrarily, topic modelling, mostly by applying the Latent Dirichlet Allocation approach, only first developed in 2019, whereby, as already indicated, there is a strong connection to sentiment analysis. The graph also clearly shows that online reviews are the main source for text analyses.

### 3 NLP Techniques

Driven by advances in machine learning, NLP has been a well-established field under the realm of artificial intelligence, linguistics, computer science, and information engineering. NLP uses machines to read, understand, and make sense of human languages so as to streamline data mining, data analysis, and business operations (Han et al., 2016). Owing to improvements in NLP, a sea change from conventional

In this chapter	In this book	Not in this book
Text cleaning	Information Retrieval and Extraction	Machine Translation
Tokenizing	Relationship Extraction	Text Summarization
Stemming	Entity matching	Natural Language Generation
Lemmatization	Text Similarity	Knowledge-Based Reasoning
Part of Speech Tagging (POS)	Sentiment Analysis	Question Answering Systems
Named Entity Recognition (NER)	Topic Modeling	Query Expansion
EDA - Visualization Wordcloud	Knowledge Graphs	Multimodel Tasks

**Fig. 3** NLP techniques covered in this book

manual coding to the automation of data collection, data cleaning, and statistical analysis has occurred. Essentially, NLP covers diverse tools and techniques such as named entity recognition, information extraction, topic classification, text analysis, sentiment analysis, and word embedding, amongst many others (Li et al., 2019).

Figure 3 shows the techniques presented in this chapter/book, respectively, as well as additional advanced tools that are not discussed here.

## 4 Text Preparation and Pre-processing

This chapter is mainly dedicated to the pre-processing of texts. “Garbage in, garbage out” is an accurate statement that can also be applied to the analysis of texts since if the pre-processing stage is not carried out properly, then the algorithms used will subsequently have problems processing the content correctly. Text pre-processing can be understood as a series of operations that are necessary for a machine to automatically process texts (Kannan & Gurusamy, 2014). Thus, the main purpose of text cleaning is to systematise the text data (Albishre et al., 2015). Uncleaned data

can contain many potential issues, such as misspelt words, incorrect punctuation, improper spacing, etc., and using uncleaned data can even distort the document's linguistics and inhibit information extraction processes (Saralegi & Leturia, 2007).

Moreover, in NLP methods, each word is viewed as a variable (dimension). On the one hand, the aim is trying to keep the vocabulary and, thus, the dimensions as small as possible, while, on the other hand, also attempting to keep them large enough so that no critical information is lost. Removing noise from the document can reduce computational costs and increase NLP models' performance (Kumar & Babu, 2019).

It must be noted that text cleaning is an intensive process. It is estimated that text pre-processing takes about 80% of the time, whereas only 20% of the effort is required for data analysis (IDC, 2018). During text cleaning, the researcher makes subjective decisions that can lead to biased results of the analysis; therefore, text cleaning has a crucial impact on text analysis and the final results. In the following, the different steps needed for text data pre-processing are described in detail.

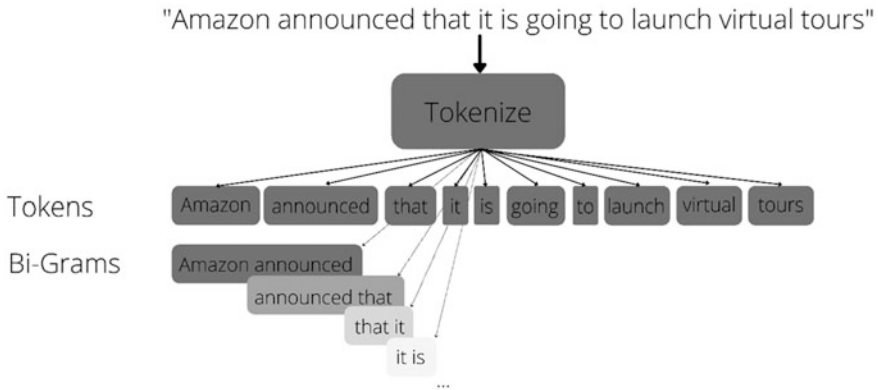
## 4.1 *Language Detection*

When it comes to texts, in the context of tourism especially, corpora often consist of documents in several languages. Until recently, most algorithms have required one specific language to be processed; however, multilingual algorithms have now become more common (Keung et al., 2020). At the same time, research often focuses on analysing texts in a defined language. As such, language detection algorithms like *spacy-langdetect* allow the language of documents to be identified. It makes sense to perform the language detection at the beginning of the text wrangling process in order to filter just those documents from the entire corpus that correspond to the target language and need to be processed further.

## 4.2 *Tokenisation*

To perform NLP tasks, a robust vocabulary is needed. Each corpus exists of a number of documents, which represent one instance each and serve as a unique identifier. This means that each document collection (corpus) consists of individual documents that are formed by individual tokens. To receive the required vocabulary, a document needs to be split into distinct tokens of meaning (Hapke et al., 2019). Languages such as English, German, or French are space-delimited, meaning that most terms are separated by white spaces. For languages like Thai or Chinese, however, this is not the case; as these languages do not provide clear boundaries between words, tokenisation becomes a challenging task (Kannan et al., 2014). Tokenisation is often the first step in a classic NLP pipeline (Cook et al., 2016),





**Fig. 4** The tokenisation process

but, if required for specific reasons, it can also be done at a later point in time, as will be shown in our practical demonstration.

The following document in Fig. 4 exists of 10 individual words. By tokenising this text, it is split into more useable tokens.

Often, however, one may not only want to represent a single word as a token, but also successive words. N-grams are, therefore, alternatives to single words. A bi-gram, for example, consists of two words in a row, a tri-gram of three words in a row, and so on (Anandarajan et al., 2019; Sarkar, 2019).

### 4.3 Lowercasing and Removal of Punctuation

Usually, the transformation of text into lowercase letters and the removal of punctuation are considered the first pre-processing steps. Lowercase text data is particularly relevant so as to avoid word redundancy. For example, when counting words, the terms “Tourism” and “tourism” would be counted as two separate words, leading to an undesired increase in the dimensions of the data. At the same time, punctuation marks create noise in the data and do not add value to the analysis, explaining why they should be removed. However, in certain situations, it makes sense to analyse individual sentences; in such cases, the corpus should be separated into individual sentences before punctuation marks are removed.

### 4.4 Expand Contractions

A contraction is “a shortening of a word, syllable, or word group by omission of a sound or letter” (Merriam-Webster, 2021); for instance, a contraction like “we’ll” is

split into two words, “we” and “will”. Again, the idea is to normalise the text, i.e. to turn tokens that mean the same thing into a normalised form. Thus, the vocabulary does not unnecessarily increase, which leads to the fact that the association of the different spellings of a token has decayed. At the same time, the likelihood of overfitting can be reduced (Hapke et al., 2019).

## 4.5 *Removal of Stop Words*

Stop words are common words in English such as “a, in, the, can, may”, and so on. These words are not useful in NLP analyses because they can be part of any sentence; in other words, stop words are not considered keywords in text mining methods (Porter, 1980). Furthermore, they increase vocabulary and hardly provide useful information (Vijayarani et al., 2015).

There are numerous Python packages that can be used to load stop words, with each package varying in size. For example, the NLTK (Natural Language Toolkit) package stores a list of stop words for 16 different languages, with 127 stop words specified for English (Bird et al., 2009), while the Stanford NLP package contains 257 English stop words (Qi et al., 2018). In most cases, one will use standard stop words but can add individually defined stop words as well.

## 4.6 *Removal of URLs, HTML Tags, and Emotions/Emojis*

Regarding social media posts from Twitter, Facebook, or Instagram, URLs are often a part of documents. However, these limit NLP algorithms from recognising the actual meaning of a sentence. Thus, they are merely noise within a text and should be deleted. (Baldwin et al., 2013; Sarker & Gonzalez, 2016; Kumar & Babu, 2019). The same applies to HTML tags as well as emojis and emoticons. Emojis are symbols such as, while an emoticon is a character such as:-). Even though they convey information about feelings, we might not want to analyse them and are better off being removed from the data.

## 4.7 *Correction of Spelling*

Spelling errors can be seen as another example of increasing the number of features in a vocabulary due to corresponding words being counted twice. For example, the words “tourism” and “turism” would be considered two different words. To correct spelling errors, many different Python packages using different approaches are available, such as “*pyspellchecker*”, “*SymSpell*”, “*TextBlob Spell Checker*”, etc. However, their execution time and performance differ significantly. Spelling

correction algorithms use two main approaches: context-sensitive spelling correction and isolated term spelling correction. In the latter, the algorithm focuses on a single document at a time, such as a single tweet, and attempts to correct it (Schütze et al., 2008). Here is an example using *TextBlob*, illustrating the power of spelling correction packages:

*Original sentence:* “Minnesota is briming with natual and cltural baauty.”

*After spelling correction:* “Minnesota is brimming with natural and cultural beauty.”

## 4.8 Stemming and Lemmatisation

Stemming and lemmatisation are two very similar concepts but should not be used together. In both cases, the overall goal is to break down the words until only the root is remaining (Mendez et al., 2005). As with the other pre-processing steps, the aim is to avoid redundancies between terms (in this case, the same root) in order to reduce the vocabulary.

Take the following as an example: “*recreation*” is reduced to “*recreat*”.

Since terms with the same root often contain similar meanings, they can be combined into a single token. Yet, although there are terms in which a reduction to the root word leads to different meanings (e.g. booking → book), the incorrect assignment of some words is accepted for the dimension reduction gained (Anandarajan et al., 2019). Popular stemmers include the *Porter Stemmer* (Porter, 1980) and the *Snowball Stemmer* (Porter, 2001).

In contrast to stemming, lemmatising takes the term’s part of speech into consideration. When a document is lemmatised, each word in that document is replaced with its lemma, which means that the verbs in the output become the uninflected form of the verb, or the base form, and all nouns are presented in the singular form (Siemens, 1996). For instance, with lemmatisation, “caring” can be transformed to “care” (Table 2).

Depending on the needs of the researcher, different Python libraries are available for the lemmatisation process. Some of them reduce the number of words more than

**Table 2** Comparison of a word’s original, stemmed, and lemmatised versions

Original word	Stemmed	Lemmatised
tourist	tourist	tourist
booking	book	book
rating	rate	rat
itinerary	itinerari	itinerary
recreation	recreat	recreation
amenities	amen	amenities
attractions	attract	attractions
sightseeing	sightse	sightsee
eating	eat	eat

others, but two of the most popular lemmatisation packages include *WordNet*<sup>1</sup> Lemmatizer and *UDPipe*<sup>2</sup> Lemmatizer.

### 4.9 Part of Speech Tagging (POS)

With part of speech tagging, individual tokens are labelled based on their parts of speech (Anandarajan et al., 2019). This can be achieved using language models that include dictionaries of terms with all their possible parts of speech (Hapke et al., 2019); for instance, both *NLTK*<sup>3</sup> and *spaCy* are Python modules that enable extensive POS. The example below shows the parts of speech for the sentence “information is the lifeblood of tourism” by using *spaCy*<sup>4</sup> and visualising the dependencies with *displaCy* (Fig. 5).

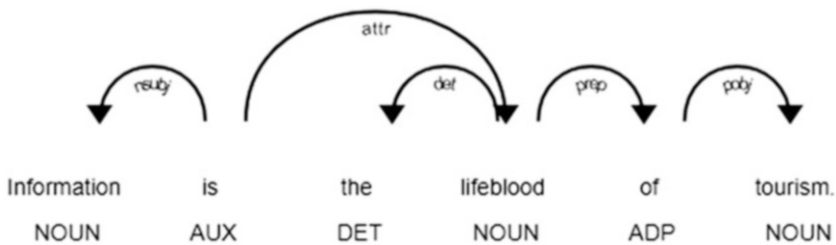


Fig. 5 Part of Speech tagging example

### 4.10 Named Entity Recognition (NER)

Named entity recognition, also known as entity identification or entity chunking, is an advanced statistical procedure that can no longer be assigned to the pre-processing stage but, rather, to the field of information extraction. Via an entity recognition system, labels (metainformation) can be assigned to contiguous spans of tokens (spaCy, 2021). For example, from the following sentence,

*Mr. Egger has booked a room from Monday to Friday at the Sheraton in Vienna for €130 per night*

<sup>1</sup><https://wordnet.princeton.edu/>

<sup>2</sup><https://ufal.mff.cuni.cz/udpipe/1>

<sup>3</sup><https://www.nltk.org/>

<sup>4</sup><https://spacy.io/>

Mr. Egger PERSON has booked a room from Monday to Friday DATE at the Sheraton FAC in Vienna GPE for € 130 MONEY per night

**Fig. 6** Named entity recognition example

it can be concluded that “Egger” is a person, “Monday” and “Friday” represent a date unit, “Sheraton” represents an organisation, “Salzburg” is a place, and “130” stands for an amount of money. For Fig. 6, *spaCy*’s entity visualiser was used.

By extracting the identified labels, new features can be generated and used for further analysis, which brings us to the final technique presented in this chapter.

### 4.11 Feature Extraction

Feature extraction is another essential part of text pre-processing. In the context of machine learning, features represent a variable, and in feature extraction, new variables are extracted from text data to be used for further analysis. These features could be the number of stop words, hashtags or numerical characters, the average word length, etc. When further NLP analyses are carried out, new features emerge that can also be constantly extracted, for example, sentiment analysis results in sentiment scores, term weights in topic modelling or entities (extracted by performing NER), and so on.

### 4.12 Visual EDA

Wordclouds are a helpful way to visualise text data and get an overview of word frequencies and the importance placed upon them. Typically, the maximum number of words to be displayed as well as parameters such as font size, colour, etc. can be defined. Additionally, a word frequency analysis, sentence length analysis, etc. can be performed and visualised, mainly using bar charts for categorical data and histograms for continuous features.

With the methods presented here, text data can be prepared and pre-processed in a suitable fashion. To be able to carry out text analysis approaches presented in the following chapters, the texts, however, must first be converted into an appropriate format. This can be done by converting the text data that resulted from pre-processing into frequencies (Hapke et al., 2019). The following chapter (16) on “Text Representation and Word Embeddings” is dedicated to these approaches.

## 5 Challenges of Working with Text

Although NLP is continuously increasing in popularity and can be applied to many scenarios, it comes with a number of challenges as well. Moreover, due to the fact that different languages involve various grammatical constructions, which NLP may be incapable of identifying and/or deciphering, working with textual data can easily turn into a complex task. Challenges of working with textile data stem from five main reasons:

**Synonymy:** Most NLP approaches pay attention to the frequency of words occurring in the text whilst analysing them. Due to the presence of synonyms, these results may be biased, and the similarity of sentences may be incorrectly evaluated. NLP methods that deal with synonyms usually focus on unigrams (single-word terms) (Blondel & Senellart, 2002), while some other tasks such as machine translation and text simplification consider the similarity between multi-word terms (Hazem & Daille, 2018).

**Lexical ambiguity:** Lexical ambiguity exists in situations where a term or phrase is considered a homonym (Fielding et al., 2017); in other words, certain words possess multiple meanings. For example, the word “watch” can be used as a noun such as in the case of “I bought a new watch”, or it can also be used as a word in a sentence such as “watch your back”.

**Language-related issues:** Different languages have different grammatical structures and rules. For example, in Turkish, there is only one pronoun for he/she/it. These types of cases are challenging for text data analysis.

**Referential ambiguity:** In many situations, it is not clear what an entity may be referring to (Boyarskaya, 2019). In the example of “Jennifer met her friend at the restaurant before she went back to the hotel”. Is she referring to Jennifer or her friend? As such, many different types of reference-ambiguity resolution systems (Manning, 2019) have been rapidly optimised in recent years thanks to the use of neural networks. Referential ambiguity, however, still poses a problem in automatic text analysis.

**Out-of-vocabulary problem:** Computers can only work with terms if they have seen them before, making new and previously unknown words difficult to process correctly. Therefore, manually created lexical databases are of high significance for text analysis. A commonly used lexical corpus and database with an extensive list of different entities is WordNet.<sup>5</sup> Here, English nouns, verbs, adjectives, and adverbs are grouped into synsets based on semantic similarities, resulting in semantic relations between words already being recorded (Sarkar, 2019). WordNet can therefore be understood as a semantic network of meaningfully related words and concepts.

---

<sup>5</sup><https://wordnet.princeton.edu/>

## 6 Practical Demonstration

In this section, we will use a Twitter dataset to perform and showcase the most relevant pre-processing steps in text analytics. The corpus used for this demonstration contains 6027 tweets that we crawled for the hashtag #travelsomeday. This very popular hashtag during the COVID-19 pandemic has been used by individuals to tag a post and express their desire to travel again. The code provided below highlights the most relevant elements only, but the dataset, together with the full code and detailed markdowns with a stepwise explanation, will be shared as a Jupyter Notebook and can be downloaded from the books' Github profile. It must be noted that if you are using another Python IDE, minor changes might be required when it comes to the code, installations, and requirements.

After importing the necessary modules and loading the text data into a Pandas data frame, it is now time for data cleaning. We will start by converting all the text into lowercase and expanding the contractions, allowing us to turn terms like “we’re” into “we are” or “I’d” into “I would”.

```
## Lowercase
df['tweet'] = df['tweet'].apply(lambda x: " ".join(x.lower() for x in
x.split()))
## For contradictions
df['tweet'] = df['tweet'].apply(lambda x: contractions.fix(x))
df.tweet.sample(3)
```

Output:

```
4080 share a moment that makes you smile. tell us w...
4192 great american summer road trips: https://t.co...
5168 7 reasons to visit #romania #travelsomeday #ar...
Name: tweet, dtype: object
```

Language detection is another useful tool for data cleaning. Especially for texts relating to tourism, being able to filter out solely the target language is an important step towards acquiring useful data. When the Language Detection algorithm is applied, the existing languages in the corpus can be seen (see output table below). After detecting all the languages in the documents, the desired language can be kept, and the rest of the tweets in the other languages can be dropped.

```
## Language detection
```

```
import os, re, nltk, spacy, string, umap
pd.set_option('display.max_colwidth', 50)
nlp = spacy.load('en_core_web_sm')
!pip install spacy-langdetect
from spacy_langdetect import LanguageDetector
nlp.add_pipe(LanguageDetector(), name='language_detector',
last=True)
# Defining 'detect language' function
def detect_language(tweet):
```

```

try:
doc = nlp(t)
language = doc._.language['language']
score = doc._.language['score']
except:
language = ''
print('sth goes wrong')
return language, score
languages = []
scores = []
for i, t in df['tweet'].iteritems():
l, s = detect_language(t)
if i % 500 == 0:
print(i, t, l)
languages.append(l)
scores.append(s)

df['lng'] = languages
df['lng_score'] = scores
# Check the output
df[['tweet', 'lng', 'lng_score']].sample(10).round(2)
df.lng.value_counts()
# Keep only the 'English' language
df = df.loc[df.lng == 'en']
# Example sentences
it was such a beautiful sunset
l'anno prossimo verremo di nuovo
お勧めのレストランです。
    
```

**Output:**

ID	Tweet	lng	lng_score
1745	it was such a beautiful sunset	en	1.0
1648	l'anno prossimo verremo di nuovo	it	1.0
885	お勧めのレストランです。	jp	0.9

The next cleaning step is removing contractions. This step should be performed before word tokenisation as NLTK has a built-in method dealing with contractions. NLTK, however, only separates contractions without expanding them.

**## Expand contractions**

```

# creating an empty list
expanded_words = []
for word in text.split():
# using contractions.fix to expand the shortened words
expanded_words.append(contractions.fix(word))

expanded_text = ' '.join(expanded_words)
print('Original text: ' + text)
    
```



```
print('\n')
print('Expanded_text: ' + expanded_text)
```

```
# Example sentence
```

```
She'll be airport in 30 mins. We are supposed to catch the arrival,
aren't we?
```

```
I'd love to welcome her personally. It'll be an awesome vacation.
```

```
Output:
```

```
She will be airport in 30 mins. We are supposed to catch the arrival, are
not we? I would love to welcome her personally. it will be an awesome
vacation.
```

Next, we will move on to stop word removal.

### ## Stopword removal

```
df['tweet'] = df['tweet'].apply(lambda x: " ".join(x for x in x.split()
if x not in stop))
df['tweet'].sample(5)
```

It might be the case that the predefined stop words do not suffice for your task, in which case you might need to additionally define your own set of stop words.

### ## Adding your own stopwords

```
add_words = ["also", "retweet", "comment", ]
stop_words = set(stopwords.words("english"))
stop_added = stop_words.union(add_words)
df['tweet'] = df['tweet'].apply(lambda x: " ".join(x for x in x.split()
if x not in stop_added))
df['tweet'].sample(3)
```

Texts that come from social media platforms in particular, like Twitter, Facebook, or Instagram, and contain URLs should be removed as they hinder the NLP algorithms' detection of the actual meaning of a sentence (Baldwin et al., 2013; Sarker & Gonzalez, 2016; Kumar & Babu, 2019). As they merely create noise in our data, we will delete all URLs in our corpus.

### ## Removing URLs

```
def remove_urls (vTEXT) :
    vTEXT = re.sub(r'(https|http)?:\//(\w|\.|\/|\?|\=|\&|\%)*\b', '',
vTEXT, flags=re.MULTILINE)
    return (vTEXT)
df['tweet'] = df.tweet.apply(remove_urls)
df.tweet.head()
```

```
# Example sentence
sentence= 'NPR can provide useful information https://www.npr.org'
remove_urls(sentence)
```

Output :  
 'NPR can provide useful information'

We may also need to strip the texts of any HTML tags.

**## Stripping HTML tags**

```
def strip_html_tags(text):
    soup = BeautifulSoup(text, "html.parser")
    stripped_text = soup.get_text()
    return stripped_text
df['tweet'] = df['tweet'].apply(lambda x: strip_html_tags(x))
df['tweet'].head()
```

```
# Example sentence
strip_html_tags('People and neighbors are for the most part very
friendly,</a> even in stores</td>')
```

Output :  
 'People and neighbors are for the most part very friendly, even in stores'

Unless you want to analyse emojis, which could indeed be interesting as they convey information about feelings, they should also be removed from the data.

**## Removing emojis**

```
def remove_emoji(text):
    emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F" # emoticons
        u"\U0001F300-\U0001F5FF" # symbols & pictographs
        u"\U0001F680-\U0001F6FF" # transport & map symbols
        u"\U0001F1E0-\U0001F1FF" # flags
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        "]" + "", flags=re.UNICODE)
    return emoji_pattern.sub(r'', text)
```

```
# Example
remove_emoji("Have fun with NLP!😄😁")
```

Output :  
 'Have fun with NLP! '

```
# Remove all emojis from tweets
df['tweet'] = df['tweet'].apply(lambda x: remove_emoji(x))
```

In the steps above, we only removed the emojis, but, in this case, it is also relevant to remove emoticons.

**## Remove emoticons**

```
!pip install emot
from emot.emo_unicode import UNICODE_EMO, EMOTICONS

# Function for removing emoticons
def remove_emoticons(text):
    emoticon_pattern = re.compile(u'(' + u'|'.join(k for k in EMOTICONS) +
u')')
    return emoticon_pattern.sub(r'', text)
df['tweet'] = df['tweet'].apply(lambda x: remove_emoticons(x))
df.tweet.sample(5)

# Example
remove_emoticons("This hotel is just awesome :)")
```

Output:  
‘This hotel is just awesome’

Next, we want to group together the inflected versions of a word, which allows us to analyse them as a single term. In this case, we will apply lemmatisation only.

**## Lemmatisation**

```
word_list = ["tourist", "booking", "rating", "itinerary", "recreation"]
print("{0:20}{1:20}{2:20}".format("Original
Word", "Stemmed", "Lemmatized"))

for word in word_list:
    print("{0:20}{1:20}{2:20}".format(word, porter.stem(word),
wordnet_lemmatizer.lemmatize(word, pos="v")))
```

Output:  
**Original Word Stemmed Lemmatized**  
tourist tourist tourist  
booking book book  
rating rate rat  
itinerary itinerari itinerary  
recreation recreat recreation

**## Lemmatisation & Tokenisation**

```
def LemmaSentence(sentence):
    token_words=word_tokenize(sentence)
    token_words
    lemma_sentence=[]
    for word in token_words:
        lemma_sentence.append(wordnet_lemmatizer.lemmatize(word, pos='v'))
    lemma_sentence.append(" ")
    return "".join(lemma_sentence)
```

```
df['tweet'] = df['tweet'].apply(lambda x: LemmaSentence(x))
df.tweet.sample(3)
```

Output :

```
671 make friends new city # travelsomeday # armcha...
2098 enter # colombia emergency # passport # travel...
2143 free things phoenix, # arizona # travelsomeda...
```

N-Grams are usually done together with the tokenisation process (Thanaki, 2017). As we did tokenisation together with the lemmatisation above, here, an individual, typical n-Gram code is presented.

### ## n-grams

```
from nltk import ngrams
sentence = 'Information is the lifeblood of tourism'
n = 3 # defines the number of tokens. Here a tri-gram
ngramsres = ngrams(sentence.split(), n)
for grams in ngramsres:
    print(grams)
```

Output :

```
('Information', 'is', 'the')
('is', 'the', 'lifeblood')
('the', 'lifeblood', 'of')
('lifeblood', 'of', 'tourism')
```

Spelling correction helps to identify typos and correct them. This is needed as, otherwise, the number of features would increase, and, for example, “standing” and “standng” words would be regarded as two different words. As previously mentioned, there are many different Python packages that use different approaches for spelling corrections, yet their execution time and performance differ drastically.

### ## Spell correction

```
# Example sentences for spell correction
print(TextBlob('In Chicago, a lakeside is a goodf llocation to relax').
      correct())
print(TextBlob('Minnesota is briming with natural and cltural bauty').
      correct())
```

Output :

```
'In Chicago, a lakeside is a good location to relax'
'Minnesota is brimming with natural and cultural beauty'
```

```
%time df['tweet'][:10].apply(lambda x: str(TextBlob(x).correct()))
```

As punctuation marks create noise in the data, they should also be excluded.

### ## Removing punctuation

```
df['tweet'] = df['tweet'].str.replace('[^\w\s]', '')
```

The following NER and POS example is not based on the Twitter dataset; instead, a short paragraph has been used to showcase these tools. *spaCy* allows the importation of a huge number of models for many different languages and in different sizes. For this example, we will load the smallest pipeline “en\_core\_web\_sm”.

### ## Named Entity Recognition (NER)

```
import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_sm")
```

```
doc = nlp(u"On September 29th, Amazon announced that it is going to launch virtual tours and activities. The launch will start in U.S soon and the service will be available in Europe next year. Jeff Bezos is happy about this 1 Billion $ business.")
```

```
for tokens in doc.ents:
    print(tokens)
```

Let us see which entities *spaCy* was able to extract.

Output :

```
September 29th
Amazon
U.S
Europe
next year
Jeff Bezos
this 1 Billion $
```

Next, we would like that all identified entities are highlighted by colour based on entity category.

```
displacy.render(doc, style="ent", jupyter=True)
```

```
On September 29th DATE , Amazon ORG announced that it is going to launch virtual tours and activities. The launch will start in U.S GPE soon and the service will be available in Europe LOC next year DATE . Jeff Bezos PERSON is happy about this 1 Billion $ MONEY business.
```

Finally, POS can be performed so as to see the different lemmas in this sentence.

## Part of Speech Tagging (POS)

```
def show_lemmas(doc):
    for token in doc:
        print (f'{token.text:12} {token.pos_:6} {token.lemma:<22} {token.lemma_}')

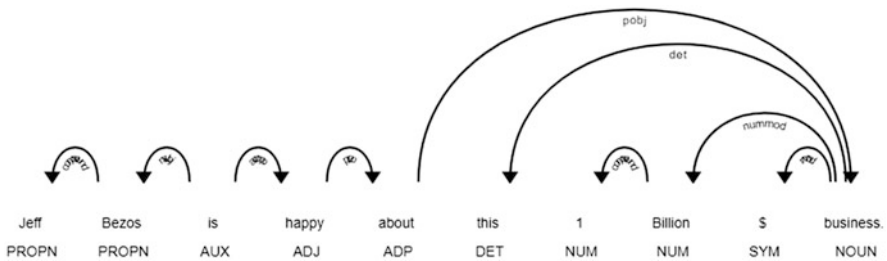
show_lemmas(doc)
```

Output :

On ADP 5640369432778651323 on  
 September PROPN 4843961188194855601 September  
 29th NOUN 14143533789794879514 29th  
 , PUNCT 2593208677638477497 ,  
 Amazon PROPN 7854762596996286480 Amazon  
 announced VERB 6100327276114004729 announce  
 that CONJ 4380130941430378203 that  
 it PRON 561228191312463089 -PRON-  
 is AUX 10382539506755952630 be  
 going VERB 8004577259940138793 go  
 to PART 3791531372978436496 to  
 launch VERB 1882817931903534611 launch  
 virtual ADJ 14425350317076788470 virtual  
 tours NOUN 6598908817114149421 tour

Using *spaCy*'s *displaCy* visualiser, we then get an overview of our example sentence along with its dependencies.

```
displacy.render(doc, style="dep", jupyter=True, options={"distance" : 90})
```



To obtain a first brief overview of the pre-processed text, simple wordclouds are most suitable as they show the frequency of the single words on the basis of the displayed size.

**## Wordcloud**

```

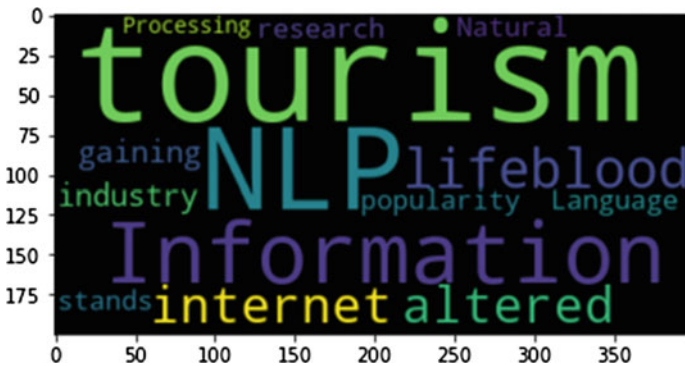
from wordcloud import WordCloud
import matplotlib.pyplot as plt

text = 'Information is the lifeblood of Tourism. The internet has altered
the tourism industry. NLP is gaining popularity in tourism research. NLP
stands for Natural Language Processing. '
# Generate a word cloud image
wordcloud = WordCloud().generate(text)

plt.imshow(wordcloud, interpolation="bilinear")
plt.show()

```

Output :



## 6.1 Tips for Using Python for an NLP Study

Despite the fact that we import packages and predefined functions many times, in many cases, it is important to check how the code is written since a code may be doing more than what it is originally designed for. For example, while using the `contractions` functions in Python, we need to pay attention to how it works as the “`contractions`” package requires apostrophes in order to identify them. In this situation, if you happen to have removed all punctuations before the `contractions`, then the `contraction` function will not be able to find apostrophes and, in turn, fail to perform its task.

**Service Section**

**Main Application Fields:** Natural Language Processing is a prominent field of Data Science, which is especially important in tourism due to the many different text data created by customers. Thus, opinions, sentiments, topics, and much more can be extracted from texts.

**Limitations and Pitfalls:** In order to be able to work analytically with texts, intensive pre-processing is required in most cases. This process step is very labour-intensive and error-prone.

**Similar Methods and Methods to Combine with:** If the text data gets vectorized (represented in numerical values), then any other ML task can be performed (classification, regression, clustering, etc.).

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-15-Introduction-Natural-Language-Processing>

**Further Readings and Other Sources**

Quantum stat provides a list with more than 300 NLP Colab-Notebooks, providing an excellent overview by describing the notebook, the language-model used, and the NLP tasks it is designed for. <https://notebooks.quantumstat.com/>

Ivan Bilan, the author of chapter 19 (Entity Matching), has established "The NLP Pandect", an incredible comprehensive and helpful collection covering almost all topics on NLP. Among them are compendiums, conference papers, NLP datasets, links to podcasts, newsletters, meetups, YouTube channels, and much more. <https://github.com/ivan-bilan/The-NLP-Pandect>

The University of Michigan offers a complete NLP course on Youtube <https://tinyurl.com/NLP-michigan>, and we also recommend to check for free courses on coursea.org, like the ones from DeepLearning.AI <https://tinyurl.com/deeplearningai-course> or the HSE University <https://tinyurl.com/NLP-HSE-course>.

Finally, a great NLP course is also provided by Lena Voita, who is teaching at the Yandex School of Data Analysis. [https://lena-voita.github.io/nlp\\_course.html](https://lena-voita.github.io/nlp_course.html)

**References**

- Aicher, J., Asiimwe, F., Batchuluun, B., Hauschild, M., Zöhrer, M., & Egger, R. (2016). Online hotel reviews: Rating symbols or text... text or rating symbols? That is the question! In A. Inversini & R. Schegg (Eds.), *Information and communication Technologies in Tourism 2016* (pp. 369–382). Springer International Publishing.
- Alaei, A. R., Becken, S., & Stantic, B. (2017). Sentiment analysis in tourism: Capitalising on big data. *Journal of Travel Research*, 1(9), 175–191.



- Albishre, K., Albathan, M., & Li, Y. (2015, December). Effective 20 newsgroups dataset cleaning. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* (Vol. 3, pp. 98–101). IEEE.
- Anandarajan, M., Hill, C., & Nolan, T. (2019). *Practical text analytics* (Vol. 2). Springer International Publishing.
- Baldwin, T., Cook, P., Lui, M., MacKinlay, A., & Wang, L. (2013, October). How noisy social media text, how different social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing* (pp. 356–364).
- Bird, S., Loper, E., & Klein, E. (2009). *Natural language processing with python*. O'Reilly Media.
- Blondel, V. D., & Senellart, P. P. (2002). Automatic extraction of synonyms in a dictionary. *vertex*, 1, x1.
- Boyarskaya, E. (2019). Ambiguity matters in linguistics and translation. *Слово.ру: балтийский акцент*, 10(3), 81–93. <https://doi.org/10.5922/2225-5346-2019-3-6>
- Bussi re, K. (2018). Chapter 4 – Text analysis (digital humanities - a primer). Available online at <https://carletonu.pressbooks.pub/digh5000/chapter/chapter-4-text-analysis/>.
- Calheiros, A. C., Moro, S., & Rita, P. (2017). Sentiment classification of consumer-generated online reviews using topic modeling. *Journal of Hospitality Marketing & Management*, 26(7), 675–693.
- Chang, Y. C., Ku, C. H., & Chen, C. H. (2020). Using deep learning and visual analytics to explore hotel reviews and responses. *Tourism Management*, 80, 104129.
- Chantrapornchai, C., & Tunsakul, A. (2019). Information extraction based on named entity for tourism corpus. In *2019 16th International Joint Conference on Computer Science and Software Engineering* (pp. 187–192). IEEE.
- Conti, E., & Lexhagen, M. (2020). Instagramming nature-based tourism experiences: A netnographic study of online photography and value creation. *Tourism Management Perspectives*, 34, 2–3.
- Cook, P., Evert, S., Sch fer, R., & Stemle, E. (Eds.). (2016). *Proceedings of the 10th Web as Corpus Workshop*. Association for Computational Linguistics.
- Egger, R. (2010). Theorizing web 2.0 phenomena in tourism: A sociological signpost. *Information Technology & Tourism*, 12(2), 125–137. <https://doi.org/10.3727/109830510X12887971002666>
- Fielding, N. G., Lee, R. M., & Blank, G. (2017). *The SAGE handbook of online research methods*. SAGE Publications Ltd.
- Garc a-Pablos, A., Cuadros, M., & Linaza, M. T. (2016). Automatic analysis of textual hotel reviews. *Information Technology & Tourism*, 16(1), 45–69.
- Guerreiro, J., & Rita, P. (2020). How to predict explicit recommendations in online reviews using text mining and sentiment analysis. *Journal of Hospitality and Tourism Management*, 43, 269–272.
- Han, H. J.; Mankad, S.; Gavirneni, N.; Verma, R. (2016). What guests really think of your hotel: Text analytics of online customer reviews. *Cornell Hospitality report*, 16(2), 3–17. Available online at <https://scholarship.sha.cornell.edu/cgi/viewcontent.cgi?article=1003&context=chrreports>, checked on 4/5/2019.
- Hannigan, T. R., Haans, R. F. J., Vakili, K., Tchalian, H., Glaser, V. L., Wang, M. S., Kaplan, S., & Jennings, P. D. (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, 13(2), 586–632.
- Hapke, H. M., Lane, H., & Howard, C. (2019). *Natural language processing in action*. Manning.
- Hazem, A., & Daille, B. (2018, May). Word embedding approach for synonym extraction of multi-word terms. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- IDC (2018). *Time Crunch: Equalising time spent on data management vs analytics*. <https://blogs.idc.com/2018/08/23/time-crunch-equalizing-time-spent-on-data-management-vs-analytics/>
- Ignatow, G., & Mihalcea, R. (2017). *Text mining: A guidebook for the social sciences*. SAGE Publications, Inc.

- Kannan, S., & Gurusamy, V. (2014). Pre-processing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1), 7–16.
- Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., & Nithya, M. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1), 7–16.
- Kao, A., & Poteet, S. R. (2007). *Natural language processing and text mining*. Springer.
- Keung, P., Lu, Y., Szarvas, G., & Smith, N. A. (2020). *The multilingual Amazon reviews corpus*.
- Kumar, C. P., & Babu, L. D. (2019). Novel text pre-processing framework for sentiment analysis. In *Smart intelligent computing and applications* (pp. 309–317). Springer.
- Li, S., Li, G., Law, R., & Paradies, Y. (2020). Racism in tourism reviews. *Tourism Management*, 80, 104100.
- Li, Q., Li, S., Zhang, S., Hu, J., & Hu, J. (2019). A review of text corpus-based tourism big data mining. *Applied Sciences*, 9(16), 3300. <https://doi.org/10.3390/app9163300>
- Ma, Y., Xiang, Z., Du, Q., & Fan, W. (2018). Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep learning. *International Journal of Hospitality Management*, 71, 120–131.
- MacCartney, B. (2014). Understanding natural language understanding. ACM SIGAI Bay Area Chapter Inaugural Meeting, 2014. Available online at <https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf>.
- Manning, C. (2019, March 21). *Coreference Resolution [Video]*. Youtube. [https://www.youtube.com/watch?v=i19m4GzBhfc&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=16&ab\\_channel=stanfordonline](https://www.youtube.com/watch?v=i19m4GzBhfc&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=16&ab_channel=stanfordonline)
- Markopoulos, G., Mikros, G., Iliadi, A., & Liontos, M. (2015). Sentiment analysis of hotel reviews in Greek: A comparison of unigram features. In *Cultural tourism in a digital era* (pp. 373–383). Springer.
- Mendez, J. R., Iglesias, E. L., Fdez-Riverola, F., Diaz, F., & Corchado, J. M. (2005, November). Tokenising, stemming and stopword removal on anti-spam filtering domain. In *Conference of the Spanish Association for Artificial Intelligence* (pp. 449–458). Springer.
- Merriam-Webster. (2021). Contraction. In *Merriam-Webster.com* dictionary. Retrieved January 14, 2021, from <https://www.merriam-webster.com/dictionary/contraction>
- Munezero, M., Montero, C. S., Sutinen, E., & Pajunen, J. (2014). Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE Transactions on Affective Computing*, 5(2), 101–111.
- Poon, A. (1993). *Tourism, technology and competitive strategies*. CAB International.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137.
- Porter, M. F. (2001). *Snowball: A language for stemming algorithms*. Available online at <http://snowball.tartarus.org/texts/introduction.html>.
- Qi, P., Dozat, T., Zhang, Y., Manning, C. D., 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies*.
- Ricci, F. (2020). Recommender systems in Tourism. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-Tourism* (pp. 1–18). Springer International Publishing; Imprint Springer.
- Rockwell, G. (2003). What is text analysis, really? *Literary and Linguistic Computing*, 18(2), 209–219.
- Saralegi, X., & Leturia, I. (2007). Kimatu, a tool for cleaning non-content text parts from HTML docs. In *Proceedings of the 3rd Web as Corpus Workshop* (pp. 163–167).
- Sarkar, D. (2019). *Text analytics with python*. Apress.
- Sarker, A., & Gonzalez, G. (2016, December). Data, tools and resources for mining social media drug chatter. In *Proceedings of the fifth workshop on building and evaluating resources for biomedical text mining (BioTxtM2016)* (pp. 99–107).
- Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39, pp. 1041–4347). Cambridge University Press.
- Siemens, R. (1996). *Lemmatization and parsing with TACT pre-processing programs*. Digital Studies/Le champ numérique.

- Thanaki, J. (2017). *Python natural language processing. Explore NLP with machine learning and deep learning techniques*. Packt.
- Tsai, C.-F., Chen, K., Hu, Y.-H., & Chen, W.-K. (2020). Improving text summarization of online hotel reviews with review helpfulness and sentiment. In *Tourism Management*, 80, 104122. <https://doi.org/10.1016/j.tourman.2020.104122>
- Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Pre-processing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1), 7–16.
- Wennker, P. (2020). *Künstliche Intelligenz in der Praxis. Anwendung in Unternehmen und Branchen: KI wettbewerbs- und zukunftsorientiert Einsetzen*. Springer Gabler. Available online at <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6326361>
- Xiang, Z. (2018). From digitisation to the age of acceleration: On information technology and tourism. *Tourism Management Perspectives*, 25, 147–150.
- Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2017). A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism. *Tourism Management*, 58, 51–65.
- Yang, L., Cao, H., Hao, F., Zhang, W. Z., & Ahmad, M. (2020). Research on tourism question answering system based on xi'an tourism knowledge graph. *Journal of Physics: Conference Series*, 1616(1), 12090. <https://doi.org/10.1088/1742-6596/1616/1/012090>
- Yu, J., & Egger, R. (2021). Tourist experiences at overcrowded attractions: A text analytics approach. In W. Wörndl, C. Koo, & J. L. Stienmetz (Eds.), *Information and Communication Technologies in Tourism 2021. Proceedings of the ENTER 2021 eTourism Conference, January 19–22, 2021* (pp. 231–243). Springer.

# Text Representations and Word Embeddings



## Vectorizing Textual Data

Roman Egger

### Learning Objectives

- Illustrate the intuition behind text representations
- Explain the most important embedding algorithms
- Appreciate the implementation of word embeddings in the field of tourism
- Demonstrate how to vectorize textual data

## 1 Introduction and Theoretical Foundations

In the previous chapter, text data was preprocessed and manipulated to such an extent that further analyses could be applied. While the stringing together of letters and signs in the form of words and sentences is achievable for us humans, provided one can correctly decipher symbols/signs based on the knowledge of a certain language, text data must be prepared accordingly for computers. To make text usable for quantitative analysis methods via computers, it must first be transformed into numerical values. Furthermore, many machine learning algorithms require a fixed-length feature vector as input (Le & Mikolov, 2014). For this reason, this chapter discusses the various methods of word representation and word embedding, portraying basic concepts of Natural Language Processing (NLP).

The overall aim of this chapter is to present different approaches to vectorizing text, which can be viewed as a form of feature engineering textual data. The order of the methods is purposely presented in such a way so as to reflect the increasing

---

R. Egger (✉)

Salzburg University of Applied Sciences, Innovation and Management in Tourism, Urstein (Puch), Salzburg, Austria

e-mail: [Roman.egger@fh-salzburg.ac.at](mailto:Roman.egger@fh-salzburg.ac.at)

complexity of the algorithms, starting with one hot encoding and ending with deep neural networks such as BERT.

In this regard, however, please keep in mind that the selection of the appropriate method should be based on the task and not necessarily on the assumption that state-of-the-art approaches guarantee high-quality results.

## 1.1 One Hot Encoding

The simplest word representation can be achieved via “One Hot Encoding,” a process in which categorical variables are represented as binary vectors. In the first step, the categorical values are mapped to integer values, and, subsequently, each value can be represented as a binary vector in the form of a 0 or a 1.

In Fig. 1, for each token, the column containing this word is filled with the value 1, and the remaining values are filled with a value of zero. This results in a vector with the dimension  $1 \times (N + 1)$ , where the size of the dictionary is represented by “ $N$ ” and the additional 1 is added to  $N$  for the “out-of-vocabulary” token.<sup>1</sup> In reality, the words of a sentence would be stored in a dictionary, and the example from Fig. 1 would have a random word order.

This method is favorable as it is uncomplicated to implement, the interpretation is simple, and it does not suffer from undesirable bias. It does, however, have one major drawback. As only words can be identified when using this method, the context of the terms gets lost. In this sense, since the terms stand alone and have no further reference to neighboring words, sentences, or paragraphs, it is impossible for the computer to learn the meaning of the terms. Additionally, this method results in highly sparse vectors, requiring a large memory capacity for computation. Thus, more complex methods are needed in order to perceive the context in which the terms occur.

Feature for:	Information	is	the	lifeblood	of	tourism
information	1	0	0	0	0	0
lifeblood	0	0	0	1	0	0
tourism	0	0	0	0	0	1
etc.						

Fig. 1 One hot encoding

<sup>1</sup><https://towardsdatascience.com/word-representation-in-natural-language-processing-part-i-e4cd54fed3d4>

## 1.2 Bag-of-Words (CountVectorizer)

By far, the most common approach to vectorizing text data is based on the “Bag-of-Words” (BOW) (Harris, 1954) representation, which summarizes a word sequence representation for a document by computing a histogram of words for the document’s word sequence and resulting in a fixed vector (Duboue, 2020). The idea of BOW is to find features (words) within a document that help unlock its meaning or allow comparison between documents in terms of similarity. Before vectorizing a text with Bag-of-Words, it is preprocessed in most cases, as shown in the previous chapter. All unnecessary special characters must be removed, the data needs to be normalized, and lemmatization or stemming should be performed. Through tokenization, one then receives the words in the document that one wishes to count. It should be mentioned, that BOW ignores the word order as they appear in a document.

For example, after lowercasing, stopword removal, and stemming (Porter Stemmer) have been applied, the sentence “*This hotel is a city hotel, and it is located in the city centre of Salzburg*” is represented as: ~~this~~(-1), hotel (2), ~~is~~(-2), ~~a~~(-1), citi (2), ~~and~~(-1), it (-1), locat (1), ~~in~~(-1), ~~the~~(-1), centre (1) ~~of~~(-1), salzburg (1).

The vocabulary count of our document now contains all unique features, including:

*hotel* – 2  
*citi* – 2  
*locat* – 1  
*centre* – 1  
*salzburg* – 1.

The final vector for our sentence “*This **hotel** is a **city** hotel and it is **located** in the city **centre** of **Salzburg***” is [02002000100101], representing the final vocabulary and its count compared to the words from our document.

Using the BOW method creates flat vectors, meaning that the original text structure becomes lost and the BOW no longer contains sequences and word order. Each word represents one dimension of the vector, where the order of the words within the vector is irrelevant, provided it is consistent across all documents in the corpus (Zheng & Casari, 2018). When the Bag-of-Words erases some semantic meaning of a sentence, the breaking down of individual words can subsequently lead to undesirable effects. For example, in the case of a pair of terms such as “not expensive,” they are split into two individual and independent words, “not” and “expensive.” Bag-of-n-Grams can solve this problem to a certain extent (Mikolov et al., 2017) but should not be considered a definitive solution. Generally speaking, the BOW approach is sufficient for simple tasks such as document classification or information retrieval as it judges documents as being similar if they show a similar distribution of specific words (Dong & Liu, 2017). Nevertheless, it is far from optimal when it comes to providing correct semantic understanding of the text (Zheng & Casari, 2018).

### 1.3 TF-IDF

The weaknesses of BOW Boolean can be partially overcome by weighting words. A widely used approach is the “Term Frequency - Inverse Dense Frequency (TF-IDF)” (BOW-TF-IDF) technique, which determines the importance or relevance of a word or n-gram, yet not the word meaning, in a document or corpus (Wang et al., 2020). The disadvantage of BOW is that the frequency of a term in a document does nothing to help in distinguishing its relevance. This means that words that occur less frequently but make the context easier to understand are neglected. In the TF-IDF transformation, however, a weighting of terms is carried out, and a score is calculated for the relevance of each word given in the document. Let's take the following two sentences, representing document A and B, as an example.

Documents	Text	Total number of words in a document
A	Information is the lifeblood of tourism	6
B	The internet has altered the tourism industry	7

In Fig. 2, the TF-IDF value of each word is calculated for these two sentences. In the first column, all individual/unique words, i.e., the vocabulary of both sentences, are listed. The TF indicates how often a term (w) occurs in a document (d) in relation to the total number of words in the document (d), while the IDF score refers to the logarithmically scaled quotient of the total number of documents (N) in a corpus (D) and the number of documents containing the word (w). The TF-IDF score is the factor of TF and IDF. To simplify the example given in Fig. 2, it should be noted that the preprocessing of the text has been omitted.

$$TF(w, d) = \frac{\text{occurrences of } w \text{ in document } d}{\text{total number of words in document } d} \quad IDF(w, D) = \ln\left(\frac{\text{Total number of documents (N) in corpus D}}{\text{number of documents containing } w}\right) \quad TFIDF(w, d, D) = TF(w, d) \cdot IDF(w, D)$$

Words	TF (for A)	TF (for B)	IDF	TFIDF(A)	TFIDF(B)
Information	1/6	0	$\ln(2/1)=0.69$	0.115	0
is	1/6	0	$\ln(2/1)=0.69$	0.115	0
the	1/6	2/7	$\ln(2/2)=0$	0	0
lifeblood	1/6	0	$\ln(2/1)=0.69$	0.115	0
of	1/6	0	$\ln(2/1)=0.69$	0.115	0
tourism	1/6	1/7	$\ln(2/2)=0$	0	0
internet	0	1/7	$\ln(2/1)=0.69$	0	0.098
has	0	1/7	$\ln(2/1)=0.69$	0	0.098
altered	0	1/7	$\ln(2/1)=0.69$	0	0.098
industry	0	1/7	$\ln(2/1)=0.69$	0	0.098

Fig. 2 TF-IDF

If a document contains numerous sentences, it is advisable to split this document up into individual sentences; the reason being that each sentence has several words indicating the context of the sentence, and each sentence, in turn, points to the context of the whole document. Ultimately, this provides us with a better way of comparing documents and identifying similarities and differences between documents. However, since longer documents have a higher probability of receiving a high score compared to shorter documents (Ramos, 2003), the TF-IDF transformation, despite term weights, also contains a bias. This problem exists because the similarity function between documents only occurs by matching individual terms and their weights (Dong & Liu, 2017).

## 1.4 Word Embeddings

Jurafsky and Martin (2000) define word embedding as the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word. As such, words appearing closer in a vector space are expected to have similar meanings. According to Aggarwal (2018), this can be helpful due to data-centric reasons; for example, when texts consist of very short sections, such as tweets, and the BOW representation simply contains too little information to make meaningful inferences. On the other hand, there could be application-centric reasons as well, such as information extraction, text summarization, or opinion mining, which rely on gaining semantic information from sequences.

When it comes to word embedding, an n-dimensional vector space representation of words that depict both similar words (“hotel” vs. “hostel”) or semantically related words (“restaurant” vs. “food”) are created based on a training corpus that the model learns from. Similar and semantically related words are then positioned close to each other in the vector space.

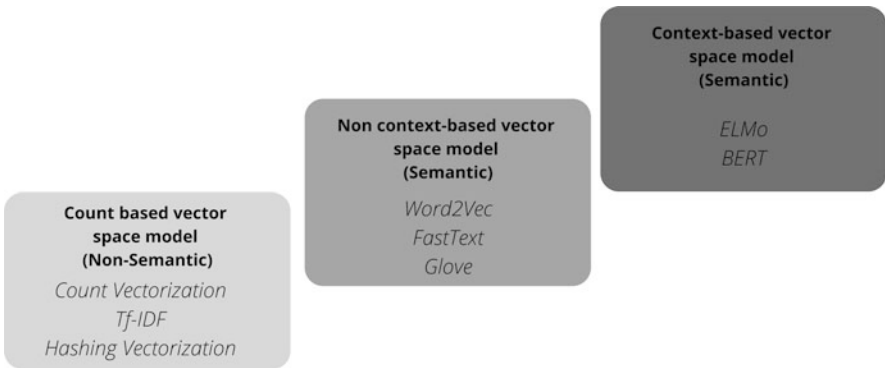
Figure 3 depicts some countries and their capitals within a vector space and also displays the fact that the vectors relate to each other in a similar way - China relates to Beijing in the same way as Russia relates to Moscow. As such, Mikolov, Chen, et al. (2013) showed that algebraic operations can be performed with vectors and, in this sense, calculations such as **France + Berlin – Germany = Paris** can be computed.

The approaches presented precedently have attempted to quantify the meaning of individual terms in one or more documents, with word order being seen as an irrelevant aspect. Yet, there are also situations in which the sequential aspect of the text is rendered important. In this respect, word embeddings are text interpretation techniques that attempt to represent the context of words in the form of fixed-length vectors (Horn et al., 2020). Unfortunately, the two count-based methods that we have discussed so far are unable to capture semantic meaning. Further on in this chapter, however, non-context-based models like Word2vec, FastText, and Glove, which take the semantic meaning of words into account, and ELMo and BERT,





**Fig. 3** Country and capital vectors (PCA Projection). Source: author’s own depiction



**Fig. 4** Types of vector space models

which additionally consider word order and are context-based, will be presented (Fig. 4).

The following example illustrates the meaning of word order and its semantic relevance:

*Salzburg has increasing booking figures compared to Vienna*  
*Vienna has increasing booking figures compared to Salzburg*

Clearly, when looking exclusively at word order, the two sentences have different meanings; yet, since the exact same words are used in both sentences, the representation of count-based and non-context-based models is equivalent (Le & Mikolov, 2014). Nonetheless, non-context-based vector space models can, at least, preserve the local word order.

**Table 1** Word vectors based on different corpora

British-National-Corpus	Google-News	English-Wikipedia	English-Gigaword
hotel 0.73	eatory 0.87	restaurant 0.61	eaterie 0.82
cafe 0.72	restaurant 0.79	eatory 0.56	eatory 0.81
bistro 0.70	restaurants 0.77	hotel 0.55	cafe 0.75
take-away 0.67	diner 0.73	grocery 0.48	diner 0.72
cafés 0.66	steakhouse 0.73	bbq 0.47	bistro 0.71
brasserie 0.66	pizzeria 0.72	hotel 0.46	bakery 0.69

Source: calculated with [vectors.nlpl.eu](http://vectors.nlpl.eu) (This site offers several tools to experiment with word embeddings: <http://vectors.nlpl.eu/explore/embeddings/en/>) based on a Word2vec Skip-gram Model

Besides word embeddings, sentence embedding (ELMo, InferSent, SBERT), where every existing sentence is encoded and/or, lastly, document embedding (Doc2Vec), where whole documents are encoded, also exist. In practice, however, there is no difference between sentence and document embedding.

Enormous developments in the field of NLP have mainly emerged due to the concept of transfer learning using pre-trained models. Yet, training language models is time-consuming and extremely computationally expensive because a model must be trained on a huge corpus, such as the Wikipedia corpus (in English = 6.2 million articles). It is therefore understandable that models trained on different corpora use a different vocabulary, create different word embeddings, and, thus, position words differently within the vector space. As an example, Table 1 shows the six most semantically similar terms to “restaurant,” each based on its corresponding training corpus.

If a text is domain-specific (e.g., financial, medical, legal, or industrial) and differs from the standard corpora that were used to develop pre-trained language models, this can be quite problematic. In other words, using a standard pre-trained model for a domain-specific task might result in insufficient word representations. The solution for such cases involves domain-specific language models that are trained from scratch based on a domain-specific corpus.

While the existence of language models dates back to the 1950s, models and algorithms have continued to develop rapidly, especially over the past 10–15 years. Many language models evolve on the basis of neural networks (Bengio, 2008), and, as of right now, word embeddings seem to be the current status quo. Luckily, the complexity of language, which indeed poses several major challenges, can be solved (more or less) with the help of existing models. A good model should manage and establish the following levels: the lexical approach, which refers to the words or vocabulary of a language; the syntactic approach, which deals with the arrangement of terms and phrases so as to construct well-formed sentences; the semantic approach, which is concerned with the meaning of words; and, finally, the pragmatic approach, which deals with the proximity between words and documents (Bender & Lascarides, 2019; Sieg, 2019a). Figure 5 shows the historical development of word

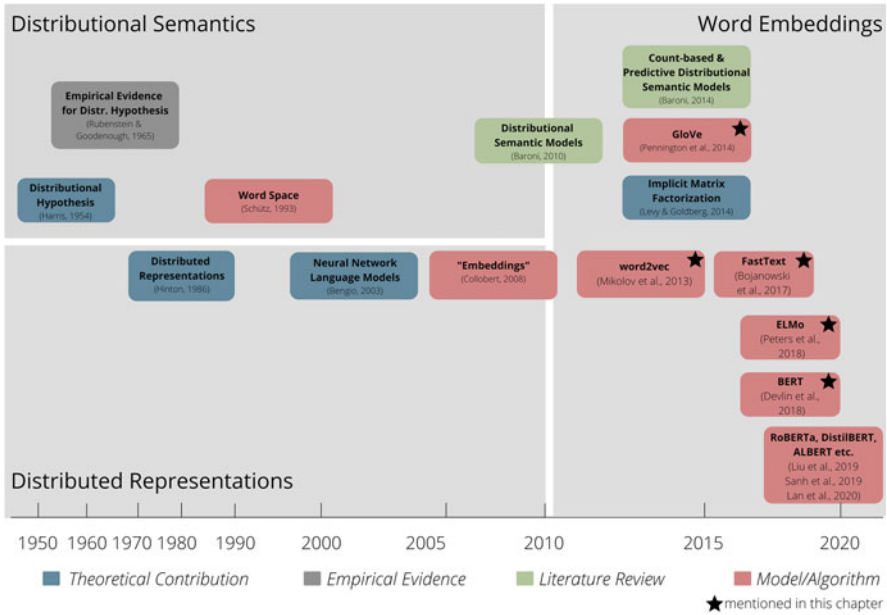


Fig. 5 The history of word Embeddings. Source: adapted from Landthaler (2020)

embeddings, starting from the once independent research fields of distributive semantics and distributed representations.

Nowadays, there are two different approaches available for creating word embeddings, either a counting-based approach, such as the GloVe algorithm, among others, or the prediction-based approach, performed by the Word2vec algorithm. The following paragraphs attempt to present the currently relevant algorithms and models (marked with a star in Fig. 5) in a brief and concise manner.

### 1.4.1 Word2vec

“Word2vec” are one of the most widely used word embedding algorithms that delivered state-of-the-art results in numerous NLP applications (Goldberg & Levy, 2014) until a new generation dawned with the development of Transformers. These prediction-based algorithms were developed by Mikolov, Chen, et al. (2013) at Google and are based on the distributive hypothesis (Sahlgren, 2008), which states that words occurring within the same contexts have similar meanings. These algorithms are based on a three-layer neural network that is able to classify individual components of a text. According to Mikolov, Chen, et al. (2013), the words and their context are embedded in a low-dimensional space (typically 300 dimensions), with a vector being assigned to each word (Kishore, 2018). The created word vectors are then mapped in a vector space in such a way that semantic similarities can be

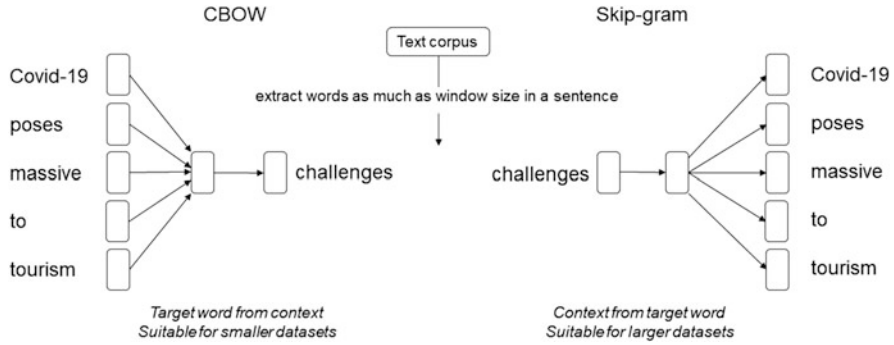


Fig. 6 Word2vec Architectures. Source: adapted from Mikolov, Chen, et al. (2013)

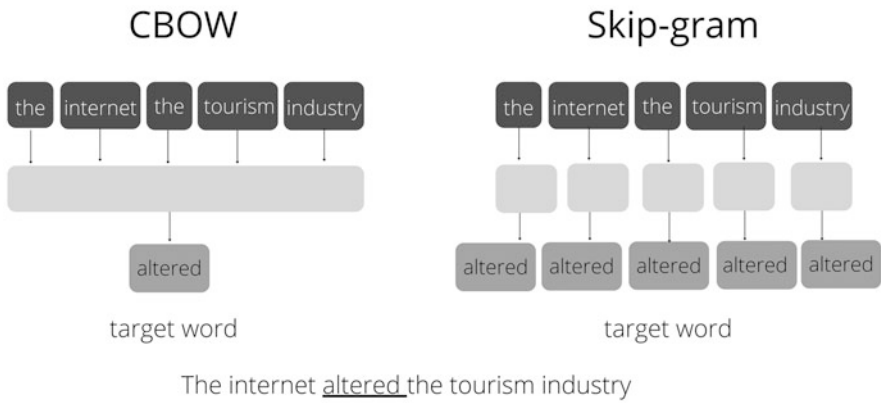


Fig. 7 CBOW versus Skip-gram. Source: author’s own depiction

calculated in a low-dimensional matrix via the cosine distance. The values can range between  $-1$  and  $1$ , although the closer the value is to  $1$ , the higher the similarity (Jatnika et al., 2019; Li, Li, et al., 2018).

Word2vec offers two neural architectures to learn dense and distributed representations of words, the Continuous Bag-of-Words (CBOW) and the Skip-gram Model (Fig. 6).

The basic difference between the two architectures is that a CBOW model combines the distributed representations of the context (the words around the target word) to predict the word in the middle. The exact opposite holds true for the Skip-gram Model in which the context is predicted using the distributed representation of the input word (Fig. 7).

If only a few training data points are available, the Skip-gram model is more suitable because it can also represent rare words and phrases adequately. CBOW, on the other hand, can be trained much faster and is somewhat better for frequently occurring words (Jang et al., 2019). Numerous studies have additionally investigated

**Table 2** Most significant Word2vec hyperparameters

Hyperparameter	Suggested value	Suggested by
<b>Window_size/Window:</b> Since Word2vec predicts whether a word $W_i$ is part of the context of word $W_t$ , a window-size must be specified, which indicates the size of the context around the word $W_t$ . Larger windows represent more topical similarities, while smaller windows more syntactic similarities (Goldberg & Levy, 2014).	<b>5</b>	Goldberg and Levy (2014), Horn et al. (2020)
<b>Vector_size/size:</b> The vector size (embedding size) is the number of dimensions for each word vector. The model quality decreases for vector sizes larger than 300 (Pennington et al., 2014).	<b>300</b>	Zhang et al. (2016), Yuan et al. (2018)
<b>Learning rate (LR):</b> Defines the size of individual steps for optimizing loss function.	<b>0.01–0.1</b>	Horn et al. (2020)
<b>ETA/alpha:</b> Refers to the initial learning rate. The larger the alpha value, the faster the network “learns”. At the same time, however, it also becomes more susceptible to exceeding the minimum.	<b>0.025 (SG)</b> <b>0.05 (CBOW)</b>	Chang et al. (2017), Landthaler (2020), Putra and Khodra (2016)
<b>SG (skip-gram)</b> set value 1 to use SG, set value to 0 to use CBOW.	<b>1</b>	Horn et al. (2020)
<b>Vocab_Min_Count/Min_Count:</b> This value restricts the word embeddings to words that occur at least as often as this value in the training corpus.	<b>100</b>	
<b>HS (hierarchical Softmax)</b> set value 1 to use, set value to 0 to use <b>negative sampling (NS)</b> . When using words with a low frequency for the training corpus, use $HS = 1$	<b>0</b>	Landthaler (2020), Mikolov, Sutskever, et al. (2013)

the importance of hyperparameter tuning in Word2vec. As such, the hyperparameters in Table 2 reveal selected values that have been proven to have a direct effect on training success.<sup>2</sup>

### 1.4.2 Doc2Vec

As discussed, Word2vec allows words to be mapped in a vector space based on their similarity to each other. This not only eliminates the problem of ignoring local word order and context, but it also takes semantics into account. The next approach, “Doc2Vec,” introduced by Le and Mikolov (2014), is a similar unsupervised

<sup>2</sup>For a detailed description of Word2Vec hyperparameters see: <https://radimrehurek.com/gensim/models/word2vec.html#introduction>

algorithm that generates paragraph vectors instead of word vectors. This makes it possible to create a fixed-length vector representation of text pieces, for instance, sentences, paragraphs, or documents. Thus, Doc2Vec allows text entities such as documents to be compared in terms of their semantic similarity. Yet, Li et al. (2019) note that there is a lack of available tourism domain-specific corpora to train domain-specific models. For this reason, Arefieva and Egger (2021) trained a Doc2Vec Model<sup>3</sup> for the travel and tourism industry, based on 3.6 million documents from travel reviews, global sightseeing descriptions, and travel experiences, which can be downloaded and used as a domain-specific model for tourism-related tasks together with the provided Jupyter Notebook.

### 1.4.3 fastText

What can be viewed as a further development of Word2vec is the open-source software “fastText,”<sup>4</sup> which was developed in 2016 by Facebook AI Research. While Word2vec considers each word in a document or corpus to be the smallest unit, fastText dives one level deeper and considers each word as a composition of character n-grams. Thus, the generated vectors are based on the sum of character n-grams (Mikolov et al., 2017). This approach is advantageous when compared to Word2vec since the morphological structure of a word contains important information about its meaning, which is particularly relevant for morphologically rich languages such as German or Turkish (Dündar et al., 2018). Unlike Word2vec, fastText also solves the “out-of-vocabulary” (OOV) problem. When Word2vec is trained, it can only process words that are present in the training data, resulting in unknown terms being completely ignored and, thus, no embedding being created for them. Contrarily, fastText’s sub-word embedding attempts to successfully embed words from the OOV (Anibar, 2021; Kenyon-Dean et al., 2020).

Similarly to Word2vec, the two models CBOW and Skip-gram can be used to compute word representations, and the most significant hyperparameter is the dimension that specifies the size of the vectors. The default value is 100 dimensions, but in practice, this can be successfully extended to 300 dimensions. Since fastText splits words into n-grams, there is a value “minn” that specifies all the minimum substrings contained in a word, and a value “maxn” defining the maximum substrings. Subwords between 3 and 6 characters are recommended for English, but a different value may be more appropriate when it comes to other languages (fastText.cc, 2020). By default, the model is iterated 5 times (epoch). Moreover, as with Word2vec, the learning rate ( $-lr$ ) defaults to 0.05, and values between 0.01 and 1 are recommended.

fastText provides pre-trained vectors for 157 languages, trained on the Common Crawl and Wikipedia corpora, with CBOW in 300 dimensions. Except for the

---

<sup>3</sup>Doc2Vec Model for tourism: [http://datascience-in-tourism.com/models/Tourism\\_Doc2vec.zip](http://datascience-in-tourism.com/models/Tourism_Doc2vec.zip)

<sup>4</sup><https://fastText.cc/>

hierarchical softmax option, the algorithm comes with the same hyperparameters as the Word2vec toolkit. It additionally includes the values for **MinN (3)** and **MaxN (6)**, describing the sizes of n-gram characters that a word is split into, as well as a decay factor modifying the learning rate **LrUpdateRate (100)** (Landthaler, 2020).

#### 1.4.4 GloVe

One particular criticism regarding both Word2vec architectures is that they ignore the different frequencies of some context words while also only capturing the local context rather than the global context (Simov et al., 2017). “GloVe”<sup>5</sup> is a log-bilinear regression model that attempts to overcome these limitations by combining the advantages of count-based and prediction-based methods (Almeida & Xexéo, 2019). From the count-based approach, GloVe takes advantage of the efficiency with which global statistics are captured and combines it with the benefits of meaningful linear substructures from prediction-based models, such as Word2vec. As a result, these word representations outperform the others (Pennington et al., 2014). GloVe can be trained from scratch, or pre-trained GloVe vectors can be loaded from Gensim.<sup>6</sup>

The GloVe algorithm uses **Vector Size**, **Min-Count**, **Window Size**, and **Iterations**, like Word2vec, but also provides some additional hyperparameters as described by Landthaler (2020).

- **Max-Vocab (–)**: This is an alternative hyperparameter to the Min-count, used to set a boundary for vocabulary size.
- **Symmetric (Left and Right context)**: This defines the context window on the left or right side of the pivot token. A symmetric context window is, however, recommended.
- **Distance Weighting (1)**: The GloVe algorithm weighs the distance between two tokens linearly (1) or non-linearly (0).
- **Eta (0.05), Alpha (0.75), and X-Max (100.0)**: These hyperparameters control the learning rate.

#### 1.4.5 ELMo

What all the approaches presented above have in common are that they assign a fixed-vector to words or substrings in the dictionary. However, it can also be that a word has several meanings, rendering the assignment of a fixed vector problematic. “ELMo,” developed by Allen NLP (Peters et al., 2018), is also able to embed these polisemic words correctly since, depending on the context, one and the same word

<sup>5</sup>For an implementation thereof, see: <https://github.com/stanfordnlp/GloVe>

<sup>6</sup><https://radimrehurek.com/gensim/>

may have different word vectors assigned to them. Thus, a word does not receive a unique word-vector but, rather, a vector that is a function of the entire sentence containing that word (Shahbazi et al., 2019). In this way, ELMo takes the entire input sentence into account in order to calculate the word-vector (Ethayarajh, 2019). In addition, the algorithm is also character-based and can, therefore, successfully embed words outside of the normal vocabulary range (Ethayarajh, 2019). ELMo has been shown to outperform alternative approaches in tasks such as named entity recognition or sentiment analysis (Krishna et al., 2018; Liu et al., 2020) and, moreover, is not based on a shallow neural network like Word2vec, fastText, or GloVe, but on a deep neural network architecture (bidirectional LSTM). Due to the complexity of the architecture, further technical background information falls outside the scope of this chapter and is therefore not provided here. Pre-trained models (trained on 1 billion words) are available on Tensorflow Hub<sup>7</sup> or can be downloaded at Allen NLP.<sup>8</sup>

### 1.4.6 BERT

With the development of “BERT,” Google was able to revolutionize the NLP landscape. It can be considered a much deeper neural network than ELMo and contains many more parameters, resulting in a greater representational power (Alsentzer et al., 2019). BERT achieves state-of-the-art results with little fine-tuning and can be used for numerous different NLP tasks, rather than just providing word embeddings as features. As shown in Fig. 5, BERT has laid the foundation for further rapid and diverse developments, including RoBERTa, Distillbert, Albert, XLNet, and Google TransformerXL, among others. The new Transformer architecture attempts to handle sequence-to-sequence tasks, taking not only the meaning of words but also the extensive dependencies between words into account. Vaswani et al. (2017) describe the importance of the attention mechanism in their influential paper as follows: “Self-attention, sometimes called intra-attention, is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence” (Vaswani et al., 2017) (Fig. 8).



**Fig. 8** BERT’s attention mechanism

<sup>7</sup><https://tfhub.dev/google/elmo/2>

<sup>8</sup><https://allennlp.org/elmo>



In this example, the attention mechanism tries to find the correct dependency between the word “it” and “tourist” or “street” and to evaluate it as an attention score. It also looks at the other words in the sequence in order to better understand a particular word. Filler words or unimportant text elements are recognized and disregarded by the algorithm. The disadvantage of this is that attention can only work with a fixed length of text strings. Therefore, existing texts must be chunked into segments of the same size before they can be processed further, which subsequently leads to a fragmentation of the context.

BERT exists as BERT Base (12 transformer layers with 110 million parameters) or as BERT Large (24 transformer layers with 340 million parameters). Further technical details are not presented here. The pre-trained models are based on large datasets, such as the Wikipedia and Books Corpora, and contain over 3 billion English words. As previously mentioned, the training corpus is of enormous importance for transfer learning, and BERT may be too inaccurate for domain-specific NLP tasks. For this reason, different versions of BERT have evolved in which additional domain-specific corpora have been used to train BERT. For example, BioBERT (Lee et al., 2020) exists for biomedical text mining, FinBERT (Chang et al., 2017) for financial communications, or SciBERT (Beltagy et al., 2019) for scientific texts. For tourism-specific NLP tasks, the TourBERT model trained by Arefieva and Egger (2021) is available at Hugging Face (<https://huggingface.co/veroman/TourBERT>).

## 1.5 Visualization of Multidimensional Data

Although word embeddings with, for example, 300 dimensions are referred to as low-dimensional representations, they can no longer be visualized and spatially represented in a human-compatible and understandable form. As described in detail in Chapter “Dimensionality Reduction,” there are numerous methods for dimensionality reduction, which can be implemented to reduce the dataset so as to make visualization possible. The visualization of multidimensional data is usually performed with Python modules like matplotlib or seaborn. At this point, however, one tool, in particular, should also be highlighted. “TensorFlow” offers the Embedding Projector,<sup>9</sup> an online solution (or installed in Python) that allows you to upload vectors and their metadata to visualize them. One can simply decide between the algorithms PCA, t-SNE, and UMAP for dimension reduction (all procedures are described in Chapter “Dimensionality Reduction”), and, thus, in order to examine and understand the data more closely, the Embedding Projector allows multi- and high-dimensional embeddings to be displayed graphically (Fig. 9).

---

<sup>9</sup><https://projector.tensorflow.org/>

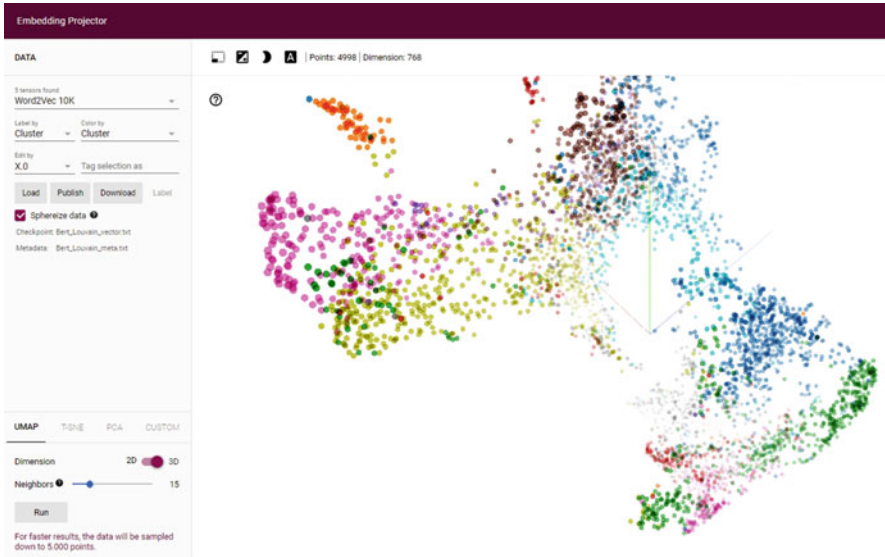


Fig. 9 Visualizing multidimensional word vectors

### 1.6 The Future of Embeddings

It is nearly impossible to keep track of all the recent innovations as new deep-learning models are constantly being introduced, with so-called transformers being viewed as game-changers and as a particular future prospect. The paper “Attention Is All You Need” by Vaswani et al. (2017) presents the sequence-to-sequence (Seq2Seq) architecture of these neural networks, which transform entire sequences, such as sequences of words, into another sequence. Long-Short-Term-Memory (LSTM) models should additionally be emphasized at this point. The attention mechanism evaluates individual sequences, determines their meaning, and either remembers or forgets these parts depending on whether or not the elements are unimportant.

### 1.7 Embeddings in Tourism-Related Research

The availability of large amounts of user-generated content has led to text analysis becoming more critical in tourism (Lee et al., 2020) and machine learning approaches being increasingly applied (Anandarajan et al., 2019). Since many of these methods require a fixed-length vector as an input value, the creation of word representations and embeddings is necessary in order to be able to process text data even further. According to Conneau and Kiela (2018), word embeddings are not

particularly useful in cases where limited training data is available, potentially leading to sparsity and poor vocabulary coverage. Therefore, pre-trained models are often used to perform a wide variety of downstream tasks (Santos et al., 2020). For instance, Chantrapornchai and Tunsakul (2020) used SpaCy and BERT to apply named entity recognition and text classification to perform information extraction tasks on corpora from the tourism industry. Moreover, Li, Li, et al. (2018) proposed a tourism-specific sentiment lexicon for sentiment polarity classification tasks. A survey of text summarization and sentiment analysis tasks performed in the tourism domain was also presented by Premakumara et al. (2019), and, in addition, Li, Zhu, et al. (2018) combined sentiment analysis with topic modeling and an attention mechanism to perform a sentiment classification task on hotel reviews. Lastly, to better investigate tourism spatio-temporal behavior, Han et al. (2019) adapted Word2vec and proposed Tourism2Vec as a destination-tourist embedding model, while, in another study, annotated Instagram images of tourists from Austria were used in a study by (Arefieva et al., 2021) to cluster the destination image. To sum up, Table 3 shows a selection of contributions to the field of tourism, highlighting algorithms and models discussed in this chapter.

**Table 3** Word representations/embeddings in tourism

Algorithm/ Model	Research objective	Authors
TF-IDF	A comparison between text extraction methods in the tourism domain	Kuntarto et al. (2015)
TF-IDF	Hotel review summarization	Nathania et al. (2021)
TF-IDF	Improving object-based opinion mining on tourism product reviews	Afrizal et al. (2019)
TF-IDF; Word2vec	Investigating hotel selection differences among different types of travelers based on online hotel reviews	Wang et al. (2020)
Word2vec	Proposing a systematic approach for integrating traditional research methods into machine learning in text analytics in tourism and hospitality	Abreu
Word2vec	Domain-specific new word detection and word propagation system for sentiment analysis in the tourism domain	Li, Guo, et al. (2018)
Word2vec	Examining Taiwan's rural image	Sun et al. (2020)
Word2vec	Exploring China's 5A global geoparks through online tourism reviews	Luo et al. (2021)
Word2vec	Development of a tour recommendation system using online customer reviews	Hayashi and Yoshida (2019))
Tourism2Vec	Investigating tourism spatio-temporal behavior through the adaption of Word2Vec	Han et al. (2019)
Doc2vec	Clustering annotated Instagram images	Arefieva et al. (2021)
Doc2vec	Automatic tracking of tourism spots for tourists	Mishra et al. (2019)
GloVe	Exploring hotel reviews and responses	Chang et al. (2020)
GloVe	Analysis of racism-related tourism reviews in terms of tendency, semantics, and characteristics	Li et al. (2020)

(continued)

**Table 3** (continued)

Algorithm/ Model	Research objective	Authors
fastText	Ranking online user reviews for tourism based on usefulness	Karanikolas et al. (2020)
fastText	Classifying hashtags of geotagged photos on Instagram	Memarzadeh and Kamandi (2020)
ELMo, GloVe, BERT	Classifying tourism reviews	Gurjar and Gupta (2020)
BERT	Information extraction of tourism-related content	Chantrapornchai and Tunsakul (2020)
BERT	Knowledge extraction on a tourism knowledge graph	Liang
BERT	Sentiment analysis and aspect categorization of hotel reviews	Ray et al. (2021)

## 2 Practical Demonstration

In this practical demonstration, we will look at four different approaches to represent text as feature vectors. To begin with, word vectors are created with the help of Bag-of-Words and TF-IDF, representative of distributional approaches. Although BOW and TF-IDF are relatively simple and represent fundamental approaches, they can still solve tasks such as text classification quite well. With Word2vec, one of the most widely used methods based on a neural network will be presented, and BERT, a state-of-the-art model, will conclude this section. Large text corpora have been omitted for demonstration purposes; instead, simple input sentences will be used. The entire code, including explanatory markdowns, is available as a Jupyter notebook, which can be found in the book's GitHub profile (<https://github.com/DataScience-in-Tourism/>).

### 2.1 BOW

As we already learned, BOW is a fast approach and is easy to implement; yet, at the same time, all words are considered independent of each other, and the meaning of the words gets lost. Furthermore, the BOW approach is only suitable for small datasets.

As a first step, we will load the modules and the stopwords from nltk. Then, we will tokenize the set and filter the stopwords.

```
sentence1=["This hotel is a city hotel and it is located in the city center of Salzburg. "]
```

```

from nltk.corpus import stopwords
#nltk.download('stopwords')
from nltk.tokenize import word_tokenize

text = "this hotel is a city hotel and it is located in the city center of
Salzburg."
text_tokens = word_tokenize(text)

tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]

print(tokens_without_sw)
print(text_tokens)

['city', 'located', 'city', 'center', 'Salzburg', '.']
['this', 'hotel', 'is', 'a', 'city', 'hotel', 'and', 'it', 'is',
'located', 'in', 'the', 'city', 'center', 'of', 'Salzburg', '.']

from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(tokens_without_sw)

print(vectorizer.get_feature_names())

['center', 'city', 'located', 'salzburg']

print(X.toarray())

[[0 1 0 0]
 [0 0 1 0]
 [0 1 0 0]
 [1 0 0 0]
 [0 0 0 1]
 [0 0 0 0]]

```

## 2.2 TF-IDF

TF-IDF is also easy to implement and provides basic metrics for describing the most descriptive terms, which allows us to calculate the similarity between two texts. As input, we will now use two sentences to calculate the vector for each word. These could then be used further to compute the similarity score between the two sentences using the cosine distance.

```

part1 = ""Salzburg has increasing booking figures compared to Vienna""
part2 = ""Vienna has increasing booking figures compared to Salzburg""

# Import module
from sklearn.feature_extraction.text import CountVectorizer

```

```

# Create an instance of CountVectorizer
vectoriser = CountVectorizer(analyzer=preprocess_text)
# Fit to the data and transform to feature matrix
X_train = vectoriser.fit_transform(X_train['speech'])

# Convert sparse matrix to dataframe
X_train = pd.DataFrame.sparse.from_spmatrix(X_train)
# Save mapping on which index refers to which terms
col_map = {v:k for k, v in vectoriser.vocabulary_.items()}
# Rename each column using the mapping
for col in X_train.columns:
    X_train.rename(columns={col: col_map[col]}, inplace=True)
X_train

```

	book	compare	figure	increase	salzburg	vienna
0	1	1	1	1	1	1
1	1	1	1	1	1	1

```

# Import module
from sklearn.feature_extraction.text import TfidfTransformer
# Create an instance of TfidfTransformer
transformer = TfidfTransformer()
# Fit to the data and transform to tf-idf
X_train = pd.DataFrame(transformer.fit_transform(X_train).toarray(),
    columns=X_train.columns)
X_train

```

	book	compare	figure	increase	salzburg	vienna
0	0.408248	0.408248	0.408248	0.408248	0.408248	0.408248
1	0.408248	0.408248	0.408248	0.408248	0.408248	0.408248

### 2.3 Word2vec

In this example, we will take three sentences as input and use the pre-trained Word2vec model to calculate the vectors for three words, followed by determining the similarity between the three terms.

```

import nltk
# import the training model
from gensim.models import Word2vec
# import the scoring metric
from sklearn.metrics.pairwise import cosine_similarity

```

```

# processed data as a list
list_of_strings = ['the internet altered the tourism industry',
 'information is the lifeblood of tourism', 'people like to travel in
summer']

# tokenizing
all_words = [nltk.word_tokenize(sent) for sent in list_of_strings]

# training the model
# since we have a very small corpus, we take a small window and min_count
# size is the embedding size
word2vec = Word2vec(sentences=all_words, min_count=1, size=30,
window=2)

# store the vocabulary
vocabulary = word2vec.wv.vocab

# get the embeddings of the words
v1 = word2vec.wv['tourism']
v2 = word2vec.wv['travel']
v3 = word2vec.wv['internet']

print(v1)

# check similarity of 2 embeddings
cosine_similarity([v1], [v2])

[ 0.01585706 -0.01524498 -0.01337044 -0.01015534 0.00407253
-0.00519751
 0.01376185 -0.01278049 0.01048789 -0.00819942 0.00103017 -0.00641114
 0.01117786 -0.01100589 -0.00404116 -0.00848718 -0.00666839 0.00896648
-0.00434796 -0.00222698 0.00373276 0.00703301 0.01109744 0.00019816
 0.01653573 -0.00737171 -0.00573208 0.00224552 0.01176719
-0.01596778]

array([[0.12688896]], dtype=float32)

```

## 2.4 BERT

Finally, it will be shown how to generate word vectors using BERT. BERT has its own tokenizer, and embeddings are trained with two training tasks. The Classification Task [CLS] determines into which category the input sentence falls, and the Next Sentence Prediction Task [SEP] examines whether the second sentence logically follows the first sentence. The code below is slightly adapted from Dhimi (2020).

```

from pytorch_transformers import BertTokenizer
from pytorch_transformers import BertModel

## Load pretrained model/tokenizer
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained('bert-base-uncased',
output_hidden_states=True)

# Define a new example sentence "
text = "Information is the lifeblood of tourism. The internet has altered
the tourism industry"

# Add the special tokens.
marked_text = "[CLS] " + text + " [SEP] "

# Split the sentence into tokens.
tokenized_text = tokenizer.tokenize(marked_text)

# Map the token strings to their vocabulary indeces.
indexed_tokens = tokenizer.convert_tokens_to_ids(tokenized_text)

# Display the words with their indeces.
for tup in zip(tokenized_text, indexed_tokens):
    print('{:<12} {:>6,}'.format(tup[0], tup[1]))

[CLS] 101
information 2,592
is 2,003
the 1,996
life 2,166
##blood 26,682
of 1,997
tourism 6,813
. 1,012
the 1,996
internet 4,274
has 2,038
altered 8,776
the 1,996
tourism 6,813
industry 3,068
[SEP] 102

import torch

# Convert inputs to PyTorch tensors
tokens_tensor = torch.tensor([indexed_tokens])

# Put the model in "evaluation" mode, meaning feed-forward operation.
model.eval()

```



```

# Run the text through BERT, get the output and collect all of the hidden
states produced
# from all 12 layers.
with torch.no_grad():

    outputs = model(tokens_tensor)

    # can use last hidden state as word embeddings
    last_hidden_state = outputs[0]
    word_embed_1 = last_hidden_state

    # Evaluating the model will return a different number of objects based on
    # how it's configured in the `from_pretrained` call earlier. In this
    case,
    # because we set `output_hidden_states = True`, the third item will be the
    # hidden states from all layers. See the documentation for more details:
    # https://huggingface.co/transformers/model_doc/bert.
    html#bertmodel
    hidden_states = outputs[2]

    # initial embeddings can be taken from 0th layer of hidden states
    word_embed_2 = hidden_states[0]

    # sum of all hidden states
    word_embed_3 = torch.stack(hidden_states).sum(0)

    # sum of second to last layer
    word_embed_4 = torch.stack(hidden_states[2:]).sum(0)

    # sum of last four layer
    word_embed_5 = torch.stack(hidden_states[-4:]).sum(0)

    #concat last four layers
    word_embed_6 = torch.cat([hidden_states[i] for i in [-1,-2,-3,-4]],
dim=-1)

word_embed_5

tensor([[[[ 0.7243, -1.2354, -0.0534, ..., -2.1715, 2.0711, 1.8883],
[-1.4286, 1.9369, 2.2106, ..., -0.3266, 0.6072, -0.6579],
[ 0.4992, 1.5606, 2.6297, ..., -1.0799, 1.9338, 1.1836],
...,
[ 2.8947, 1.6819, 4.9113, ..., -0.4594, 2.3798, -0.0314],
[-0.7187, 1.2479, 0.6565, ..., -2.9043, 3.4472, -1.7060],
[ 0.4703, -0.0677, 0.4450, ..., -0.1805, -0.0650, -0.6184]]]])

```

### Service Section

Word representations and embeddings are central components of NLP. The main idea behind these various algorithms is to transform a text into a number format by creating vectors. This allows calculations to be made using text, and for many ML algorithms, vectors are mandatory as input format. As described in this chapter, there are numerous different approaches that have originated in the past and developed over time, differing greatly in both complexity and performance.

**Main Application Fields:** Word vectors are needed for tasks such as text classification, sentiment analysis, text summarization, knowledge extraction, similarity matching, text clustering, named entity recognition, etc.

**Limitations and Pitfalls:** The choice of an algorithm should always be based on and adapted to the task at hand, and it is important to keep in mind that newer, more powerful approaches do not necessarily lead to better results. It is therefore essential to understand what the strengths and weaknesses of each algorithm are.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-16-Text-Representation-and-Word-Embeddings>

## Further Readings and Other Sources

Manning, Chris (2019) NLP with Deep Learning. Introduction and Word Vectors: <https://www.youtube.com/watch?v=8rXD5-xhemo>

Manning, Chris (2017) Word Vector Representations: Word2vec: <https://www.youtube.com/watch?v=ERibwqs9p38>  
<https://towardsdatascience.com/from-pre-trained-word-embeddings-to-pre-trained-language-models-focus-on-bert-343815627598>

Bornstein, Aron (2018) Beyond Word Embeddings  
<https://towardsdatascience.com/beyond-word-embeddings-part-2-word-vectors-nlp-modeling-from-bow-to-bert-4ebd4711d0ec>

Sieg, Adrien (2019) From Pre-trained Word Embeddings To Pre-trained Language Models — Focus on BERT: <https://towardsdatascience.com/from-pre-trained-word-embeddings-to-pre-trained-language-models-focus-on-bert-343815627598>

## References

- Afrizal, A. D., Rakhmawati, N. A., & Tjahyanto, A. (2019). New filtering scheme based on term weighting to improve object based opinion mining on tourism product reviews. *Procedia Computer Science*, 161, 805–812. <https://doi.org/10.1016/j.procs.2019.11.186>
- Aggarwal, C. C. (2018). *Machine learning for text*. Springer. Retrieved from <https://link.springer.com/content/pdf/10.1007/978-3-319-73531-3.pdf>

- Almeida, F., & Xexéo, G. (2019). *Word Embeddings: A survey*.
- Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Di Jin, Naumann, T., & McDermott, M. B. A. (2019, April 6). Publicly available Clinical BERT Embeddings. Retrieved from <http://arxiv.org/pdf/1904.03323v3>
- Anandarajan, M., Hill, C., & Nolan, T. (2019). *Practical text analytics* (Vol. 2). Springer International Publishing. <https://doi.org/10.1007/978-3-319-95663-3>
- Anibar, S. (2021, April 11). Text classification — From Bag-of-Words to BERT — Part 3 (fastText). Retrieved from <https://medium.com/analytics-vidhya/text-classification-from-bag-of-words-to-bert-part-3-fasttext-8313e7a14fce>
- Arefieva, V., & Egger, R. (2021). Tourism\_Doc2Vec [computer software].
- Arefieva, V., Egger, R., & Yu, J. (2021). A machine learning approach to cluster destination image on Instagram. *Tourism Management*, 85, 104318. <https://doi.org/10.1016/j.tourman.2021.104318>
- Beltagy, I., Lo Kyle, & Cohan, A. (2019). *SciBERT: A pretrained language model for scientific text*. Retrieved from <http://arxiv.org/pdf/1903.10676v3>
- Bender, E. M., & Lascarides, A. (2019). Linguistic fundamentals for natural language processing II: 100 essentials from semantics and pragmatics. *Synthesis Lectures on Human Language Technologies*, 12(3), 1–268. <https://doi.org/10.2200/s00935ed1v02y201907hlt043>
- Bengio, Y. (2008). Neural net language models. *Scholarpedia*, 3(1), 3881. <https://doi.org/10.4249/scholarpedia.3881>
- Chang, Y.-C., Ku, C.-H., & Chen, C.-H. (2020). Using deep learning and visual analytics to explore hotel reviews and responses. *Tourism Management*, 80, 104129. <https://doi.org/10.1016/j.tourman.2020.104129>
- Chang, C.-Y., Lee, S.-J., & Lai, C.-C. (2017). Weighted word2vec based on the distance of words. In *Proceedings of 2017 International Conference on Machine Learning and Cybernetics: Crowne Plaza City center Ningbo, Ningbo, China, 9–12 July 2017*. IEEE. <https://doi.org/10.1109/icmlc.2017.8108974>
- Chantrapornchai, C., & Tunsakul, A. (2020). Information extraction tasks based on BERT and SpaCy on tourism domain. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, 15(1), 108–122. <https://doi.org/10.37936/ecti-cit.2021151.228621>
- Conneau, A., & Kiela, D. (2018, March 14). *SentEval: An evaluation toolkit for universal sentence representations*. Retrieved from <http://arxiv.org/pdf/1803.05449v1>
- Dhami, D. (2020). *Understanding BERT - Word Embeddings*. Retrieved from <https://medium.com/@dhartidhami/understanding-bert-word-embeddings-7dc4d2ea54ca>
- Dong, G., & Liu, H. (Eds.). (2017). *Chapman & Hall/CRC data mining & knowledge discovery series: No. 44. Feature engineering for machine learning and data analytics* (1st ed.). CRC Press/Taylor & Francis Group.
- Duboue, P. (2020). *The art of feature engineering*. Cambridge University Press. <https://doi.org/10.1017/9781108671682>
- Dündar, E. B., Çekiç, T., Deniz, O., & Arslan, S. (2018). A hybrid approach to question-answering for a banking Chatbot on Turkish: Extending keywords with embedding vectors. In A. Fred & J. Filipe (Eds.), *Proceedings: Volume 1, KDIR*. [S. l.]: SCITEPRESS = science and technology publications. <https://doi.org/10.5220/0006925701710177>
- Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 Embeddings.
- FastText.cc (2020, July 18). fastText – Library for efficient text classification and representation learning. Retrieved from <https://fasttext.cc/>
- Goldberg, Y., & Levy, O. (2014). *word2vec Explained: deriving Mikolov et al. 's negative-sampling word-embedding method*.
- Gurjar, O., & Gupta, M. (2020, December 18). *Should I visit this place? Inclusion and exclusion phrase mining from reviews*. Retrieved from <http://arxiv.org/pdf/2012.10226v1>
- Han, Q., Leid, Z., & Margarida Abreu, N. (2019). *tourism2vec*, Available at SSRN 3350125.

- Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2–3), 146–162. <https://doi.org/10.1080/00437956.1954.11659520>
- Hayashi, T., & Yoshida, T. (2019). Development of a tour recommendation system using online customer reviews. In J. Xu, F. L. Cooke, M. Gen, & S. E. Ahmed (Eds.), *Lecture notes on multidisciplinary industrial engineering. Proceedings of the twelfth international conference on management science and engineering management* (pp. 1145–1153). Springer International Publishing. [https://doi.org/10.1007/978-3-319-93351-1\\_90](https://doi.org/10.1007/978-3-319-93351-1_90)
- Horn, N., Erhardt, M. S., Di Stefano, M., Bosten, F., & Buchkremer, R. (2020). Vergleichende Analyse der Word-Embedding-Verfahren Word2Vec und GloVe am Beispiel von Kundenbewertungen eines Online-Versandhändlers. In R. Buchkremer, T. Heupel, & O. Koch (Eds.), *FOM-edition. Künstliche Intelligenz in Wirtschaft & Gesellschaft* (pp. 559–581). Springer Fachmedien Wiesbaden. [https://doi.org/10.1007/978-3-658-29550-9\\_29](https://doi.org/10.1007/978-3-658-29550-9_29)
- Jang, B., Kim, I., & Kim, J. W. (2019). Word2vec convolutional neural networks for classification of news articles and tweets. *PLoS One*, 14(8), e0220976. <https://doi.org/10.1371/journal.pone.0220976>
- Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2Vec model analysis for semantic similarities in English words. *Procedia Computer Science*, 157, 160–167. <https://doi.org/10.1016/j.procs.2019.08.153>
- Jurafsky, D., & Martin, J. H. (2000). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. In *Prentice Hall series in artificial intelligence*. Prentice Hall.
- Karanikolas, N. N., Voulodimos, A., Sgouropoulou, C., Nikolaidou, M., & Gritzalis, S. (Eds.). (2020). *24th Pan-Hellenic Conference on Informatics*. ACM.
- Kenyon-Dean, K., Newell, E., & Cheung, J. C. K. (2020). Deconstructing word embedding algorithms. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 8479–8484). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.emnlp-main.681>
- Kishore, A. (2018). Word2vec. In *Pro machine learning algorithms* (pp. 167–178). Apress.
- Krishna, K., Jyothi, P., & Iyyer, M. (2018). *Revisiting the importance of encoding logic rules in sentiment classification*.
- Kuntarto, G. P., Moechtar, F. L., Santoso, B. I., & Gunawan, I. P. (2015). Comparative study between part-of-speech and statistical methods of text extraction in the tourism domain. In G. Kuntarto, F. Moechtar, B. I. Santoso, & I. P. Gunawan (Eds.), *2015 International Conference on Information Technology Systems and Innovation (ICITSI)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICITSI.2015.7437675>
- Landthaler, J. (2020). *Improving semantic search in the German legal domain with word Embeddings*. Technische Universität München. Retrieved from <https://mediatum.ub.tum.de/1521744>
- Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196) Retrieved from <http://proceedings.mlr.press/v32/le14.html>
- Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). Biobert: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics (Oxford, England)*, 36(4), 1234–1240. <https://doi.org/10.1093/bioinformatics/btz682>
- Li, W., Guo, K., Shi, Y., Zhu, L., & Zheng, Y. (2018). DWWP: Domain-specific new words detection and word propagation system for sentiment analysis in the tourism domain. *Knowledge-Based Systems*, 146, 203–214. <https://doi.org/10.1016/j.knsys.2018.02.004>
- Li, Q., Li, S., Hu, J., Zhang, S., & Hu, J. (2018). Tourism review sentiment classification using a bidirectional recurrent neural network with an attention mechanism and topic-enriched word vectors. *Sustainability*, 10(9), 3313. <https://doi.org/10.3390/su10093313>
- Li, S., Li, G., Law, R., & Paradies, Y. (2020). Racism in tourism reviews. *Tourism Management*, 80, 104100. <https://doi.org/10.1016/j.tourman.2020.104100>

- Li, Q., Li, S., Zhang, S., Hu, J., & Hu, J. (2019). A review of text corpus-based tourism big data mining. *Applied Sciences*, 9(16), 3300. <https://doi.org/10.3390/app9163300>
- Li, W., Zhu, L., Guo, K., Shi, Y., & Zheng, Y. (2018). Build a tourism-specific sentiment lexicon via Word2vec. *Annals of Data Science*, 5(1), 1–7. <https://doi.org/10.1007/s40745-017-0130-3>
- Liu, Y., Che, W., Wang, Y., Zheng, B., Qin, B., & Liu, T. (2020). Deep contextualized word Embeddings for universal dependency parsing. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(1), 1–17. <https://doi.org/10.1145/3326497>
- Luo, Y., He, J., Mou, Y., Wang, J., & Liu, T. (2021). Exploring China's 5A global geoparks through online tourism reviews: A mining model based on machine learning approach. *Tourism Management Perspectives*, 37, 100769. <https://doi.org/10.1016/j.tmp.2020.100769>
- Memarzadeh, M., & Kamandi, A. (2020). Model-based location recommender system using geotagged photos on Instagram. In *2020 6th International Conference on Web Research (ICWR)* (pp. 203–208). IEEE. <https://doi.org/10.1109/ICWR49608.2020.9122274>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, January 16). *Efficient estimation of word representations in vector space*. Retrieved from <http://arxiv.org/pdf/1301.3781v3>
- Mikolov, T., Grave, E., Bojanowski, P., Puhirsch, C., & Joulin, A. (2017). *Advances in pre-training distributed word representations*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality*.
- Mishra, R., Lata, S., Llavoric, R. B., & Srinathand, K. (2019). Automatic tracking of tourism spots for tourists. *SSRN Electronic Journal*. Advance online publication. <https://doi.org/10.2139/ssrn.3462982>
- Nathania, H. G., Siautama, R., Amadea Claire, I. A., & Suhartono, D. (2021). Extractive hotel review summarization based on TF/IDF and adjective-noun pairing by considering annual sentiment trends. *Procedia Computer Science*, 179, 558–565. <https://doi.org/10.1016/j.procs.2021.01.040>
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In M. Alessandro, P. Bo, & D. Walter (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. <https://doi.org/10.3115/v1/D14-1162>
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations*.
- Premakumara, N., Shiranthika, C., Welideniya, P., Bandara, C., Prasad, I., & Sumathipala, S. (2019). Application of summarization and sentiment analysis in the tourism domain. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)* (pp. 1–5). IEEE. <https://doi.org/10.1109/I2CT45611.2019.9033569>
- Putra, Y. A., & Khodra, M. L. (2016). Deep learning and distributional semantic model for Indonesian tweet categorization. In *Proceedings of 2016 International Conference on Data and Software Engineering (ICoDSE): Udayana University, Denpasar, Bali, Indonesia, October 26th–27th 2016*. IEEE. <https://doi.org/10.1109/icodse.2016.7936108>
- Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. In: *Proceedings of the first instructional conference on machine learning*.
- Ray, B., Garain, A., & Sarkar, R. (2021). An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. *Applied Soft Computing*, 98, 106935. <https://doi.org/10.1016/j.asoc.2020.106935>
- Sahlgren, M. (2008). The distributional hypothesis. *Italian Journal of Disability Studies*, 20), 33–53. Retrieved from <https://www.diva-portal.org/smash/get/diva2:1041938/fulltext01.pdf>
- Santos, J., Consoli, B., & Vieira, R. (Eds.) (2020). *Word embedding evaluation in downstream tasks and semantic analogies*.
- Shahbazi, H., Fern, X. Z., Ghaeini, R., Obeidat, R., & Tadepalli, P. (2019). *Entity-aware ELMO: Learning contextual entity representation for entity disambiguation*.
- Sieg, A. (2019a). *FROM Pre-trained Word Embeddings TO Pre-trained Language Models: FROM Static Word Embedding TO Dynamic (Contextualized) Word Embedding*. Retrieved from

<https://towardsdatascience.com/from-pre-trained-word-embeddings-to-pre-trained-language-models-focus-on-bert-343815627598>

- Simov, K., Boytcheva, S., & Osenova, P. (2017). Towards lexical chains for knowledge-graph-based Word Embeddings. In *RANLP 2017 – Recent Advances in Natural Language Processing Meet Deep Learning*. Incoma Ltd. [https://doi.org/10.26615/978-954-452-049-6\\_087](https://doi.org/10.26615/978-954-452-049-6_087)
- Sun, Y., Liang, C., & Chang, C.-C. (2020). Online social construction of Taiwan's rural image: Comparison between Taiwanese self-representation and Chinese perception. *Tourism Management*, 76, 103968. <https://doi.org/10.1016/j.tourman.2019.103968>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). *Attention is all you need*.
- Wang, L., Wang, X., Peng, J., & Wang, J. (2020). The differences in hotel selection among various types of travellers: A comparative analysis with a useful bounded rationality behavioural decision support model. *Tourism Management*, 76, 103961. <https://doi.org/10.1016/j.tourman.2019.103961>
- C. Yuan, J. Wu, H. Li, & L. Wang (2018). Personality recognition based on user generated content. In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*.
- Zhang, X., Lin, P., Chen, S., Cen, H., Wang, J., Huang, Q., . . . Huang, P. (2016). Valence-arousal prediction of Chinese Words with multi-layer corpora. In M. Dong (Ed.), *Proceedings of the 2016 International Conference on Asian Language Processing (IALP): 21–23 November 2016, Tainan, Taiwan*. IEEE. <https://doi.org/10.1109/ialp.2016.7875992>
- Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: Principles and techniques for data scientists* (1st ed.). O'Reilly.

# Sentiment Analysis

## Gaging Opinions of Large Groups



Andrei P. Kirilenko, Luyu Wang, and Svetlana O. Stepchenkova

### Learning Objectives

- Define sentiment analysis goals
- Describe variety of data for sentiment analysis
- Explain main approaches used in text sentiment analysis
- Apply sentiment analysis to tourism domain data
- Indicate popular software used for sentiment analysis

## 1 Introduction

“Sentiment analysis or opinion mining is the computational study of people’s opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes” (Liu & Zhang, 2012, p. 215). The word “sentiment” represents peoples’ feelings such as joy, sadness, anger, and similar. With the explosive popularity of social media leading to the necessity of fast processing of huge volumes of data, e.g., from customer reviews, the traditional methodologies of manual estimation of people’s opinion about topics of products of interest are being increasingly replaced with the automated sentiment analysis (Liu, 2012). Consequently, the scholarship on the methodologies and practices of the computer-based sentiment analysis is in demand and exhibits fast growth. For example, the paper by Bakshi et al. (2016) on using sentiment analysis of tweets to predict changes in stock prices was cited over 10,000 times.

---

A. P. Kirilenko (✉) · L. Wang · S. O. Stepchenkova  
Department of Tourism, Hospitality and Event Management, University of Florida, Gainesville,  
United States  
e-mail: [Andrei.kirilenko@ufl.edu](mailto:Andrei.kirilenko@ufl.edu); [luyuw.93@ufl.edu](mailto:luyuw.93@ufl.edu); [Svetlana.step@ufl.edu](mailto:Svetlana.step@ufl.edu)

Sentiment analysis attempts to measure the emotional valence of the text using a one-dimensional numerical scale from positive to negative sentiment. Depending on the goal of the analysis, it can be applied to the entire document, separate sentences, or the aspects of interest (e.g., different aspects of a consumer product). In addition, the comparative sentiment analysis attempts to compare different sentiment estimates, e.g., “Bonaire is better than Aruba.” Liu (2012) and Hu and Bing (2004) postulate the following essential elements of an expressed opinion: the sentiment (*s*), the opinion target (*g*), the aspect of the target on which the opinion is expressed (*a*), the opinion holders (those who holds the opinion) (*h*), and the time when the opinion is expressed (*t*). The opinion then can be formally written as a 5D vector (*g*, *a*, *s*, *h*, *t*). For example, a hotel review sentence “I hated beddings in the hotel, but liked the view” written on May 5, 2020, by user *cat1967* could be expressed as two vectors (hotel, bed, negative, *cat1967*, 05052020) and (hotel, view, positive, *cat1967*, 05052020). This makes possible a variety of probes such as aspect sentiment analysis, comparative sentiment analysis, evolution of sentiment over time, and so on.

The document-level analysis is the simplest one. Its goal is finding the sentiment of an entire document; thus, it assumes that opinions expressed in an analyzed document are coming from a single person and related to a single event or product (Liu, 2020). This assumption mainly holds for the review-type documents since they are typically authored by one person and express an opinion on one product and for microblogs such as Twitter, but generally, it is too restrictive.

The sentence-level analysis is free of the abovementioned restriction and hence can be applied to many more types of documents. The drawback is that the amount of information used to determine the sentiment is much smaller compared to the document-level classification, making the problem more complex. In addition, while it is generally possible to classify documents into two classes, positive and negative, many sentences contain no sentiment. Hence, instead of the two-class classification of a document, a three-class sentence-level classification is a must. The latter drastically reduces accuracy of the classification algorithms (for comparison, see Ribeiro et al., 2016).

The sentence-level analysis, however, does not assign the sentiment to a specific target. For example, in the sentence “I liked Disneyland but driving there was terrible” the sentiment “terrible” relates only to driving experience, but not to Disneyland. Complicating the analysis, in a sentence “Bonaire diving was excellent” the sentiment “excellent” relates only to the target “diving,” but not to Bonaire as a whole. The approaches aiding in finding the target or an entity of the sentiment are described in detail by Liu (2020).

The sentiment itself may be characterized by its orientation (also called polarity or valence) and intensity. In terms of orientation, the sentiment can be positive or negative, with some researchers also including neutral sentiment. The intensity can be measured using a variety of scales; however, for practical purposes Liu (2020) advises no more than five levels, with two levels frequently being adequate. For example, the sentiment of the statement “I hated beddings” could be  $-4$  intensity on a one-dimensional scale  $[-5, 5]$  due to the presence of word “hate” with negative valence. This method was accepted by the authors of the popular software



SentiStrength (Thelwall et al., 2010) that is based on a large dictionary containing word stems rated according to their sentiment scores (Thelwall, 2016).

The lexicon-based approach is based on a list of words and phrases together with their sentiment orientation and strength; this list is referred to as a sentiment (opinion) lexicon. In the most simplistic implementation, the software performs sentiment analysis by matching each word with the lexicon, thus, extracting the sentiment score. This sentiment score would then be reversed if negation words (such as “not”) are present. In addition, modifiers may weaken or strengthen the sentiment: compare “suspicious person” with “deeply suspicious person” (Polanyi & Zaenen, 2006). The document’s sentiment is then defined as a sum of sentiment scores for all words in the document or as two separate sums of positive and negative sentiments.

A specific problem in the lexicon-based approach is how to generate the sentiment lexicon (for detail, see Liu, 2020). The most straightforward approach is dictionary based. In this approach, a small manually collected seed set of sentiment carrying words with known orientation is used to search a dictionary in order to extract the synonyms, which in turn are used as new seeds. When no new candidate sentiment words are found, the generated list is manually cleaned. The shortcoming of this approach is, however, that the obtained sentiment list is generic and lacks the context. For example, the word “cold” in phrases “cold beer” and “cold person” carry opposite sentiment. This problem is tackled with the corpus-based approach, which applies a variety of approaches to extract sentiment from a collection of representative texts from the field of interest (a corpus). For example, provided that the corpus contains the phrase “he is a cold and greedy person” and that the sentiment of “greedy” is negative, we could conclude that the word “cold” is also negative. A better approach further enhances specificity by including the context in which the adjective “cold” is used.

Essentially, sentiment analysis is a classification problem. The dictionary-based approach is frequently described as an unsupervised classification, that is, classification performed without providing additional external information regarding classification patterns. A competing supervised classification approach is based on machine learning (Liu, 2020). Here, a sample of documents from the same domain is manually classified according to the sentiment expressed (those documents are called “labeled”). This sample is then used to train and validate a machine learning algorithm which is finally applied to the rest of the documents (which are called “unlabeled”). Notice that this approach does not require a list of sentiment carrying words or phrases. Instead, the sentiment is learned by the algorithm during the training process on a pre-processed huge dataset of representative documents. In terms of the machine learning models, many papers apply Naïve Bayes or SVM (Alpaydin, 2020). Recently, a new crop of machine learning models optimized for natural language processing are being successfully used to improve sentiment analysis process; among these models, the most visible is BERT (Bidirectional Representation for Transformers) developed by a Google team. The idea of BERT is to simplify learning process by introducing a new pre-training step which uses a model that is already pre-trained on generic texts. The results can then be fine-tuned

using the field-specific data, resulting in lower training data requirements and faster training process.

The comparative analysis of two approaches typically demonstrates that the machine learning algorithms outperform the lexicon-based ones when the formers are properly trained (Hailong et al., 2014). The lexicon-based methods, however, have distinct advantage by being transparent compared to the “black-box” machine learning algorithms; they require no human- and computer-intensive model training and, therefore, are not sensitive to training quality (Ibid.). The latter point illustrates critical dependency of the machine learning approach on high-quality labeling of a sample of documents by human raters; when a model is pre-trained on documents from a somewhat different domain, the advantage of the machine learning approach disappears (Kirilenko et al., 2018). Even though the lexicon-based methods rely on generic language dictionaries and, hence, are less effective in recognizing emotions in specialized texts such as tweets, they are easier to use and more robust, which frequently make them preferable.

Recently, a new crop of semi-supervised methods has appeared that radically reduces demands of the machine learning methods by injecting the unlabeled documents into the algorithm training process (Van Engelen & Hoos, 2020). These algorithms can be applied to sentiment analysis as well (Lee et al., 2019) and make the machine learning methods more user-friendly. Finally, machine-based methods can be used to improve outcomes of the lexicon-based approach (Zhang et al., 2011).

As a final note, one area closely related to sentiment analysis is emotion detection. While sentiment can be expressed in a single “negative to neutral to positive” dimension, emotion recognition involves classification into multiple emotion classes, for example, Happiness, Sadness, Fear, Disgust, Anger and Surprise (Eckman, 1992). Some researchers experiment with lexicon-based approaches, similar to those used in sentiment analysis; for example, Mohammad and Turney (2013) developed a large multi-language emotion dictionary based on Plutchik (1980) “wheel of emotions.” Nevertheless, it seems that currently emotion detection is better progressing in image and audio analysis (Gajarla & Gupta, 2015), as opposed to text analysis. Indeed, one could imagine the difficulties in recognizing the emotion in a sentence “I work mostly over Zoom nowadays,” which could express happiness, sadness, or be just neutral. For this reason the emoticons and emojis are frequently used in social media to aid conveying writer’s emotions. In a distinct line of research, the emoticons as indicators of emotions are used to successfully train emotion recognition models (Felbo et al., 2017).

In tourism and hospitality, sentiment analysis is an emerging field. The existing reviews found only 26 (Ma et al., 2018), 24 (Alaei et al., 2019), and 68 (Jain & Pamula, 2021) articles; the latter review mostly included papers published in non-tourism journals. The most comprehensive upcoming publication by Mehraliyev et al. (2021) used a systematic search and uncovered 70 articles published in hospitality and tourism journals that used sentiment analysis up to June 2020. The main venues include *Tourism Management*, followed by the *International Journal of Hospitality Management*, *International Journal of*

*Contemporary Hospitality Management* and *Journal of Travel Research*. Notably,  $\frac{1}{4}$  of all articles was published in the first half of 2020, indicating that the interest toward sentiment analysis in tourism scholarship is very recent. Further, the absolute majority of scholarship was focused on market intelligence, with very few papers dealing with other fields such as destination management, strategic management, or social media management. Methodologically, the majority (72%) of the papers used the lexicon approach; half of those papers employed one of the four most popular packages: SentiStrength (Thelwall et al., 2010), AFINN (Nielsen, 2011), LIWC (Pennebaker et al., 2001), or SentiWordNet (Baccianella et al., 2010). Overall, it seems that tourism and hospitality academics only recently discovered sentiment analysis and methodology is mainly based on the most accessible and widely available approaches and packages.

## 2 Theoretical Foundations

The problem of unearthing sentiment in texts was recognized as a distinct aspect of content analysis in the first half of the twentieth century. To differentiate on people's evaluative judgments and affective responses to stimuli (issues, topics, etc.) conveyed in texts, Osgood et al., (1957) identified three aspects of meaning: Evaluation, Potency, and Activity (EPA system) which, taken together, make three-dimensional space where the meaning of each word can be located. Evaluation dimension represents cognitive appraisals on the good-bad continuum. Potency reflects the intensity of the evaluative judgments on the strong-weak continuum. The last dimension, Activity, is represented by the active-passive pair of anchors. The EPA three-factor system was determined through a factor analysis of a large collection of semantic-differential scales and provided the foundation to the attitude research, and numerous studies supported validity of the approach (Heise 1970). Research has also found the stability of EPA structure across various cultures (Osgood 1964; Jakobovits 1966). Not only adjectives but also concepts can be tagged with the meaning along the EPA dimensions. For example, the concept of "war" would score very high on bad, strong, and active dimensions, while the word "baby" would likely score as highly positive, highly weak, and somewhat passive.

Currently, a large amount of works on sentiment analysis involves determining valence, which can be roughly equated with the evaluative EPA dimension; that is, where the sentiment is identified as good/bad; positive/negative or favorable/unfavorable (e.g., Pang and Lee 2008; Liu 2015). Valence, arousal, and dominance are the three dimensions of the Russell's (1980) core affect framework for study of emotions, where valence is associated with pleasure and is also placed on the positive/negative scale. For example, joy is considered as carrying positive valence and, thus, indicates positive sentiment, while anger is indicative of a negative sentiment. The intensity of the emotion can be measured by how far from a neutral point on the positive-negative scale it is located: e.g., wrath is judged as a stronger emotion than anger. This idea that various concepts, descriptors, and affective states

have valence and, thus, can be assigned a score on a positive–negative dimension, lies at the foundation of the automated sentiment analysis (e.g., Pang & Lee, 2008).

### 3 Practical Demonstration

This section provides a brief explanation of the methodology steps, while detailed implementation will be covered in the case study discussed in the next section. Generally, the analysis starts with data cleaning and normalization. The goal of this step is broadly described as increasing data quality and cohesiveness. That may include the following:

- Removal of noise and artifacts such as HTML tags, pictograms, and unwanted characters.
- Tokenization and decapitalization, which breaks textual data into the atomic analysis units, for example, lower-case words.
- Stopword removal: examples include the words like “in,” “of,” “are,” “the,” and “it”; One popular list of stopwords comes from the Natural Language Toolkit ([nltk.org](http://nltk.org)).
- Resolving the attached words such as encountered in hashtags, e.g., “#AwesomeDay”.
- Spelling and grammar correction.
- Resolving negations (e.g., “no good”).
- Part-of-speech (POS) tagging with retaining the words of interest (e.g., adjectives and nouns only).
- Lemmatization or stemming. This step reduces the inflectional and derivational forms of words to a common base form, which in turn increases data cohesiveness. This step is especially important when the machine learning approach is used but may be skipped otherwise.

In no way those steps should be applied without validation. For example, multiple recommendations of a popular tourist guide Mr. Luck or Ms. Grim may dramatically skew distribution of park visitors’ sentiment. Spell checking reviews of Manuel Antonio National Park may replace “Manuel” for “manual.” As a solution, customization of data normalization algorithms is a must.

When the rule-based lexicon approach is used, the next step includes matching the tokens with one of the sentiment or emotion dictionaries, as discussed in Software section. The machine learning approach will include manual processing of a sample of documents classifying them according to expressed emotions. To improve reliability, it is recommended to attract multiple raters. The classified (“labeled”) data then are used to train a classifier such as Naïve Bayes, SVM, or many others, followed by algorithm validation. Finally, the trained algorithm is used to process the unlabeled data.

During the final step, the outcomes are validated, analyzed, and interpreted. The following two sections present a case study demonstrating how those steps are realized in practice.

## 4 Research Case 1: Lexicon-Based Sentiment Analysis<sup>1</sup>

In this section, we demonstrate how the lexicon-based sentiment analysis is used to understand the sentiment expressed by visitors to Manuel Antonio National Park, Costa Rica. Manuel Antonio is the smallest Costa Rica national park (land area 6.8 km<sup>2</sup>) famous for its beaches, wildlife viewing opportunities, beauty of landscapes, and hiking opportunities. Owing to the park's proximity to the national capital (130 km), the park is visited by 150,000 tourists annually, making it the busiest park in the country ([govisitcostarica.com](http://govisitcostarica.com)).

The following case study shows how sentiment analysis was applied to TripAdvisor data to measure the polarity of tourists' reviews covering personal opinions and real travel events. To demonstrate both approaches covered in this article, this section covers both the supervised feature-based machine learning and the rule-based lexicon approaches. The data includes 2700 TripAdvisor park reviews from February 2016 to September 2020 in all languages. All non-English reviews were translated to English by Goggle Cloud Translate. Then, reviews were normalized following the steps discussed in the previous section.

The scope of the project did not allow us to do the manual classification of sentiment reflected in customer reviews as required by the machine learning approach; hence, the decision was made to use the lexicon-based approach. Specifically, two widely used lexicon methods, SentiWordnet and VADER (Valence Aware Dictionary for Sentiment Reasoning) were applied to extract tourists' sentiment about the park. Both methods are based on opinion (sentiment) lexicons which contain the words with positive sentiment such as happy or enjoyable and negative sentiment such as terrible or bad. The sentiment is then defined by mapping the text into the respective lexicon (Al-Shabi, 2020). SentiWordNet (Baccianella et al., 2010) is based on the WordNet ([wordnet.princeton.edu](http://wordnet.princeton.edu)) lexical database of English language (Bonta & Janardhan, 2019). The algorithm assigns each text three scores: objectivity, positivity, and negativity, which range from 0 to 1.

As opposed to SentiWordNet, optimized for texts written in general English, VADER (Valence Aware Dictionary for Sentiment Reasoning) is specifically optimized for microblogs (Gilbert & Hutto, 2014). For each review, Vader generates four sentiment scores: text neutrality, positivity, negativity score, and a compound summary score. The compound score ranges between  $-1$  for the most negative sentiment and 1 for the most positive. A typical sentence with positive sentiment

---

<sup>1</sup>The sentiment analysis code used in this article is publicly available at <https://github.com/luyuwang1993/Sentiment-Analysis/tree/dev-sentiment>

**Table 1** Sentiment analysis validation for SentiWordNet and Vader algorithms

	SentiWordNet	Vader
Accuracy	0.681	0.681
Precision	0.711	0.710
Recall	0.872	0.990
F1 measure	0.783	0.827

would have a compound score greater than 0.05, and a negative sentiment sentence would have a compound score lesser than  $-0.05$ .

In order to validate sentiment predictions, we manually labeled 300 reviews (Table 1). Notice a slightly better performance of VADER; this is to be expected since this algorithm is optimized for social media as opposed to SentiWordNet, which would be preferable for texts written in standard English. Also, notice multiple metrics used for performance evaluation; the data distribution and intended application of the sentiment analysis indicate which metrics is the most useful. In our case, the sentiments were highly imbalanced with many more positive reviews than the negative ones, which makes F1 measure a preferable indicator of model quality. Another good choice of classification quality is Cohen's kappa.

Finally, the reviews carrying negative sentiment were manually processed to find the main topics of dissatisfaction shared by park visitors. The analysis revealed five shared areas of complaint: overcrowding, unprofessional staff, trail condition, opportunistic locals selling parking tickets, and monkeys thieving personal belongings.

## 5 Research Case 2: Machine Learning Sentiment Analysis

In this section, we demonstrate how the machine learning sentiment analysis is used to understand the sentiment expressed by the airline travelers. The dataset<sup>2</sup> represents scraped Twitter data representing six US airlines, subsequently processed by volunteers who classified the tweets into three categories: positive, negative, and neutral, together with the volunteer's confidence score. For this case study, we selected only the tweets with 0.6 or better confidence scores, which removed 1.6% of tweets. Together, that constituted 14,402 airline reviews.

The reviews were pre-processed as described in the How-to section and then vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) metrics (Liu, 2020). Further, the data was split into training and testing sets with 90% of tweets used for training and 10% reserved for testing. Three models, Bernoulli and Multinomial Naïve Bayes and SVM, were trained on the training dataset; then, models were validated on the testing dataset.

<sup>2</sup><https://www.kaggle.com/crowdflower/twitter-airline-sentiment>

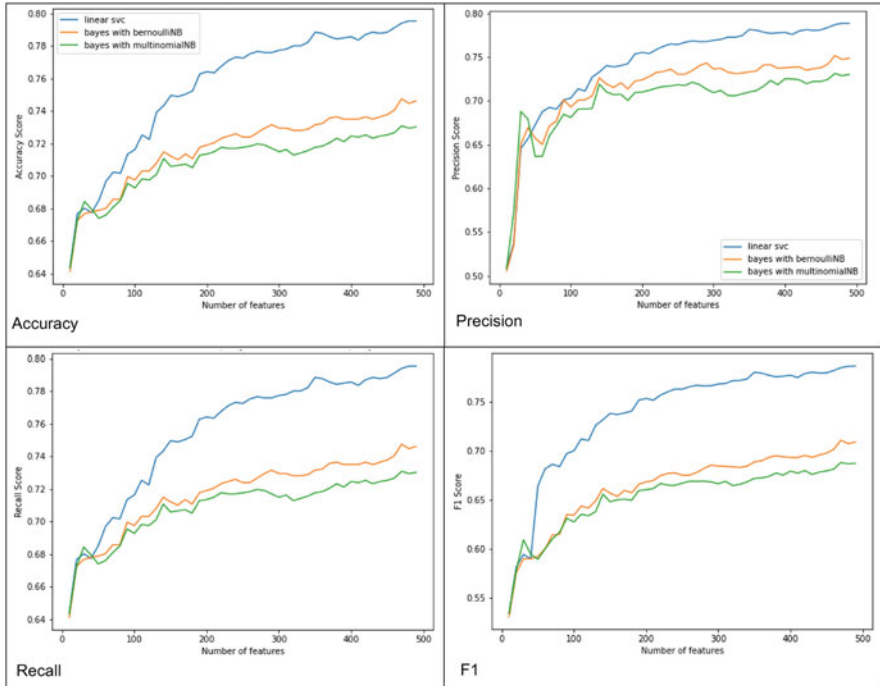


Fig. 1 Performance comparison of algorithms

An important step in the machine learning approach is feature selection. The higher the number of features (e.g., words) selected for model training, the better model predictions on the training data are. However, model performance on the testing dataset follows the bell shape and is reduced when the number of features is too high (“model overfitting”). In addition, a large number of features negatively affect model complexity, requiring expensive computer resources. A performance comparison of models utilizing a progressively increasing number of features (Fig. 1) was used to make decision on the optimal number of features. Notice that after the initial fast growth the performance curve eventually flattens down as more and more features (words) are taken into account by the machine learning algorithm. Hence, the decision was made to limit the number of features at  $N = 300$ .

Similar to the lexicon-based approach, the final decision on satisfactory model implementation was made based on the analysis of multiple indicators of model performance on an independent dataset selected for model testing (Table 2). Similarly to the Manuel Antonio park, the dataset is highly unbalanced: while the natural park reviews are predominantly positive, the airline reviews are predominantly negative. In our case, 61% of the tweets were negative, 22% neutral, and only 17% positive, which makes F1 measure preferable for judging model performance. Overall, SVM model was selected over two other.

**Table 2** Overall performance of all tested algorithms (at 300 features for machine learning approaches)

	Bernoulli NB	Multinomial NB	SVM
Accuracy	72.94%	71.48%	77.72%
Precision	73.63%	70.92%	76.90%
Recall	72.94%	71.48%	77.72%
F1 measure	68.47%	66.65%	76.85%

### Service Section

**Main Application Fields:** Computational study of people’s emotions, attitudes, and opinions, usually expressed in a written text. In tourism, the primary area of application is the analysis of visitors’ reviews of the hotels, destinations, points of interest, and similar.

**Limitations and Pitfalls:** Uncritical use of the computational sentiment analysis without deep understanding of the methods results in unwarranted predictions. For the lexicon-based approach, the dictionary used by the algorithm and the analyzed data much originate from similar domains (e.g., social media). For the machine learning approach, manual classification of a sample of data from same domain is a must. Both approaches require accurate validation on an independent manually classified dataset using multiple performance indices; the latter should account for data distribution and the purpose of analysis.

**Similar Methods and Methods to Combine with:** The sentiment analysis is frequently used together with content analysis and share many approaches and methods.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-17-Sentiment-Analysis>

### Further Readings and Other Sources

Books: “Sentiment Analysis: Mining Opinions, Sentiments, and Emotions” by Bing Liu (2020) is a good introductory text covering all important aspects of computational analysis of sentiment and emotions as well as the most popular algorithmic approaches and major developments in the field.

Videos: “Sentiment Analysis: extracting emotion through machine learning” by Andy Kim. A 10-minutes TED talk introducing sentiment analysis. <https://www.youtube.com/watch?v=n4L5hHFcGVk>

Web sites: [Medium.com](https://medium.com), [towardsdatascience.com](https://towardsdatascience.com), and [KDnuggets.com](https://KDnuggets.com) sites have an excellent set of AI articles including those covering sentiment analysis.



## References

- Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press.
- Alaei, A. R., Becken, S., & Stantic, B. (2019). Sentiment analysis in tourism: Capitalizing on big data. *Journal of Travel Research*, 58(2), 175–191.
- Al-Shabi, M. A. (2020). Evaluating the performance of the most important lexicons used to sentiment analysis and opinions mining. *IJCSNS*, 20(1), 1.
- Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec* (Vol. 10, no. 2010, pp. 2200–2204).
- Bakshi, R. K., Kaur, N., Kaur, R., & Kaur, G. (2016, March). Opinion mining and sentiment analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 452–455). IEEE.
- Bonta, V., & Janardhan, N. K. N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, 8(S2), 1–6.
- Eckman, P. (1992). An argument for basic emotions. *Cognitive Emotions*, 6, 169–200.
- Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.
- Gajarla, V., & Gupta, A. (2015). *Emotion detection and sentiment analysis of images*. Georgia Institute of Technology.
- Gilbert, C. H. E., & Hutto, E. (2014, June). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)* (Vol. 81, p. 82). Available at (20/04/16) <http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf>.
- Hailong, Z., Wenyan, G., & Bo, J. (2014, September). Machine learning and lexicon based methods for sentiment classification: A survey. In *2014 11th web information system and application conference* (pp. 262–265). IEEE.
- Heise, D. R. (1970). The semantic differential and attitude research. *Attitude Measurement*, 235–253.
- Hu, M., & Bing, L. (2004). Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*.
- Jain, P. K., & Pamula, R. (2021). A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. *Computer Science Review*, 41. Available at: <https://arxiv.org/pdf/2008.10282.pdf>
- Jakobovits, L. A. (1966). Comparative psycholinguistics in the study of cultures. *International Journal of Psychology*, 1(1), 15–37.
- Kirilenko, A. P., Stepchenkova, S. O., Kim, H., & Li, X. (2018). Automated sentiment analysis in tourism: Comparison of approaches. *Journal of Travel Research*, 57(8), 1012–1025.
- Lee, V. L. S., Gan, K. H., Tan, T. P., & Abdullah, R. (2019). Semi-supervised learning for sentiment classification using small number of labeled data. *Procedia Computer Science*, 161, 577–584.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1), 1–167.
- Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.
- Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.
- Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415–463). Springer.
- International Journal of Contemporary Hospitality Management, In second review.
- Ma, E., Cheng, M., & Hsiao, A. (2018). Sentiment analysis – A review and agenda for future research in hospitality contexts. *International Journal of Contemporary Hospitality Management*, 30(11), 3287–3308.

- Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3), 436–465.
- Mehraliyev, F., Chan, I. C. C., & Kirilenko, A. P. (2021). Sentiment analysis in hospitality and tourism: A thematic and methodological review. *International Journal of Contemporary Hospitality Management*.
- Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.
- Osgood, C. E. (1964). Semantic differential technique in the comparative study of cultures. *American Anthropologist*, 66(3), 171–200.
- Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). *The measurement of meaning* (No. 47). University of Illinois Press.
- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found Trends Inf Retr*, 2(1–2), 1–135.
- Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001), 2001.
- Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In R. Plutchik & H. Kellerman (Eds.), *Emotion: Theory, research and experience, theories of emotion* (Vol. v. 1, pp. 3–33). Academic Press.
- Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications* (pp. 1–10). Springer.
- Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). Sentibench – a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1), 1–29.
- Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6), 1161–1178.
- Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12), 2544–2558.
- Thelwall, M. (2016). *Sentiment analysis for small and big data. The SAGE handbook of online research methods* (pp. 344–355).
- Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440.
- Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). *Combining lexicon-based and learning-based methods for twitter sentiment analysis*. HP Laboratories, Technical Report HPL-2011, 89.

# Topic Modelling



## Modelling Hidden Semantic Structures in Textual Data

Roman Egger

### Learning Objectives

- Understand the main intuition behind the most relevant topic modelling techniques
- Appreciate the application of topic modelling in the tourism industry
- Identify various hurdles and pitfalls causing bad topic quality
- Apply LDA, NMF, CorEx, Top2Vec, and BERTopic in Python to a dataset about Airbnb experiences

## 1 Introduction and Theoretical Foundations

It has been estimated that approximately 80% of the Internet’s data is available in text format (Anandarajan et al., 2019). In particular, the large amount of user-generated content (UGC) produced every day fosters the increase of unstructured text data (Rossetti et al., 2015), especially in regard to the tourism sector (Li et al., 2019). As a result, the unstructured and unorganised nature of information in the digital sphere complicates the process of traditional quantitative and qualitative analytics. Although topic modelling is based on quantitative methods, it still serves as a means for qualitative research (Evans, 2014; Nikolenko et al., 2017). It can be described as an inductive approach with quantitative measurements and is therefore suitable for descriptive and explorative analyses (Banks et al., 2018).

---

R. Egger (✉)

Salzburg University of Applied Sciences, Innovation and Management in Tourism, Urstein (Puch), Salzburg, Austria

e-mail: [Roman.egger@fh-salzburg.ac.at](mailto:Roman.egger@fh-salzburg.ac.at)

To gain a deeper understanding of a large volume of text, topic modelling has long been considered an effective technique in the lead disciplines of tourism, including marketing and management (Hannigan et al., 2019; Reisenbichler & Reutterer, 2019). Rooted in machine learning and natural language processing (NLP), topic modelling is a method that attempts to efficiently structure large amounts of text, based on co-occurrences of terms in similar texts (Daenekindt & Huisman, 2020). For example, words like “snow”, “ski”, and “snowboard” have a semantic relationship to one another; therefore, in a text document, they could be expected to form a topic called “winter sports”. If only one topic appears in a document, it is called a single-membership model. In most cases, however, mixed-membership models, in which documents consist of a mix of numerous different topics (Maier et al., 2018), are to be found. Thus, this chapter will concentrate exclusively on this variant. In summary, topic modelling refers to a group of methods that attempts to identify topics and their prevalence within a corpus (a collection of documents) in an automated way (Sotomayor & Bellono, 2019).

Many different topic modelling approaches have been developed in recent years, with Latent Dirichlet Allocation (LDA) being the best known and most widely used algorithm (Jockers & Thalken, 2020). Other less known approaches such as Latent Semantic Analysis (LSA) (Landauer et al., 1998), Structural Topic Modelling (STA) (Lindstedt, 2019), Non-Negative Matrix Factorisation (NMF) (Wei et al., 2003), Correlation Explanation (CorEX) (Gallagher et al., 2017), Top2Vec (Angelov, 2020a), or BERTopic (Grootendorst, 2021), amongst others, often outperform and can still be considered exciting alternatives, depending on the requirements. As social media posts are often the data basis for topic modelling projects, in which hotel and restaurant reviews and/or Facebook, Twitter, or Instagram posts are analysed, the fact that these texts are usually short text sections is a particular challenge. The evaluation of topic modelling methods is, therefore, of particular importance. For example, Albalawi et al. (2020) evaluated different topic modelling methods for short-text data like Facebook or Twitter posts.

## 2 Topic Modelling Approaches

As introduced in the previous paragraph, a multitude of different topic modelling approaches exist. In the following section, five methods particularly worth mentioning will be presented and discussed. First, we will turn to the Latent Dirichlet Allocation (LDA), which is typically viewed as the standard approach. Thereafter, Non-Negative Matrix Factorisation (NMF), also popular in the social sciences, will be explained, followed by CorEx. Although much less known, the latter rival algorithm will be discussed in addition to being part of the practical demonstration as it, according to the author’s experience, often achieves better results than LDA and also offers a possibility to define anchor words that enable a “seeded” or “guided” topic modelling process. The fourth approach, Top2Vec, which is a new and, therefore, unfamiliar approach, can be seen as having high potential as well.

Lastly, BERTopic, an approach that makes use of BERT generated word embeddings, will conclude this section.

### 2.1 Latent Dirichlet Allocation (LDA)

LDA, developed by Blei et al. (2003), is currently the most popular topic model algorithm and is implemented in numerous toolkits such as Gensim,<sup>1</sup> Stanford TM toolbox,<sup>2</sup> Machine Learning for Language Toolkit (MALLET<sup>3</sup>) (Albalawi et al., 2020), or Promoss.<sup>4</sup> The basic assumption of the LDA approach is that documents with similar topics also use similar word groups. By searching for groups of words that often co-occur in documents within a certain corpus, latent topics can be found (Evans, 2014). It is also assumed that documents have probability distributions (Dirichlet distribution) over latent topics, and topics have probability distributions over words (Blei, 2012b) (Fig. 1).

On the left side, one can see certain numbers of topics reflecting the distribution of individual terms across the entire document. The assumption is that each

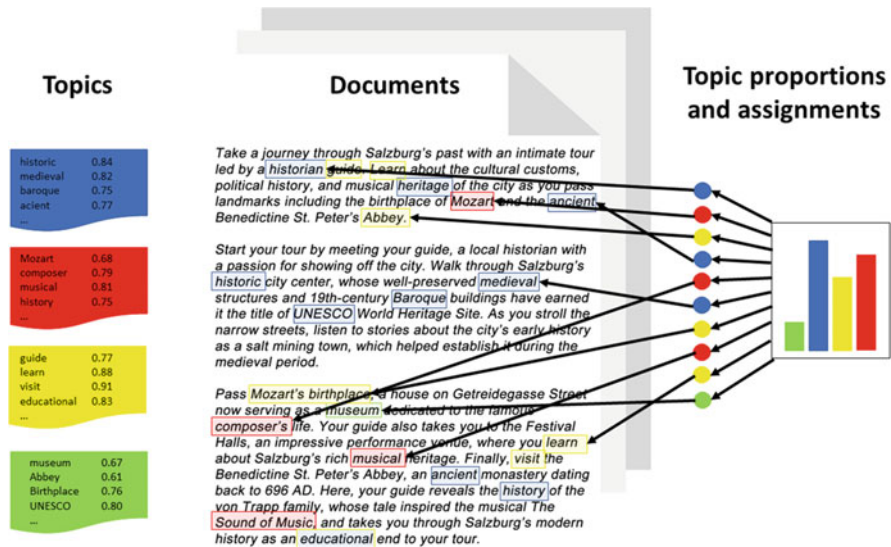


Fig. 1 The intuition behind LDA. Source: Author’s presentation based on Blei (2012b)

<sup>1</sup> <https://radimrehurek.com/gensim/>

<sup>2</sup> <https://nlp.stanford.edu/software/tmt/tmt-0.4/>

<sup>3</sup> <http://mallet.cs.umass.edu/>

<sup>4</sup> <https://github.com/ckling/promoss>

document is generated by choosing a distribution over the topics (topic proportions and assignments/histogram). Thereafter, a topic assignment is chosen for each word (coloured circles), and the word is selected from the corresponding topic. As such, each document is composed of several topics, where each topic consists of words that characterise it and is treated as a “bag-of-words” with no importance attached to word order. The algorithm finds related terms by randomly assigning each word to a topic. This assumes that the number of topics has been defined in advance. In a second, iterative step, each word is reassigned based on the probability of a certain word belonging to a topic as well as the probability of a topic determining the document. The algorithm then calculates these probabilities and reassigns keywords until the model converges.<sup>5</sup>

It is essential to understand that these are not hard clusters like in a K-means clustering. In contrast, the result is an allocation of different topics with different weights per document. For example, a document may be 65% dominated by topic A and only contain 20% of topic B and 15% of topic C. For each topic, the words with the highest probability of belonging to this specific cluster are obtained. It is therefore the user’s task to find a meaningful topic title/umbrella term for the words. The output could appear as follows: document A is assigned to Topic #3, where the words with the highest probability for this topic include “ski, snowboard, sledge, . . . ice skating”. Topic #3 could thus be labelled “winter sports”. Since a topic is formed by finding terms that belong together in terms of probability, it should be noted that the number of terms belonging together largely depends on the size of the bag-of-words. Therefore, it is advisable to divide large text units (e.g. whole papers or even whole books) into segments in order to extract topics that might otherwise disappear.

Due to the popularity of LDA, numerous extensions and adaptations have been developed over time. For instance, some examples include labelled LDA (supervised) (Ramage, Hall, et al., 2009), polylingual LDA topic models (Mimno et al., 2009), LDA models discovering topics over time (Hu et al., 2015; Ungar et al., 2006). Other researchers have extended LDA to a joint sentiment-topic model, enabling simultaneous extraction of both topics and sentiments to improve rating prediction and tourist recommendation applications (Rossetti et al., 2016). Moreover, when constructing the latent topics, a correlated topic model (CTM) was proposed. The CTM incorporates the correlation between words to optimise data fitting, which is particularly helpful when a large amount of corpus-based data is involved (Loureiro et al., 2020). Stemming from this CTM idea, a recent study advanced LDA to a structural topic model (STM) in order to assist hoteliers in identifying various aspects of consumers’ dissatisfaction in online reviews (Hu et al., 2019). Specifically speaking, the STM distinguishes itself from LDA in that it considers document-level metadata when estimating a topic model (Roberts et al.,

---

<sup>5</sup> Assuming we want to extract 8 topics (T) with 1000 tokens (n) from our corpus, the computation of the Bayesian probability formula would be 81,000 (i.e. it would require an enormous amount of computing power). Therefore, Gibbs sampling is used at this point to estimate the result. For a detailed explanation, it is recommended to read Sotomayor and Bellono (2019).

2019). Similarly, research conducted by Park et al. (2018) was one of the first in the hospitality domain to incorporate a more holistic application of topical content analysis through the use of STM.

### 2.1.1 LDA Hyperparameters

The quality of the extracted topics depends not only on the input text but also on the hyperparameters of the LDA model. Depending on which hyperparameters are chosen, the extracted topics can be very general or very specific (Blair et al., 2020). Essentially, there are three crucial hyperparameters that must be defined. The first value is the number of topics (**num\_topics**) to be extracted. LDA generates the number of topics based purely on quantitative calculations, which is why social scientists often rely on a “middle-ground” (Lesnikowski et al., 2019) approach to select the **K** topics. This combines statistical parameters of topic stability with an assessment of interpretability done by experts. It is recommended to run several experiments and ultimately select the number of topics with the highest coherence score (Mohammed & Al-augby, 2020).

The other two hyperparameters are alpha and eta (sometimes also referred to as beta). The **alpha** value controls the prior distribution over the topic weights in each document and, thus, the mix of topics for a document. A higher alpha smooths out the document preference over topics and results in documents with a greater mix of topics. **Eta** (a.k.a. **beta**), on the other hand, controls prior distribution across word weights in each topic. A higher eta smooths out the topic preference across words and leads to topics that are likely to receive more words. It is recommended to perform a grid-search<sup>6</sup> in order to identify the optimal LDA hyperparameter values.

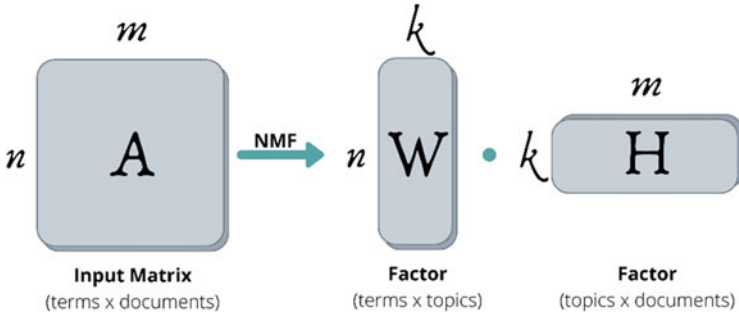
## 2.2 Non-negative Matrix Factorisation (NMF)

Non-negative Matrix Factorisation (NMF) is a decompositional method and belongs to the family of linear algebra algorithms. Therefore, NMF, unlike LDA, is a non-probabilistic algorithm. NMF attempts to decompose a term-document matrix  $A$  (input) as a product of two matrices  $W$  and  $H$  in such a way that both have rank  $k$  (and the total size of  $W$  and  $H$  is significantly smaller than  $A$ ) (Papilloud & Hinneburg, 2018) (Fig. 2). Another condition is that all entries of  $W$  and  $H$  must be non-negative (Lee & Seung, 1999). This helps to prevent difficulties that may arise when interpreting topics with negative entries for certain words (Wang & Zhang, 2013).

Preprocessing of the documents involving steps such as lowercasing, removing stopwords, lemmatising/stemming, RegEx with removing brackets, punctuation,

---

<sup>6</sup>See the Jupyter Notebook at the books Github-Profile for more details.



**Fig. 2** Non-negative matrix factorisation. Source: Adapted from Kuang et al. (2017)

etc., should be performed. Term-document matrix should typically be TF-IDF normalised, and the number of topics  $k$  needs to be defined beforehand.

Let's take three sentences (documents) as an example to visualise NMF's intuition and select  $k = 2$  topics:

*Information is the lifeblood of tourism*  
*The internet has altered the tourism industry*  
*Covid-19 challenges the hospitality industry*

**A - Term Document Matrix**

	Document 1	Document 2	Document 3
Information			
lifeblood			
tourism			
internet			
altered			
industry			
Covid-19			
challenges			
hospitality			

**W - Weights for terms**

	Topic A	Topic B
Information		
lifeblood		
tourism		
internet		
altered		
industry		
Covid-19		
challenges		
hospitality		

**t**

	Document 1	Document 2	Document 3
Topic A			
Topic B			

After decomposition of the original  $n$  tokens by  $k$  topics ( $W$ ) and  $k$  topics by the original  $m$  documents ( $H$ ), we get two non-negative matrices as a result.



### 2.3 *Correlation Explanation (CorEX)*

Gallagher et al. (2017) proposed CorEx as a hierarchical topic modelling approach with minimal domain knowledge. It can be used for large corpora and allows for the integration of domain knowledge by seeding anchor words, making it a semi-supervised approach (Greg Ver Steeg, 2016). The inclusion of anchor words is an optional extension that allows the user to interact with the corpus and to explore the content in an innovative way. When compared with LDA, Gallagher et al. (2017) identified numerous advantages of CorEx and emphasised that higher homogeneity can be reached.

Topics are considered latent factors that may or may not appear in a document. By adding this binary information as input to another layer, hierarchical topic modelling can be achieved. Another advantage over other topic modelling methods is that the number of topics that need to be selected can be easily estimated. Since each topic explains a certain proportion of the total correlation (TC), further latent topics can be added until an increase in the total correlation is only insignificant (Greg Ver Steeg, 2016).

The optional anchoring strategy helps to reveal a topic that, had it been unsupervised, may not have appeared in the first place (Reing et al., 2016). If domain knowledge exists, anchor words such as “Covid”, “hotel”, and “cancellation” can be used to extract a topic around these terms. Anchor words can be assigned to a single topic or to several topics (Cai et al., 2018). Additionally, the strength of anchor words can also be defined; nevertheless, the rule of thumb is that values between 1 and 3 gently nudge a topic, whereas values higher than 5 strongly encourage a topic (Greg Ver Steeg, 2016). With regard to any preprocessing steps for this approach, a binarisation of documents is suggested.

### 2.4 *Top2Vec*

An approach, still largely unknown due to its recent introduction, is Top2Vec. Although new topic modelling algorithms are appearing constantly, this approach along with the corresponding Jupyter notebook was chosen to be presented in this chapter since Top2Vec eliminates some of the weaknesses that arise in the LDA and NMF approaches. Top2Vec uses the joint semantic embedding of documents and words to find topic vectors and does not require the specification of the number of topics to be extracted or any preprocessing. Thus, steps such as the removal of stopwords, stemming, or lemmatisation are rendered unnecessary (Angelov, 2020a). It additionally contains a built-in search function to search for topics by keywords and for documents by topics as well as to find similar words and similar documents (Angelov, 2020b).

Top2Vec uses Doc2Vec or a pre-trained model to find the numeric representation of the given document. By doing so, the semantic relationships between similar

documents and similar words are preserved (Egger, 2022). UMAP is then used to perform a dimensionality reduction and to identify dense areas with HDBSCAN. For these dense areas, a topic vector is ultimately created by taking the arithmetic mean of all the document vectors of a cluster and assigning each document a topic number (Weng, 2020). Finally, the resulting topic vectors are embedded together with the word and document vectors. The distance between the vectors can then be interpreted as semantic similarity (Angelov, 2020a).

## 2.5 *BERTopic*

The last topic model method in this section is BERTopic, which also follows an embedding approach. It uses state-of-the-art sentence-transformers and a class-based version of TF-IDF to create interpretable topic clusters (Grootendorst, 2020). BERTopic can be used for any language if an embedding model already exists for it.<sup>7</sup> As a result, this even allows for the performance of multilingual topic modelling in cases where the documents contain several different languages. Overall, the possibility of using self-created embedding models instead of pre-trained embeddings may prove to be particularly useful. In numerous experiments, it seems that tourism domain-specific models provide much more accurate results than publicly-available pre-trained models.<sup>8</sup> According to Grootendorst (2021), no preprocessing is needed unless the used transformer model requires it. The algorithm allows the manual or automatic merging of similar topics according to the cosine similarity between the vectors. As with other topic modelling approaches, topics are shown as a set of words, and the researcher can define the n-gram range. BERTopic generates a value for the topics and a probability score as output, demonstrating the likelihood that a document belongs to any specific topic. Additionally, BERTopic comes with its own topic visualisation, similar to LDAvis.

## 3 Topic Modelling Limitations and Challenges

We have now learned about different topic modelling approaches, each of which comes with its own set of advantages and disadvantages and is suitable for particular application scenarios. However, for all approaches, it is up to the researcher to interpret the topic clusters and make sense of them (Mimno et al., 2011). Most topic modelling techniques require domain knowledge to correctly interpret the connections and relationships between the extracted terms of a latent topic. Although the identified topics make sense from a mathematical perspective, they often do not

---

<sup>7</sup>Pretrained models available: [https://www.sbert.net/docs/pretrained\\_models.html](https://www.sbert.net/docs/pretrained_models.html)

<sup>8</sup>A tourism-domain specific embedding is available at:

match human judgement (Chang et al., 2009; Nikolenko et al., 2017). As such, the poor quality of identified topics seems to stunt the acceptance of statistical topic models outside the machine learning community (David Mimno et al., 2011).

Cai et al. (2018) mention the following reasons as to why extracted topics may appear suspicious:

1. Two or more themes are merged into one theme.
2. Two themes are extracted that, to humans, look like duplicates.
3. Extracted keywords of topics do not seem to make sense.
4. Topics contain too many generic terms.
5. Topics that are based on seemingly unrelated terms are extracted.
6. Topics do not match human judgement.
7. Topics appear irrelevant.
8. The relationship between topics and documents is not apparent.
9. Several similar topics are extracted.

However, in most cases, the poor quality of topic models does not lie in the algorithms themselves but, rather, in an insufficient understanding of how to apply them. In many studies, there is insufficient documentation on how the input text data was preprocessed, how hyperparameters were defined, how models were evaluated, how reliability and interpretability were increased, and how the results were validated. All of these suggest that default values are used quite often and results are published without further processing. Yet, this is problematic as DiMaggio et al. (2013) note that “producing an interpretable solution is the beginning, not the end, of an analysis” (p. 586).

For most algorithms, preprocessing data is a prerequisite to achieving good results, and the saying “garbage in–garbage out” most certainly applies. Lim and Buntine (2014) stress the importance of preprocessing as the basis of a successful topic modelling process in which a number of standard steps and also the order of performing them is emphasised (Denny & Spirling, 2018). The standard steps include language recognition, if necessary, and correction of misspellings, tokenisation, lowercasing of words, removing non-informative features such as punctuation and special characters like emojis, numbers, HTML-codes, and URLs, filtering out stopwords and highly frequent and infrequent terms, and stemming or lemmatising (Kadhim et al., 2014; Maier et al., 2018). For further information on these steps, refer to Chapter 15 (“Introduction: Natural Language Processing”).

When it comes to social media posts in particular, the underlying text elements may be too short to achieve meaningful results. In this case, it is advisable to either aggregate texts (Liangjie & Davison, 2010; Zhao et al., 2011) or apply special algorithms for short-text analysis (Qiu & Shen, 2017; Vo & Ock, 2015). Ultimately, the quality of the topics also depends on hyperparameter tuning. Depending on the modelling approach, numerous settings can be adjusted, which, in turn, greatly impact the results. For example, LDA uses random initialisation and stochastic inference, making the results non-deterministic, which is why Maier et al. (2018) recommend always using reliability checks to verify the robustness of the results.

One of the biggest challenges in topic modelling, however, is to determine the optimal number of topics. This important step is often only estimated or determined by trial and error without paying much attention to quantitative key figures or qualitative insights. It has been observed that researchers tend to choose a larger number of topics in order to be able to make fine-grained delimitations. Yet, as a result, this may lead to entities that can no longer be meaningfully distinguished because they are too similar (Grimmer & Stewart, 2013). Too few topics, on the other hand, may lead to entities that are too broad (Evans, 2014), and revealed insights can no longer be isolated and interpreted. As David Mimno et al. (2011) show in their study, there is a strong correlation between the number of topics and the judgement of nonsense topics by domain experts. However, it is also possible to determine key figures that allow an optimal number of topics to be derived. For example, Greene et al. (2014) developed a stability analysis<sup>9</sup> tool for topic models, which measures the robustness to perturbations in the data so as to identify an appropriate number of topics.

### ***3.1 Evaluating and Interpreting Topics***

Interpreting and evaluating topic models can often be challenging (Wallach et al., 2009) and frustrating since, with unsupervised approaches like LDA, the interpretability of the results is not always guaranteed (Röder et al., 2015). The result is a list of words ranked by relevance, and naming the topic based on these words can indeed be difficult (Hindle et al., 2013).

There are numerous approaches to measuring the quality of topic models. The evaluation should correctly assess the generalisation capability of a topic model, being computationally efficient and independent of a specific use case (Wallach et al., 2009). Since there is no gold standard for the data, the question of how to quantify the quality of topics becomes the first challenge. Statistical topic models are mostly evaluated either by extrinsic methods or quantitative intrinsic methods (Mimno et al., 2011; Wallach et al., 2009). Extrinsic methods test for associations between topics using data that was not used while learning the topic model. This can include annotations or metadata of documents. If a connection between these external data points and the topics exists, it should be possible to interpret them meaningfully (Papilloud & Hinneburg, 2018). On the other hand, for intrinsic evaluation, the existing text corpus is used to make a quantitative statement about the quality of the topics based on word probabilities estimated during the learning of the model (Blair et al., 2020).

---

<sup>9</sup><https://github.com/derekgreene/topic-stability>

An established method for latent variable models like LDA is the calculation of topic coherence, where the notion coherent refers to a set of statements that support each other (Röder et al., 2015). Thus, a coherent set of facts can be interpreted in a context that includes all or most of the facts. In very simplified terms, topic coherence measures how often the top words co-occur in a topic

$$\text{Topic Coherence (T)} = \sum_{w_1, w_2} \log \frac{\text{docs with both words} + 1}{\text{docs with word } w_2}$$

Since the overall goal should be to generate a well interpretable topic result, it is recommended to create several candidate models (Maier et al., 2018) by changing the hyperparameters (how to perform this is shown in the Jupyter notebook for this chapter). The comparison of the topic coherence between the different models already provides a good hint as to which model to use. Additionally, word intrusion can be used as a form of human judgement in which a person is shown high probability words for a specific topic without knowing the word list of the topic (Fig. 3). The assumption is that a topic model is good if the person can correctly guess which word should not belong. The model precision is then calculated by dividing the number of the true intruder’s correct guesses by all guesses. The higher the score, the better the model (Chang et al., 2009).

Intertopic distance maps are an additional option that can be used to support a visual decision. These can be created as scatterplots (after performing dimensionality reduction); in the case of LDA, the Python module LDAvis (Sievert & Shirley, 2014) is available for visualising topics. Based on multidimensional scaling of the topics, their size and the most relevant keywords per topic are presented. This visualisation especially helps in identifying overlapping topics and getting a better feeling in terms of selecting the optimal number of K topics (Fig. 4).

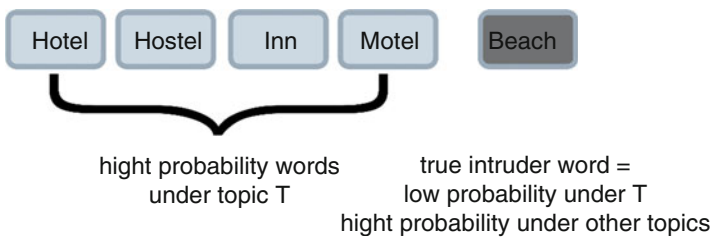


Fig. 3 Word intrusion. Source: Adapted from Dietz (2016)

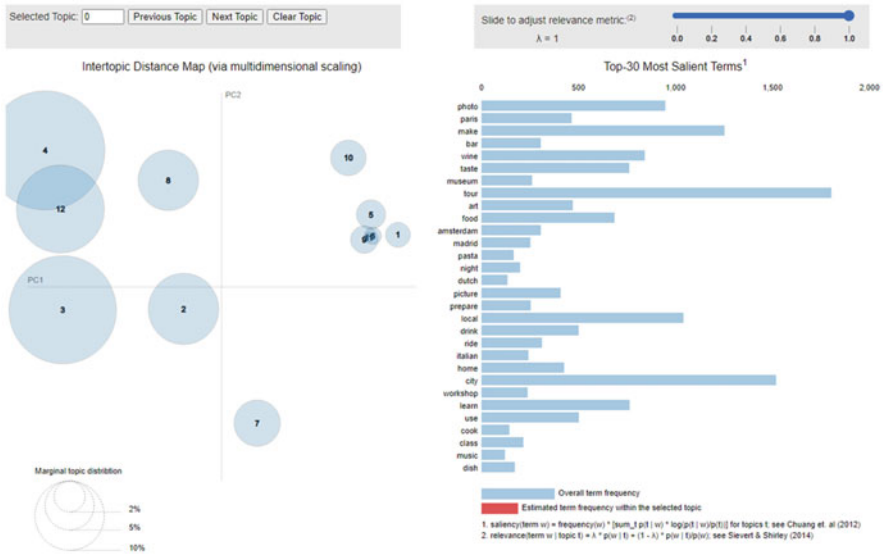


Fig. 4 LDAvis

## 4 Topic Modelling in Tourism Studies

In a nutshell, topic models assume that meanings are relational (Saussure, 1959), and each document is composed of diverse topics that comprise a collection of highly related words. In a practicable sense, topic modelling is often applied in exploratory studies (Guo et al., 2017) as it can disclose tourism experiences from a bottom-up approach, thereby providing insights that are often neglected by marketers (Shafqat & Byun, 2020). When it comes to tourism-related studies, user-generated content from various social media channels is mainly analysed in order to gain insights into the users’ characteristics, attitudes, and opinions (Cai et al., 2018).

Compared to other methods of social research, the literature surrounding topic modelling in the field of tourism is quite limited; the main reasons being that there is a learning gap when it comes to effectively applying topic modelling to tourism research, especially regarding statistical and mathematical basics as well as programming skills (Papilloud & Hinneburg, 2018) in coding languages like Python or R. Only recently has topic modelling been integrated into more studies, most likely because more and more tools that do not require programming skills are becoming readily available on the market. However, such “Topic Modelling Toolkits”, which can be applied without any programming knowledge, often have a disadvantage as they contain insufficient data preprocessing and lack hyperparameter tuning, ultimately leading to, as previously noted, bad quality results.

Existing studies suggest that LDA is the most widely accepted approach (Calheiros et al., 2017; Park et al., 2018) because of its capability to assign a

probability composition of the document to a latent theme through Bayesian inference (Blei, 2012a; Park et al., 2018). This suggests the iterative process of topic models, where documents are first assigned via random probability, but the performance of the algorithm becomes more accurate once more data has been processed (Vu et al., 2019). However, as can be seen from existing research with LDA methods, a strong emphasis has been placed on extracting the perceptions of tourism experiences based on online reviews and comments (Bi et al., 2019; Dickinger et al., 2017; Kim et al., 2019; Taecharunroj & Mathayomchan, 2019). One study (Vu et al., 2019) even took this further by using LDA to uncover tourist activity preferences in the context of travel itineraries. On the other hand, Wang et al. (2020) adopted LDA to uncover tourists’ spatial and psychological involvement in multiphasic travel stages, while Shafqat and Byun (2020) used LDA and sentiment analysis to recommend under-emphasised locations. The following table lists some of the most recent research projects in the tourism domain, highlighting their research objectives, the applied methodology and tools/software used for topic modelling (Table 1).

**Table 1** Tourism-related research projects using topic modelling

Name	Year	Title	Objectives	Methodology (Software)
Egger et al.	2022	Topic modeling of tourists dining experiences based on the GLOBE model	Identifying the dining preferences per cultural dimension	<b>LDA</b> , MDS, FastText (Python, Orange)
Yu & Egger	2021	Tourist experiences at overcrowded attractions: A text analytics approach	To explore the perception and feelings of tourists when visiting overcrowded attractions	<b>LDA</b> , sentiment analysis (Python, Orange)
Egger & Yu	2021	Identifying hidden semantic structures in Instagram data: a topic modelling comparison	Evaluating the effectiveness of different topic modelling algorithms	<b>LDA</b> , NMF, <b>CorEX</b> (Python)
Luo, He, Mou, Wang, & Liu,	2021	Exploring China’s 5A global geoparks through online tourism reviews: A mining model based on machine learning approach	To provide valuable suggestions for managers by increasing the understanding of the psychological cognition of tourists	<b>LDA</b> ; SVM; IPA (Python)
Shafqat & Byun	2020	A recommendation mechanism for under-emphasized tourist spots using topic modelling and sentiment analysis	To help the tourism industry in designing effective promotional activities for under-emphasised locations	<b>LDA</b> ; SVM; cross mappings (Python)
Wang, Li, Wu, & Wang	2020	Tourism destination image based on tourism user generated content on internet	To study tourists’ spatial and psychological involvement reflected through a tourism destination image	<b>LDA</b> ; SNA (Unknown)

(continued)

**Table 1** (continued)

Name	Year	Title	Objectives	Methodology (Software)
Zou	2020	National park entrance fee increase: A conceptual framework	To understand the public acceptance and opposition of a fee increase in the context of public park tourism	<b>LDA</b> ( <i>R</i> )
Sun, Liang, & Chang	2020	Online social construction of Taiwan's rural image: Comparison between Taiwanese self-representation and Chinese perception	To determine how the objective discourse concerning Taiwanese rurality represented on online media is constructed and maintained	<b>Word embeddings</b> (Word2Vec) Keyword analysis; correspondence analysis; ( <i>R</i> )
Wen, Park, Tao, Chae, Li, & Kwon	2020	Exploring user-generated content related to dining experiences of consumers with food allergies	To explore factors influencing perceptions of consumers with food allergies towards restaurants when accommodating allergen-free requests	<b>Structural topic model</b> ( <i>Python</i> )
Han, Zejnilovic, & Novais	2019	Tourism2vec: An adaptation of Word2Vec to investigate tourism spatio-temporal behaviour	To propose tourism2vec for the investigation of tourism spatio-temporal behaviour	<b>Word embedding</b> ( <i>Python</i> )
Hu, Zhang, Gao, & Bose	2019	What do hotel customers complain about? Text analysis using structural topic model	To identify the antecedents of hotel customers' dissatisfaction across different classes of hotels	<b>Structural topic model</b> ( <i>R</i> )
Kim, Park, Barr, & Yun	2019	Tourists' shifting perceptions of UNESCO heritage sites: Lessons from Jeju Island-South Korea	To analyse the shifting perceptions of international tourists to Jeju Island	<b>LDA</b> ( <i>R</i> )
Taecharungroj & Mathayomchan	2019	Analyzing TripAdvisor reviews of tourist attractions in Phuket, Thailand	To develop a methodology that can analyse online reviews using machine learning techniques for tourism practitioners to improve their attractions	<b>LDA</b> ; naïve Bayes modelling ( <i>KNIME</i> )
Vu, Li, & Law	2019	Discovering implicit activity preferences in travel itineraries by topic modelling	To introduce a framework for travel itinerary analysis that can reveal the underlying activity preferences of tourists	<b>LDA</b> ( <i>Unknown</i> )
Hayashi	2019	Applying the document vector model to tour recommendation	To propose a system for recommending tours and their customer reviews	<b>Word embedding</b> (Word2Vec) ( <i>Python</i> )

(continued)



**Table 1** (continued)

Name	Year	Title	Objectives	Methodology (Software)
Huang, Wang, Yang, & Xu	2018	Topic mining of tourist attractions based on a seasonal context aware LDA model	To detect the representative and comprehensive attributes corresponding to various seasonal contexts for each attraction	<b>Season topic model LDA</b> (Unknown)
Li, Li, Hu, Zhang, & Hu	2018	Tourism review sentiment classification using a bidirectional recurrent neural network with an attention mechanism and topic-enriched word vectors	To analyse the emotions, preferences, feelings, and opinions expressed by visitors based on hotel review comments	<b>Lda2vec;</b> Bidirectional gated recurrent unit neural network model (Python)
Li, Zhu, Guo, Shi, & Zheng	2018	Build a tourism-specific sentiment lexicon via Word2Vec	To mine useful knowledge which can help tourism websites make decisions and improve their travel products	<b>Word embedding</b> (Word2Vec) (HowNet)
Calheiros, Moro, & Rita	2017	Sentiment classification of consumer-generated online reviews using topic modelling	To gather relevant topics that characterise a given hospitality issue through a sentiment	<b>LDA;</b> Sentiment analysis; (R)
Xu & Li	2016	The antecedents of customer satisfaction and dissatisfaction toward various types of hotels: A text mining approach	To discover and compare the determinants of customer satisfaction and dissatisfaction towards different hotels	<b>LSA</b> (RapidMiner)

## 5 Topic Model Toolkits and Software Solutions

Although an overview of available topic modelling solutions is given in Chapter 26 (Software & Tools), some further information is given here. For those working with Python, *Gensim*, presented by Rehurek and Sojka (2010) is a widely used and accepted topic modelling toolkit. Alternatively, *MALLET* (McCallum, 2002) or *tmtoolkit* (Konrad, 2017) can be used. For users who prefer R, *MALLET* is also available along with numerous packages such as *topicmodels* (Hornik & Grün, 2011) or *LDA* (Chang et al., 2009). A graphical user interface is provided by *Orange 3* offering the “Topic Modelling” widget in the text mining module. Three algorithms, namely LSI (Latent Semantic Indexing), LDA, and HDP (Hierarchical Dirichlet Process) can be used here. Unfortunately, for LDA and LSI, only the number of topics can be defined, which leads to obtaining results that must be accepted without having a closer look at the quality criteria. *Rapidminer* also offers topic modelling with the *LDA operator*. Stand-alone solutions include the *Stanford*

*Topic Modelling Toolbox* (Ramage, Rosen, et al., 2009), the *Topic Modelling Tool* by Scott Enderle or *Serendip*, which is a system for visually exploring topic models (Alexander et al., 2014), or *Topics Explorer* (Simmler et al., 2019). Alternatively, browser-based solutions like *jsLDA* (David Mimno, 2013), or *topix.io* can be used. At this point, however, it must be pointed out, once again, that solely the tools that allow for the preprocessing of input texts as well as extensive control of the hyperparameters and fine-tuning of the results are to be taken seriously. It is therefore recommended to perform topic modelling either in Python or R.

## 6 Practical Demonstration

In this section, we will complete a walkthrough with a dataset from Airbnb, applying LDA (using Gensim) and CorEX as two distinct topic modelling approaches. This dataset was crawled by the author and contains 2890 descriptions of Airbnb experiences from the following European cities: Amsterdam, Athens, Berlin, Brussels, Copenhagen, Helsinki, London, Madrid, Oslo, Paris, Prague, Rome, Stockholm, Vienna, and Warsaw. The complete code and the dataset are available as a Jupyter notebook, together with an NMF exercise. For Top2Vec and BERTopic, two separate Jupyter notebooks are provided in the book’s Github-profile.

### 6.1 LDA: Data Preparation & Preprocessing

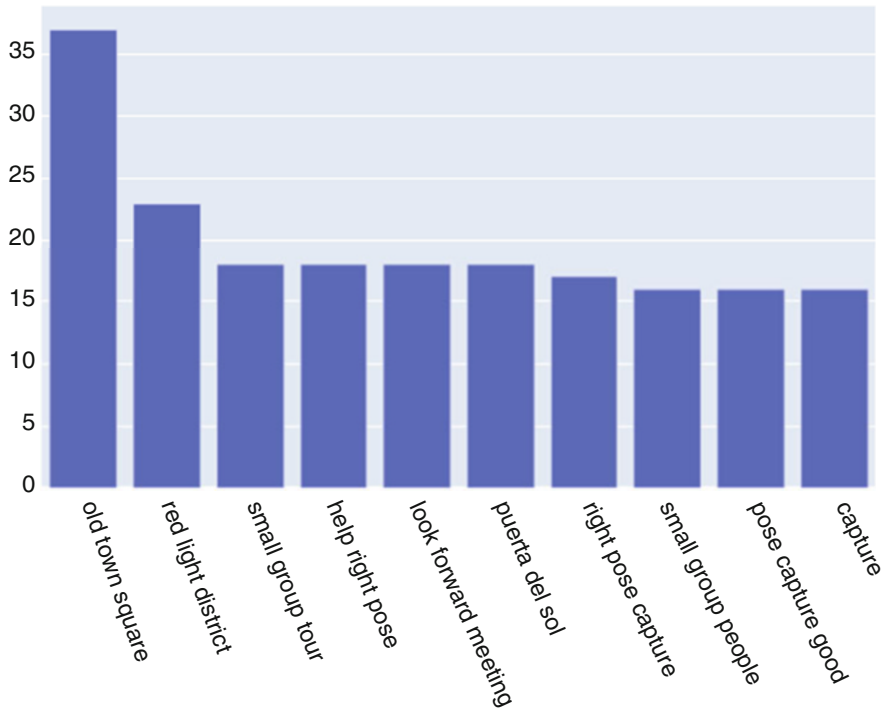
Since we start with LDA in our example, we need to preprocess the text accordingly. As already mentioned, however, not every method requires this exact form of preprocessing. To prepare the text data for the application of LDA, the text is first converted to lowercase. It is then followed by some regex operations, the removal of punctuation, special characters and numbers (note: depending on the objective, this may or may not be necessary), the removal of stopwords, and tokenisation and lemmatisation of the text. These steps are described in detail in Chapter 15 (Introduction: Natural Language Processing) and are therefore disregarded here.

If you use the Gensim library, you can use the integrated preprocessing module (`preprocess_string`), which strips punctuation, removes stopwords, cleans HTML tags and non-alphabetical characters, stems the text, and much more. Table 2 shows an excerpt of the original text and what the text looks like after preprocessing or lemmatising.

To get a first impression of the dataset, generating a wordcloud for better visualisation is recommended and worthwhile (Fig. 5).

In order to avoid looking at individual tokens in isolation and, consequently, drawing the wrong conclusions in the analysis, one should also have a look at bi- and tri-grams. In this way, instead of the three words “red”, “light”, and “district”, the





**Fig. 6** Histogram of the top 20 tri-grams

```
common_words = get_top_n_trigram(df['lemmatized'], 20)
df_toptri = pd.DataFrame(common_words, columns = ['trigram' ,
'count'])

fig = go.Figure([go.Bar(x=df_toptri['trigram'], y=df_toptri
['count'])])
fig.update_layout(title=go.layout.Title(text="Top 20 trigrams in the
airbnb documents"))
fig.show()
```

Once the data has been preprocessed, it is then time to generate the first baseline topic model. We start by using the class `gensim.corpora.Dictionary` to set a unique ID for each word in our corpus. Thus, we have a dictionary defining all the words that we are currently working with. In the next step, a document can be represented either as a vector or as a bag-of-words (BOW); in this case, the latter will be used.

```
#Create Dictionary
df["lemmatized_split"] = df["lemmatized"].map(lambda x: x.split())
id2word = corpora.Dictionary(df["lemmatized_split"])

#Keep top n words ordered by term frequency across the corpus
n=5000
```

```
id2word.filter_extremes(no_below=1, no_above=1, keep_n=n)
#Get bag of words representation (word_id, frequency)
corpus = [id2word.doc2bow(doc) for doc in df['lemmatized_split']].
tolist()
```

Now, we can fit a baseline model with three topics to get an overall idea of the data. This model will be fine-tuned in one of the next steps.

```
lda = LdaModel(corpus, num_topics = 3, id2word=id2word, passes=50)
```

After visualising the three topics with the interactive LDAvis tool (see Jupyter notebook), we have a rough idea of the most relevant words per topic. In the next step, hyperparameters are tuned so as to optimise the model and measure the coherence score between candidate models. We use standard values for the chunksize, which defines how many documents are processed in the training at a time, the passes value (or epochs), which controls how often the model is trained on the entire corpus (Kapadia, 2019), the topic range, and the step size. All three hyperparameters (number of Topics (K), Document-Topic Density ( $\alpha/a$ ), and Word-Topic Density ( $\eta/\beta/b$ )) are tested step by step, where one parameter always varies and the other parameters are kept constant. This procedure is applied to two different corpus validation sets. After that, we can use the coherence score  $C_v$  as a performance measure to find the best constellation.

```
def compute_coherence_values(mycor, mydic, k, a, b):
    lda_model = gensim.models.LdaMulticore(corpus=mycor,
#id2word=dictionary,
id2word=mydic,
num_topics=k,
random_state=100,
chunksize=100,
passes=10,
alpha=a,
eta=b)

    coherence_model_lda = CoherenceModel(model=lda_model, texts=df
['lemmatized_split'].tolist(),
dictionary=mydic, coherence='c_v')

    return coherence_model_lda.get_coherence()

def main_hyperparameters_search():
    grid = {}
    grid['Validation_Set'] = {}

    #Topics range
    min_topics = 2
    max_topics = 15
    step_size = 1
    topics_range = range(min_topics, max_topics, step_size)
```

```

# Alpha parameter
alpha = list(np.arange(0.01, 1, 0.3))
alpha.append('symmetric')
alpha.append('asymmetric')

# Beta parameter
beta = list(np.arange(0.01, 1, 0.3))
beta.append('symmetric')

# Validation sets
num_of_docs = len(corpus)
corpus_sets = [# gensim.utils.ClippedCorpus(corpus,
num_of_docs*0.25),
# gensim.utils.ClippedCorpus(corpus, num_of_docs*0.5),
# gensim.utils.ClippedCorpus(corpus, num_of_docs*0.75),
corpus]

corpus_title = ['100% Corpus']

model_results = {'Validation_Set': [],
'Topics': [],
'Alpha': [],
'Beta': [],
'Coherence': []
}

# Can take a long time to run
if 1 == 1:
pbar = tqdm.tqdm(total=(len(beta)*len(alpha)*len(topics_range)*len(
corpus_title)))

# iterate through validation corpuses
for i in range(len(corpus_sets)):
# iterate through number of topics
for k in topics_range:
# iterate through alpha values
for a in alpha:
# iterare through beta values
for b in beta:
# get the coherence score for the given parameters
cv = compute_coherence_values(mycor=corpus_sets[i], mydic=id2word,
k=k, a=a, b=b)
# Save the model results
model_results['Validation_Set'].append(corpus_title[i])
model_results['Topics'].append(k)
model_results['Alpha'].append(a)
model_results['Beta'].append(b)
model_results['Coherence'].append(cv)

pbar.update(1)
df = pd.DataFrame(model_results)
df.to_csv('lda_tuning_results.csv', index=False)

```

```
pbar.close()
return df

if os.path.exists(os.path.join(os.getcwd(), 'lda_tuning_results.csv')):
    print('Results loaded from {}'.format(os.path.join(os.getcwd(), 'lda_tuning_results.csv')))
    df_hyper = pd.read_csv('lda_tuning_results.csv')
else:
    print('Hyperparameters search, it will take some time...')
    df_hyper = main_hyperparameters_search()
df_hyper.head()
```

This will provide us with the coherence score for all different combinations of  $K$ ,  $\alpha$ , and  $\beta$  (only the first five combinations are listed below) (Table 3).

Next, let's see the highest coherence score archived.

```
best = df_hyper[df_hyper.Coherence == df_hyper.Coherence.max()]
print('The highest coherence score is {}'.format(best.Coherence.values[0]))
```

*Output: The highest coherence score is 0.601.*

Lastly, we also want to see the combination producing the highest score.

```
print('And the corresponding parameters are following:')
best[['Topics', 'Alpha', 'Beta']].reset_index(drop=True)
```

*Output:*

	Topics	Alpha	Beta
0	13	0.91	0.61

To get a better overview of the grid-search results, we can then visualise them in an interactive 3D-scatterplot (Fig. 7).

Finally, let's visualise the results with the interactive pyLDAvis tool.

```
# Visualize LDA after optimization
lda_optim = LdaModel(corpus, num_topics = ntopics, id2word=id2word,
passes=50, alpha=alpha, eta=beta)
```

**Table 3** Coherence scores

	Validation_Set	Topics	Alpha	Beta	Coherence
0	100% corpus	2	0.01	0.01	0.42945
1	100% corpus	2	0.01	0.31	0.425401
2	100% corpus	2	0.01	0.61	0.431354
3	100% corpus	2	0.01	0.91	0.41898
4	100% corpus	2	0.01	Symmetric	0.425401

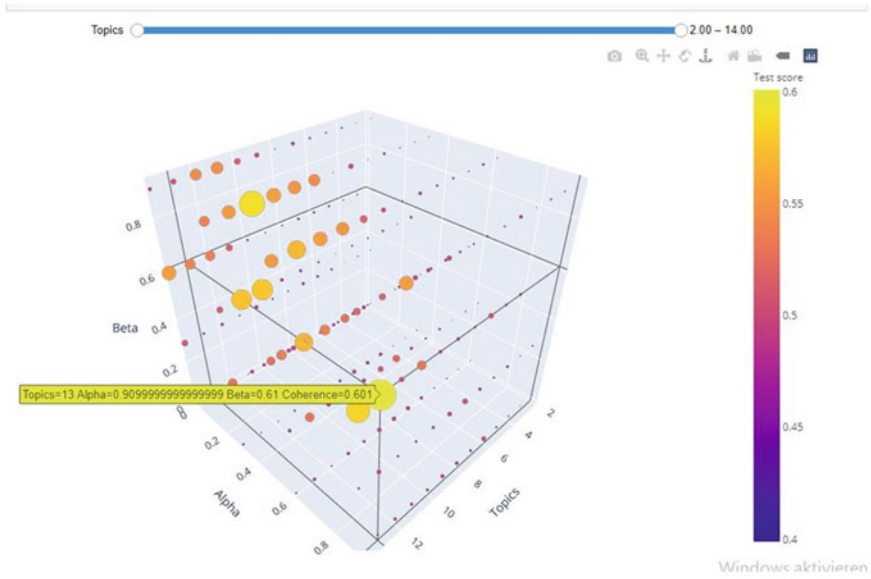
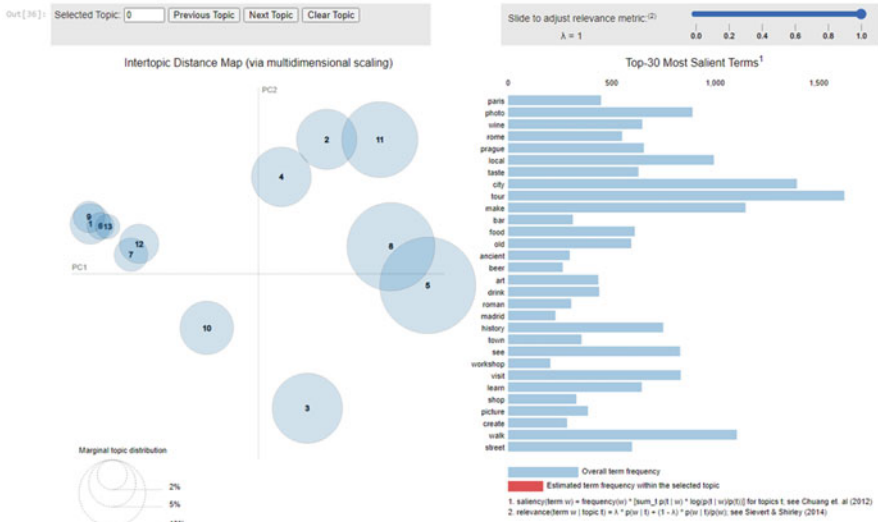


Fig. 7 3D-Scatterplot of results

```
lda_visualization = pyLDAvis.gensim.prepare(lda_optim, corpus, id2word, sort_topics=False) pyLDAvis.display(lda_visualization)
```





As we can see, topics 1, 6, 9, and 13 are overlapping; we should therefore go into more detail by inspecting the most salient terms. If necessary, word intrusion or topic intrusion can be used to further evaluate the results with a qualitative human judgement approach (Chang et al., 2009). Based on this, we can decide whether further changes need to be made or if the results can be accepted as the final solution.

## 6.2 Topic Modelling with CorEx

After having loaded the preprocessed dataset, we will again create a dictionary a dictionary and a apply TF-IDF transformation

```
df["lemmatized_split"] = df["lemmatized"].map(lambda x: x.split())
id2word = corpora.Dictionary(df["lemmatized_split"])

text_string = [' '.join(d) for d in df['lemmatized_split']].tolist()
np.random.seed(42)
n_features=n_features

tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,
max_features=n_features, ngram_range=(1,2), stop_words='english')
tfidf = tfidf_vectorizer.fit_transform(text_string)
vocab = tfidf_vectorizer.get_feature_names()
```

Next, we can define the anchors to nudge the model towards specific terms. A certain minimum domain knowledge is necessary and it must be justifiable why and how the anchoring is applied in the research context.

```
anchors = [
    ["sight"],
    ["activity"],
    ["photo"],
]

anchors = [
    [a for a in topic if a in vocab]
    for topic in anchors
]

anchors_dict = {w[0] : a for a,w in enumerate(anchors)}

model = ct.Corex(n_hidden=len(anchors), seed=42)
model = model.fit(
    tfidf,
    words=vocab,
    anchors=anchors,
    anchor_strength=6 # Tell the model how much it should rely on the anchors
)
```

Finally, lets generate the topics and preview the most relevant ten terms of each topic

```
topic_words = []
for i, topic_ngrams in enumerate(model.get_topics(n_words=10)):
    topic_ngrams = [ngram[0] for ngram in topic_ngrams if ngram[1] > 0]
    print("Topic #{}: {}".format(i+1, " ".join(topic_ngrams)))
    topic_words.append(topic_ngrams)
```

Output:

```
Topic #1: paris, pilot, notre dame, dame, notre, maisonslaffitte, pont
neuf, neuf, louvre, saintgermain
Topic #2: workshop, material, shopping, instawalk, perfume, leather,
fashion, boating, climbing
Topic #3: photo, shoot, photoshoot, picture, capture, photography,
edit, camera, pose, session
```

### Service Section

**Main Application Fields:** There are numerous different topic modelling approaches, all of which have their respective advantages and disadvantages. Basically, all methods try to extract latent topics from texts. They therefore have an inductive and explorative character.

**Limitations and Pitfalls:** For most topic modelling approaches, the number of topics to be extracted must be determined in advance. This is a task that requires both sensitivity and knowledge of appropriate hyperparameter tuning. If default settings are used, it is all too easy to extract nonsense topics that are too broad, too fine granular or simply have no meaningfulness.

**Similar Methods and Methods to Combine with:** Topic Modelling is particularly suitable in combination with semantic analysis. Text clustering could be seen as a similar procedure.

**Code:** The Python Code is available at: <https://github.com/DataScience-in-Tourism/Chapter-18-Topic-Modeling>

## Further Readings and Other Sources

A great tutorial about LDA - “An approachable explanation of how Topic Modelling works” – can be found at <https://topix.io/tutorial/tutorial.html>. This tutorial also provides an interactive explanation of the Gibbs Sampling process.

Greene and Cross (2017) present a paper together with the code for a dynamic topic modeling approach, allowing to track how topics evolve over time. <https://github.com/derekgreene/dynamic-nmf>

As further literature for R-users the book by Jockers and Thalken (2020) can be suggested

A very good overview on the application of topic models by social scientists is given in the book by Papilloud and Hinneburg (2018) “Qualitative Textanalyse mit Topic Modellen, eine Einführung für Sozialwissenschaftler” (only in German language available)

Furthermore, the following online articles are recommended:

<https://towardsdatascience.com/topic-modelling-of-2019-hr-tech-conference-tweet-d16cf75895b6>

<https://towardsdatascience.com/topic-modeling-with-nlp-on-amazon-reviews-an-application-of-latent-dirichlet-allocation-lda-ae42a4c8b369>

<https://medium.com/@kurtsenol21/topic-modeling-lda-mallet-implementation-in-python-part-1-c493a5297ad2>

<https://towardsdatascience.com/short-text-topic-modeling-70e50a57c883>

<https://towardsdatascience.com/introduction-to-nlp-part-5b-unsupervised-topic-model-in-python-ab04c186f295>

<https://sagarpanwar249.medium.com/guide-to-topic-modeling-eac693c9d3e0>

## References

- Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020). Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in Artificial Intelligence*, 3, 42. <https://doi.org/10.3389/frai.2020.00042>
- Alexander, E., Kohlmann, J., Valenza, R., Witmore, M., & Gleicher, M. (2014). Serendip: Topic model-driven visual exploration of text corpora. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)* (pp. 173–182). IEEE. <https://doi.org/10.1109/VAST.2014.7042493>
- Angelov, D. (2020a). *Top2Vec: Distributed Representations of Topics*. Retrieved from <http://arxiv.org/pdf/2008.09470v1>
- Angelov, D. (2020b, April 16). *COVID-19: Topic modeling and search with Top2Vec: Kaggle – Jupyter Notebook*. Retrieved from <https://www.kaggle.com/dangelov/covid-19-topic-modeling-and-search-with-top2vec>
- Banks, G. C., Woznyj, H. M., Wesslen, R. S., & Ross, R. L. (2018). A review of best practice recommendations for text analysis in R (and a user-friendly app). *Journal of Business and Psychology*, 33(4), 445–459. <https://doi.org/10.1007/s10869-017-9528-3>
- Bi, J.-W., Liu, Y., Fan, Z.-P., & Cambria, E. (2019). Modelling customer satisfaction from online reviews using ensemble neural network and effect-based Kano model. *International Journal of Production Research*, 57(22), 7068–7088. <https://doi.org/10.1080/00207543.2019.1574989>
- Blair, S. J., Bi, Y., & Mulvenna, M. D. (2020). Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, 50(1), 138–156. <https://doi.org/10.1007/s10489-019-01438-z>
- Blei, D. M. (2012a). Probabilistic topic models. *Communications of the ACM*, 55(4), 77–84.
- Blei, D. M. (2012b). Probabilistic topic models. *Communications of the ACM*, 55(4), 77–84. <https://doi.org/10.1145/2133806.2133826>
- Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan), 993–1022.

- Cai, G., Sun, F., & Sha, Y. (2018). Interactive visualization for topic model curation. *IUI Workshops*.
- Calheiros, A. C., Moro, S., & Rita, P. (2017). Sentiment classification of consumer-generated online reviews using topic modeling. *Journal of Hospitality Marketing & Management*, 26(7), 675–693. <https://doi.org/10.1080/19368623.2017.1310075>
- Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *Advances in Neural Information Processing Systems*. Retrieved from <http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf>
- Daenekindt, S., & Huisman, J. (2020). Mapping the scattered field of research on higher education. A correlated topic model of 17,000 articles, 1991–2018. *Higher Education*, 80(3), 571–587. <https://doi.org/10.1007/s10734-020-00500-x>
- Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Analysis*, 26(2), 168–189. <https://doi.org/10.1017/pan.2017.44>
- Dickinger, A., Lalicic, L., & Mazanec, J. (2017). Exploring the generalizability of discriminant word items and latent topics in online tourist reviews. *International Journal of Contemporary Hospitality Management*, 29(2), 803–816. <https://doi.org/10.1108/IJCHM-10-2015-0597>
- Dietz, L. (2016). Topic model evaluation: How much does it help? WebSci2016, University Mannheim.
- DiMaggio, P., Nag, M., & Blei, D. (2013). Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding. *Poetics*, 41(6), 570–606. <https://doi.org/10.1016/j.poetic.2013.08.004>
- Egger, R., & Yu, J. (2021). Identifying hidden semantic structures in Instagram data: A topic modelling comparison. *Tourism Review*.
- Egger, R. (2022). Machine learning in tourism – a brief overview. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications (n.a)*. Springer.
- Egger, R., Pagiri, A., Prodinger, B., Liu, R., & Wettinger, F. (2022, January). Topic modelling of tourist dining experiences based on the GLOBE Model. In *ENTER22 e-tourism conference* (pp. 356–368). Springer.
- Evans, M. S. (2014). A computational approach to qualitative analysis in large textual datasets. *PLoS One*, 9(2), e87908. <https://doi.org/10.1371/journal.pone.0087908>
- Gallagher, R. J., Reing, K., Kale, D., & Ver Steeg, G. (2017). Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5, 529–542. [https://doi.org/10.1162/tac1\\_a\\_00078](https://doi.org/10.1162/tac1_a_00078)
- Greene, D., & Cross, J. P. (2017). Exploring the political agenda of the European Parliament using a dynamic topic modeling approach. *Political Analysis*, 25(1), 77–94. <https://doi.org/10.1017/pan.2016.7>
- Greene, D., O’Callaghan, D., & Cunningham, P. (2014, April 16). *How many topics? Stability analysis for topic models*. Retrieved from <http://arxiv.org/pdf/1404.4606v3>
- Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3), 267–297. <https://doi.org/10.1093/pan/mps028>
- Grootendorst, M. (2020, May 10). *Topic modeling with BERT. | Towards data science*. Retrieved from <https://towardsdatascience.com/topic-modeling-with-bert-779f7db187e6>
- Grootendorst, M. (2021, June 1). *Interactive topic modeling with BERTopic | Towards data science*. Retrieved from <https://towardsdatascience.com/interactive-topic-modeling-with-bertopic-1ea55e7d73d8>
- Guo, Y., Barnes, S. J., & Jia, Q. (2017). Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management*, 59, 467–483. <https://doi.org/10.1016/j.tourman.2016.09.009>

- Hannigan, T. R., Haans, R. F. J., Vakili, K., Tchalian, H., Glaser, V. L., Wang, M. S., . . . Jennings, P. D. (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, 13(2), 586–632. <https://doi.org/10.5465/annals.2017.0099>
- Hindle, A., Ernst, N. A., Godfrey, M. W., & Mylopoulos, J. (2013). Automated topic naming. *Empirical Software Engineering*, 18(6), 1125–1155. <https://doi.org/10.1007/s10664-012-9209-9>
- Hornik, K., & Grün, B. (2011). Topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13), 1–30. Retrieved from <https://epub.wu.ac.at/3987/>
- Hu, J., Sun, X., & Li, B. (2015). Explore the evolution of development topics via on-line LDA. In *IEEE 22nd international conference* (pp. 555–559). IEEE. <https://doi.org/10.1109/SANER.2015.7081876>
- Hu, N., Zhang, T., Gao, B., & Bose, I. (2019). What do hotel customers complain about? Text analysis using structural topic model. *Tourism Management*, 72, 417–426. <https://doi.org/10.1016/j.tourman.2019.01.002>
- Jockers, M. L., & Thalken, R. (2020). Topic modeling. In M. L. Jockers & R. Thalken (Eds.), *Quantitative methods in the humanities and social sciences. Text analysis with R* (pp. 211–235). Springer International Publishing. [https://doi.org/10.1007/978-3-030-39643-5\\_17](https://doi.org/10.1007/978-3-030-39643-5_17)
- Kadhim, A. I., Cheah, Y.-N., & Ahamed, N. H. (2014). Text document preprocessing and dimension reduction techniques for text document clustering. In *2014 4th international conference on artificial intelligence with applications in engineering and technology* (pp. 69–73). IEEE. <https://doi.org/10.1109/ICAJET.2014.21>
- Kapadia, S. (2019, August 19). *Evaluate topic models: Latent Dirichlet Allocation (LDA). Towards data science*. Retrieved from <https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0>
- Kim, K., Park, O., Barr, J., & Yun, H. (2019). Tourists' shifting perceptions of UNESCO heritage sites: Lessons from Jeju Island-South Korea. *Tourism Review*, 74(1), 20–29. <https://doi.org/10.1108/TR-09-2017-0140>
- Konrad, M. (2017). Tmtoolkit [computer software].
- Kuang, D., Brantingham, P. J., & Bertozzi, A. L. (2017). Crime topic modeling. *Crime Science*, 6(1), 12. <https://doi.org/10.1186/s40163-017-0074-0>
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3), 259–284. <https://doi.org/10.1080/01638539809545028>
- Lee, D., & Seung, H. S. (1999). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13, 556–562.
- Lesnikowski, A., Belfer, E., Rodman, E., Smith, J., Biesbroek, R., Wilkerson, J. D., . . . Berrang-Ford, L. (2019). Frontiers in data analytics for adaptation research: Topic modeling. *Wiley Interdisciplinary Reviews: Climate Change*, 10(3), e576. <https://doi.org/10.1002/wcc.576>
- Liangjie, H., & Davison, B. D. (2010). Empirical study of topic modeling in twitter. In P. Melville (Ed.), *Proceedings of the first workshop on social media analytics* (pp. 80–88). ACM.
- Lim, K. W., & Buntine, W. (2014). Twitter opinion topic model. In J. Li, X. S. Wang, M. Garofalakis, I. Soboroff, T. Suel, & M. Wang (Eds.), *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (pp. 1319–1328). ACM. <https://doi.org/10.1145/2661829.2662005>
- Lindstedt, N. C. (2019). Structural topic modeling for social scientists: A brief case study with social movement studies literature, 2005–2017. *Social Currents*, 6(4), 307–318. <https://doi.org/10.1177/2329496519846505>
- Loureiro, S. M. C., Guerreiro, J., & Ali, F. (2020). 20 years of research on virtual reality and augmented reality in tourism context: A text-mining approach. *Tourism Management*, 77, 104028. <https://doi.org/10.1016/j.tourman.2019.104028>
- Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., . . . Adam, S. (2018). Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. *Communication Methods and Measures*, 12(2–3), 93–118. <https://doi.org/10.1080/19312458.2018.1430754>

- McCallum, A. (2002). MALLET: A machine learning for language Toolkit from <https://ci.nii.ac.jp/naid/20001704926/>
- Mimno, D. (2013). jsLDA [Computer software].
- Mimno, D., Hanna, W., Edmond, T., Miriam, L., & Andrew, M. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 262–272. Retrieved from <https://www.aclweb.org/anthology/D11-1024.pdf>
- Mimno, D., Wallach, H., Naradowsky, J., Smith, D. A., & McCallum, A. (2009). Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.
- Mohammed, S. H., & Al-augby, S. (2020). LSA & LDA topic modeling classification: comparison study on e-books. *Indonesian Journal of Electrical Engineering and Computer Science*, 19(1), 353. <https://doi.org/10.11591/ijeecs.v19.i1.pp353-362>
- Murugan, A., Chelsey, H., & Thomas, N. (2019). *Practical text analytics*. Springer International Publishing.
- Nikolenko, S. I., Koltcov, S., & Koltsova, O. (2017). Topic modelling for qualitative studies. *Journal of Information Science*, 43(1), 88–102. <https://doi.org/10.1177/0165551515617393>
- Papiloud, C., & Hinneburg, A. (2018). *Qualitative Textanalyse mit topic-Modellen*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-21980-2>
- Park, E., Chae, B., & Kwon, J. (2018). The structural topic model for online review analysis. *Journal of Hospitality and Tourism Technology*, 11(1), 1–17. <https://doi.org/10.1108/JHTT-08-2017-0075>
- Qin, L., Shaobo, L., Sen, Z., Jie, H., & Jianjun, H. (2019). A Review of text corpus-based tourism big data mining. In *Applied Sciences*, 9(16), 3300. <https://doi.org/10.3390/app9163300>
- Qiu, Z., & Shen, H. (2017). User clustering in a dynamic social network topic model for short text streams. *Information Sciences*, 414, 102–116. <https://doi.org/10.1016/j.ins.2017.05.018>
- Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (Ed.) (2009). *A supervised topic model for credit attribution in multi-labeled corpora*.
- Ramage, D., Rosen, E., Chuang, J., Manning, C. D., & McFarland, D. A. (2009). *Topic modeling for the social sciences: Topic modeling for the social sciences*. NIPS. NIPS 2009 workshop on applications for topic models: Text and beyond. Retrieved from <http://nlp.stanford.edu/dramage/papers/tmt-nips09.pdf>
- Rehurek, R., & Sojka, P. (2010). *Software framework for topic modelling with large corpora: Rehurek, Radim, and Petr Sojka. "Software"*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.695.4595>
- Reing, K., Kale, D. C., Steeg, G. V., & Galstyan, A. (2016). *Toward interpretable topic discovery via anchored correlation explanation*. arXiv preprint arXiv:1606.07043.
- Reisenbichler, M., & Reutterer, T. (2019). Topic modeling in marketing: Recent advances and research opportunities. *Journal of Business Economics*, 89(3), 327–356. <https://doi.org/10.1007/s11573-018-0915-7>
- Roberts, M. E., Stewart, B. M., & Tingley, D. (2019). Stm: An R package for structural topic models. *Journal of Statistical Software*, 91(2), 1–40. <https://doi.org/10.18637/jss.v091.i02>
- Röder, M., Both, A., & Hinneburg, A. (Eds.) (2015). *Exploring the space of topic coherence measures*.
- Rossetti, M., Stella, F., Cao, L., & Zanker, M. (2015). Analysing User Reviews in Tourism with Topic Models. In I. Tussyadiah & A. Inversini (Eds.), *Information and Communication Technologies in Tourism 2015* (pp. 47–58). Cham: Springer International Publishing. [https://doi.org/10.1007/978-3-319-14343-9\\_4](https://doi.org/10.1007/978-3-319-14343-9_4)
- Rossetti, M., Stella, F., & Zanker, M. (2016). Analyzing user reviews in tourism with topic models. *Information Technology & Tourism*, 16(1), 5–21. <https://doi.org/10.1007/s40558-015-0035-y>
- Saussure, F. D. (1959). *Course in general linguistics* (W. Baskin, Trans.). Philosophical Library.

- Shafqat, W., & Byun, Y.-C. (2020). A recommendation mechanism for under-emphasized tourist spots using topic modeling and sentiment analysis. *Sustainability*, 12(1), 320. <https://doi.org/10.3390/su12010320>
- Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. In J. Chuang, S. Green, M. Hearst, J. Heer, & P. Koehn (Eds.), *Proceedings of the workshop on interactive language learning, visualization, and interfaces* (pp. 63–70). Association for Computational Linguistics. <https://doi.org/10.3115/v1/W14-3110>
- Simmler, S., Vitt, T., & Pielström, S. (2019). Topic modeling with interactive visualizations in a GUI tool. In *Proceedings of the Digital Humanities Conference*.
- Sotomayor O. D., & Bellono G. (2019, January 3). *Automated topic discovery: An approachable explanation*. Retrieved from <https://topix.io/tutorial/tutorial.html>
- Taecharungroj, V., & Mathayomchan, B. (2019). Analysing TripAdvisor reviews of tourist attractions in Phuket, Thailand. *Tourism Management*, 75, 550–568. <https://doi.org/10.1016/j.tourman.2019.06.020>
- Ungar, L., Craven, M., Gunopulos, D., & Eliassi-Rad, T. (2006). Topics over time: A non-Markov continuous-time model of topical trends: Proceedings of the twelfth ACM SIGKDD International Conference on Knowledge Discovery and data mining August 20–23, 2006, Philadelphia, PA, USA, 424–433.
- Ver Steeg, G. (2016). *Open source project implementing hierarchical topic models on sparse data*. Retrieved from [https://github.com/gregversteeg/corex\\_topic](https://github.com/gregversteeg/corex_topic)
- Vo, D.-T., & Ock, C.-Y. (2015). Learning to classify short text from scientific documents using topic models with various types of knowledge. *Expert Systems with Applications*, 42(3), 1684–1698. <https://doi.org/10.1016/j.eswa.2014.09.031>
- Vu, H. Q., Li, G., & Law, R. (2019). Discovering implicit activity preferences in travel itineraries by topic modeling. *Tourism Management*, 75, 435–446. <https://doi.org/10.1016/j.tourman.2019.06.011>
- Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. In A. Danyluk, L. Bottou, & M. Littman (Eds.), *Proceedings of the 26th Annual International Conference on Machine Learning – ICML '09* (pp. 1–8). ACM Press. <https://doi.org/10.1145/1553374.1553515>
- Wang, J., Li, Y., Wu, B., & Wang, Y. (2020). Tourism destination image based on tourism user generated content on internet. *Tourism Review*. <https://doi.org/10.1108/TR-04-2019-0132>. (ahead-of-print).
- Wang, Y.-X., & Zhang, Y.-J. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6), 1336–1353. <https://doi.org/10.1109/TKDE.2012.51>
- Wei, X., Xin, L., & Yinhong, G. (2003). Document clustering based on non-negative matrix factorization. In J. Callan (Ed.), *Special issue of the SIGIR forum, Sigir 2003: Proceedings of the twenty-sixth annual international ACM SIGIR conference on Research and Development in information retrieval, Toronto, Canada, July 28 to august 1, 2003* (pp. 267–273). ACM Press.
- Weng, J. (2020, December 21). Topic modeling in one line with Top2Vec – towards data science. *Towards data science*. Retrieved from <https://towardsdatascience.com/topic-modeling-in-one-line-with-top2vec-a413991aa0ef>
- Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., & Li, X. (2011). Comparing twitter and traditional media using topic models. In P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, & V. Mudoch (Eds.), *Lecture notes in computer science, advances in information retrieval* (pp. 338–349). Springer. [https://doi.org/10.1007/978-3-642-20161-5\\_34](https://doi.org/10.1007/978-3-642-20161-5_34)

# Entity Matching: Matching Entities Between Multiple Data Sources



Ivan Bilan

## Learning Objectives

- Illustrate the steps required to build an entity matching pipeline
- Explain how entity matching can be applied in the tourism industry
- Demonstrate how to engineer an end-to-end entity matching pipeline using Python

## 1 Introduction and Theoretical Foundations

### 1.1 Entity Matching Problem Statement

Entity matching describes the approach of finding records that refer to the same real-world entity across different databases or any other data storage types. These entities are identified by cross-checking their identifiers, such as name, address, phone number, and the like. Entity matching, also known as record linkage, has applications in various scientific fields and industrial solutions, ranging from matching people in census data (Christen, 2012), bibliographic databases (Christen, 2012), forensic data and DNA matching (Tai, 2018), physical objects like businesses, and many more. It is mainly used to consolidate records of the same type and especially used with textual data. For example, when matching company entities from two databases, they can be matched by name and address. Additionally, entity matching is often used in the process of finding duplicates within the same database.

The challenge of entity matching arises mainly from there being no cross-industry standard on how to store such entities and their identifiable markers in a consistent

---

I. Bilan (✉)  
TrustYou GmbH, Munich, Germany



way. The simple example of matching the same company entity from two databases becomes challenging if these entities are stored in different formats. For example, in one case, the company may be stored in the database using just two columns, one for the full name and legal entity type and another one for the full address, whereas, in the second database, the company entity might have separate columns for the name, legal entity, city, street, building number, and so on. Working on an entity matching solution requires a significant amount of time to invest in generating quality training data that, for a subset of records, includes the match status of whether the items in a record pair belong to a single real-world entity or not (Christen, 2012).

## 1.2 Entity Matching Examples in the Travel Industry

Entity matching is also widely used in the tourism industry, and there are various scenarios in which a company needs to rely on entity matching approaches. One of the most used applications involves the deduplication of hotel reviews that are crawled from various sources. Another widespread example is matching hotels or accommodations between various sources or deduplicating them within one source (Bayrak et al., 2019; Kozhevnikov & Gorovoy, 2016). One scenario relating to this might be to extend a database when onboarding a new hotel chain. Even if the company already has hotels in their database, they might not have all of the hotels the clients want to use for their solution. As such, the company would need to align its database to the database of their clients in order to find the correct data of a specific hotel and deliver their analytic insights to the correct hotel.

Another typical scenario is the need of consolidating all reviews for specific hotels from various websites that post traveler reviews. Often, differences can be observed in how the hotels are displayed on such websites. The names of the hotel can differ, especially if the websites are based in different countries, and addresses can also vary or display varying levels of detail. Some sources may display the phone number and the official website of the hotel, or, if the hotel belongs to a specific chain of hotels, these may sometimes simply point to the chain's headquarters instead.

Table 1 shows a general example of how two hotel records involving the same entity can be stored in various data sources. This particular example illustrates why entity matching goes beyond merely matching two databases via a direct comparison of database columns. One of the specific issues with this example is that the

**Table 1** Example of different records from the same hotel entity

	Hotel name	Address	City	Street	Zip	Country
Data Source 1	Hotel Kaltbräu City	Tal 11, Lehel, München	–	–	–	Deutschland
Data Source 2	Hotel Kaltbraeu City	–	Munich	Tal 11	80331	Germany

**Table 2** Example of different records from the same hotel entity in different languages

	Hotel name	Address	Phone number	Website
Data Source 1	サン シャイ ン	東京都豊島区東池袋2-3-4	0356411167	<a href="http://www.japanhotels.co.jp/hotelsunshine">www.japanhotels.co.jp/hotelsunshine</a>
Data Source 2	Sunshine	2-3-4 Higashi-Ikebukuro, Toshima-ku, Tokyo	+081 035-641- 1167	<a href="http://japanhotels.co.jp">japanhotels.co.jp</a>

addresses are saved in very different formats. In the first example, the address is stored in a single address column, while in the second example, each address identifier, such as city, street, or zip code, is given separately. There is also a small difference in how the name is normalized in each data source, with the second data source normalizing German umlauts by using the English alphabet. Such a record cannot be matched directly by its name, and the address information is scattered between various database columns.

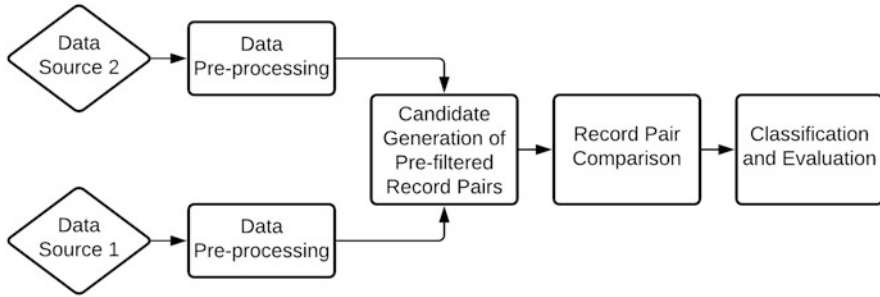
Table 2 shows an extreme case of the above in which two records from the same hotel entity are stored between two data sources with very different identifiers. The first data source shows the hotel's name and address in Japanese, while the other data source is written using the transliterated English form. Moreover, the phone numbers do not fully match since, in one of the sources, the number has the country-specific identifier attached at the beginning. Additionally, the websites also do not fully match.

More difficulties arise when dealing with a large number of records; for example, there might be numerous hotels at the same address or the names of the hotels might be very similar. Moreover, some data records may have incomplete data points when compared to others.

### 1.3 Overview of the Stages of an Entity Matching Approach

Entity matching approaches usually follow similar steps. The process usually starts with the (1) data pre-processing step followed by (2) pre-filtering potential candidate pairs, (3) record pair comparison, and, finally, (4) classification of each record pair as a true or negative match. This process is illustrated in Fig. 1.

Pre-processing data is necessary because the structure of the data and its general representation usually varies among data sources. Many steps can be taken to pre-process data. In the particular hotel examples provided above, the address data points can be stored uniformly between data sources by either splitting them into the smallest available identifiers or by joining them together into one address identifier. The data can also be normalized to a single language or a language-agnostic representation. For example, instead of using the name of the hotel in various languages, it can be transliterated into the English alphabet. For most languages,



**Fig. 1** General steps of an entity-matching pipeline

textual identifiers can be normalized to English with the help of various Unicode normalization techniques.<sup>1</sup> Other specialized tools dedicated to such transliterations, for instance, *jaconv*<sup>2</sup> for Japanese, also exist. Additionally, the transliterated form can be transformed into a phonetic representation that would only reflect how the text is pronounced and not necessarily how it is written. This is possible with such phonetic algorithms as Soundex<sup>3</sup> (a detailed overview of it and many other similar algorithms is given by Christen (2012)). Moreover, as the type of hotel or another part of the address often end up in the hotel name field of various record representations, names can be cleaned up by removing general words in the name field that do not necessarily belong to the actual name.

Before comparing records, the next step is to pre-filter potential matching record pairs. This step can technically be omitted if the amount of records in need of comparison is very small. However, when comparing large databases, computational limitations may be reached fairly quickly. If the record pairs were not pre-filtered, the entity matching pipeline will have to compare every single record from the first source to each record in the second source and then repeat the process for each individual record in source one. This approach has quadratic computational complexity (Christen, 2012); hence, the pre-filtering step is recommended for production-ready solutions. The pre-filtering of potential matching record pairs is also known in the research literature as indexing, blocking (Kirsten et al., 2010), or candidate generation (Kong et al., 2016). In the end, the ultimate goal of this step is to find a way to limit the number of comparisons needed for each record. Continuing with the hotel example from above, a fast way to do blocking is to only compare records from the first source to records in the second source that are within the same city or on the same street. This will largely alleviate the computational complexity of the task.

After the blocking step, each record pair needs to be compared. Even after extensive pre-processing, many differences between record identifiers may still

<sup>1</sup>Unicode Normalization Forms <https://unicode.org/reports/tr15/>

<sup>2</sup>Japanese character interconverter <https://github.com/ikegami-yukino/jaconv>

<sup>3</sup>The Soundex Indexing System <https://www.archives.gov/research/census/soundex>

remain; in other words, relying on one of the identifiers between the sources to be an exact match or not is often not enough. As a result, a measure of how similar the identifiers are to each other is required. There are various approaches to conduct such a comparison, and it usually depends on the type of identifier. When comparing names or addresses, various string similarity algorithms can be used on the text itself, or such a comparison can be done using word embedding vector representations of the compared identifiers. Both approaches can be combined as well.

There are many approaches available to compute string similarity between two textual data points, such as Levenshtein Edit Distance (Hyyrö, 2003), Jaro Winkler (Keil, 2019), Hamming, Monge-Elkan string comparison algorithms (Cohen et al., 2003), and many more. Christen (2012) gives a detailed overview of various string comparison algorithms and how they can be applied to record pair comparison. The ultimate use-case of these algorithms for the entity matching task is to provide a numeric similarity score between two texts. For example, when comparing different written variants of hotel names in two data sources, such as “Batu Faly Shamrock Beach 22” and “Batu Faly Shamrock Villa,” such hotel names would not be matched to the same entity when using a one-to-one comparison. Instead, for the latter example, one of the string comparison algorithms could be applied, for instance, Hamming Similarity, which would output a similarity score of 0.70 (on a scale of 0 to 1).

After the pipeline compares each record to its potential match candidate and also each of the record’s identifiers (name, address, phone, etc.) to its mirror identifiers from the compared record in the second source, a classification decision needs to be made on whether a record pair is indeed a match or not. In this regard, mainly two approaches to record pair classification can be exemplified: a threshold-based approach and an approach using an end-to-end neural network classifier.

The threshold-based approach is a simple way of applying entity matching and, if necessary, can be used without a training set. It entails computing a similarity score for each record identifier, summing it up, and deciding whether a record pair is a match if the sum exceeds a set threshold value (Christen, 2012). For example, given two record identifiers in both databases, name and address, each of them can be compared, providing a maximum similarity score of 1 for each identifier, which adds up to 2 if both identifiers are exactly the same. If the threshold of the sum of all similarity scores is set to 1.5, only the record pairs that have a joint similarity score at least that high will be matched. Another example: if the address is the same and yields a similarity score of 1, but the name only yields a similarity score of 0.5, the record pairs will be matched as referring to the same entity because of the threshold. This grows in complexity with more identifier fields, and setting a correct threshold is a matter of experimenting with the quality evaluation of the output of such classifiers. This method can also be applied without having a pre-annotated gold standard. Furthermore, it can be used to help build a new training set much faster than having to annotate data from scratch, without any additional indications of the probability of a record pair being a match.

There are various alternatives to threshold-based classification, one of which is the neural network-based approach. Such approaches can work end-to-end without

defining any specific similarity scores or thresholds. However, they often require a considerable amount of annotated training data. A neural network-based entity matching system typically uses the contextual representation of record identifiers with learned vector embeddings, such as word2vec (Mikolov et al., 2013), fastText (Joulin et al., 2016), or BERT (Devlin et al., 2019) (see Chapter “Text Representations and Word Embeddings” for more details). In addition to the word embeddings, a neural classifier based on recurrent neural networks, a Transformer encoder (Vaswani et al., 2017), or similar neural approaches are used for classification. Zhao and He (2019) provide a more detailed overview of the variants and complex structures of such entity matching systems. In the next section, DeepMatcher, one of the most popular, open-source end-to-end neural entity matching frameworks will be presented (Mudgal et al., 2018).

## 2 Practical Demonstration

### 2.1 Data Formatting and Pre-processing

Let us take a look at a hands-on example of matching hotel entities between data sources. For this demo, we will use Python and various data analysis libraries like “pandas”.<sup>4</sup> The sample dataset contains two different sources with 84 hotel records each. Around two-thirds of the dataset consists of record pairs each referring to a single hotel entity. One-third of the record pairs have very similar record identifiers, like name and address, however, they are not referring to the same hotel entity. All of the hotels provided in this chapter and the dataset are automatically generated and do not refer to real-world hotels.

Let us load the data from each source into a data frame:

```
import pandas as pd
df_source1 = pd.read_csv("./data/source_one.csv")
df_source2 = pd.read_csv("./data/source_two.csv")
```

Table 3 shows a sample of records from our first source, and Table 4 shows a data sample from our second source.

In this particular example, the first source has columns referring to name, address, and phone number, while the second source has name, city, country, zip, street, and phone number. To compare each record identifier separately, we need to make sure that the two datasets have the same columns. In this case, we have two options:

1. In the second source, combine each single address identifier into one address column for each record.

---

<sup>4</sup><https://pandas.pydata.org>

**Table 3** Data sample of hotel records from source 1

Hotel name	Address	Phone number
Valley Country Inn	433 East Route 77, Johnscity, AZ 87030, United States of America	922-456-1178
Comfort Inn Westcity	1821 Harrison Drive, Westcity, WY 81340, United States of America	

**Table 4** Data sample of hotel records from source 2

Hotel name	Street	City	Zip code	Country	Hotel phone number
Valley Country Inn Bed & Breakfast	East Route 77433	Johnscity	87030	United States	+1 922-456-1178
Comfort Inn Westcity	Harrison Drive 1821	Westcity	81340	United States	+1 342-562-8865

2. In the first source, programmatically extract city, country, zip, and street name from the single address column.

The second option will allow us to be more precise in our comparisons and help us create better record candidate pairs. Fortunately, there are tools in the Python ecosystem that will allow us to do such data pre-processing. Using a Python package called “postal,”<sup>5</sup> we start by extracting specific address information from the first source. This particular Python library allows us to automatically parse a single string of address data into separate address units like city, street, country, zip code, etc. It has a simple interface, and with just one function, we can get the expected results. Let us look at a quick demonstration below.

```
from postal.parser import parse_address
print(parse_address("433 East Route 77, Johnscity, AZ 87030, United States of America"))
```

```
Output: [(433, 'house_number'), (east route 77, 'road'), (Johnscity, 'city'), ('az', 'state'), (87030, 'postcode'), ('united states of america', 'country')]
```

With this function, we can take the data from the first source and transform it to have the same columns as the data in the second source. Table 5 shows one record from the first source after having applied the address transformation step.

We still have multiple inconsistencies between data sources in terms of how each column is represented. When we investigate each column in more detail, we can see that the country names are inconsistent. For example, in one source, we can see “United States,” while in the other “United States of America.” Furthermore, the

<sup>5</sup><https://github.com/openvenues/pypostal>

**Table 5** Data sample of hotel records from source 1 after address transformation

Hotel name	Street	City	Zip code	Country	Hotel phone number
Valley Country Inn	433 east route 77 az	johnscity	87030	united states of america	922-456-1178

phone numbers in the first source have no country code, while the records in the second source have the country code attached to the phone number. The first issue can be easily resolved by using the “pycountry” Python library,<sup>6</sup> which allows one to look up a country name in almost any format and to normalize it to a predefined one. For example, we can look up the country names we have in our data and transform all of them into a uniform two-character country code using the following code snippet:

```
import pycountry
pycountry.countries.search_fuzzy("United States")[0].alpha_2
pycountry.countries.search_fuzzy("United States of America")[0].alpha_2
```

Output: “US” for both

The issue with the phone number formatting can also be clarified by using another specialized Python library. With “python-phonenumbers,”<sup>7</sup> we can get a direct local national number:

```
import phonenumbers
phonenumbers.parse("+1 922-456-1178", None).national_number
```

Output: “9224561178”

Various other pre-processing steps can be applied to the data to make it more uniform throughout, and the approaches that need to be taken usually depend on the type of record identifier. After the pre-processing step is finalized, we need to pre-filter the potential candidate record pairs.

## 2.2 Candidate Generation

As discussed in the theoretical part, we do not want to compare each record from the first source to each record in the second source. To avoid this, we can use various approaches to candidate generation of potential match pairs. In this demo, we will

<sup>6</sup><https://pypi.org/project/pycountry/>

<sup>7</sup><https://github.com/daviddrysdale/python-phonenumbers>

make sure to only compare hotels within the same country. This should decrease the number of comparisons we need to compute considerably. In real-world applications, candidate generation is one of the most important steps for the entity matching algorithm. For the purpose of this demo, we are only generating candidates based on the country. However, there are many ways this step can be refined. Explore the various options and approaches to candidate generation by following the tutorials provided on the documentation page of the “py\_entitymatching” Python library.<sup>8</sup>

We will first generate the candidate pairs based on the country code:

```
import py_entitymatching as em
# Instantiate the overlap blocker object
ob = em.OverlapBlocker()

# apply the candidate generation step based on a predefined column
match_candidate_pairs_df = ob.block_tables(df_source1, df_source2,
'country', 'country', word_level=True, overlap_size=1,
l_output_attrs=['name', 'phone', 'city', 'zip', 'street'],
r_output_attrs=['name', 'phone', 'city', 'zip', 'street'])
```

Based on a small dataset of 84 records in each source, by simply overlapping the two sources using the country code, we will need to do computations on more than 746 record pairs. This number grows quadratically with the dataset’s size if no blocking is performed, making the selection of the right approach during candidate selection very important. We can filter out a few more record pairs by, for example, only allowing record pairs in which at least one word in the hotel name exists in both record pairs. This is controlled by defining a column on which blocking can occur and the size of the word overlap. In our example, this is set to 1:

```
strict_match_candidate_pairs_df = ob.block_candset
(match_candidate_pairs_df, 'name', 'name', word_level=True,
overlap_size=1, show_progress=False)
```

Now we have reduced the number of record pairs we need to compare to only around 262. There are many ways this number can be reduced even more; however, it all depends on which record identifiers are available. For example, if you have the longitude and latitude coordinates of the hotels, you can base your candidate generation on a radial distance around a specific hotel from one of the sources and only match it with other hotels within that given distance.

At this stage, both data sources are combined into one data frame including each potential candidate match pair (record identifiers from the first source and record identifiers from the second source) produced by the candidate generation step available for direct comparison. Table 6 illustrates what the dataset should look

---

<sup>8</sup>[http://anhaidgroup.github.io/py\\_entitymatching/v0.3.x/user\\_manual/guides.html#stepwise-guides](http://anhaidgroup.github.io/py_entitymatching/v0.3.x/user_manual/guides.html#stepwise-guides)



**Table 6** Data sample of hotel records following the candidate generation step

record_pair_id	ltable_name	ltable_phone	ltable_country	rtable_name	rtable_city	rtable_country
1	Valley Country Inn	9224561178	US	Valley Country Inn Bed & Breakfast	9224561178	US
2	Valley Country Inn	9224561178	US	Beach Country Inn	9223453468	US

like at this stage (the columns that start with “ltable\_” come from the first source, while the “rtable\_” identifiers are from the second source). Only a subset of record identifiers is shown in this example.

A data frame with the candidate pair from both sources provided in one row allows for within-row record identifier comparison.

### 2.3 Record Pair Comparison (Threshold-based)

Our next step is to compare each identifier in each potential record pair from the previous step and compute a similarity score for each column comparison between the first and second sources. We can use a binary comparison for simple columns, such as phone number and zip code, to see if they are a full match or not. For more complex columns, such as name, street, and city, we can use one of the available string comparison algorithms implemented in the “textdistance” Python library<sup>9</sup> or the “jellyfish” Python library.<sup>10</sup> In this particular demo, we will use the Hamming distance. However, you should experiment with various string similarity measures and see which one yields the best result for your particular data type. The following is an example of how to calculate the Hamming Distance for two strings:

```
import textdistance
textdistance.hamming.normalized_similarity('Batu Faly Shamrock
Beach 22', 'Batu Faly Shamrock Villa')
```

Output: 0.7037037037037037

The similarity score generates a value between 0 and 1 and indicates how similar the two strings are, with 1 denoting complete similarity. We have previously mentioned that we can compare zip codes and phone numbers in a binary fashion: assign 1 if they match entirely and 0 otherwise. However, matching the phone number or the zip code is less significant than matching a name or full street address. For this reason, we should boost the scores of each of these columns accordingly. For the demo, we will boost the name of the hotel by multiplying the final score by 3 and the address by 2. If the phone matches, we will assign a score of 1 to that record identifier comparison, and if the name matches only partially, we will multiply the score by 3, which in our example from above, will boost the similarity score to

---

<sup>9</sup><https://pypi.org/project/textdistance/>

<sup>10</sup><https://pypi.org/project/jellyfish/>

approximately 2.1. For the purpose of a production-ready system, the boosting weights should be computed using a more elaborate approach, for example, a machine learning approach that defines the best boosting values based on the accuracy of the final prediction on the whole gold standard.

Next, we need to sum up all the similarity scores and set a threshold at which we could regard a record pair as an actual match. If we compare name (maximum boosted similarity score of 3), street (maximum boosted similarity score of 2), city (maximum binary similarity score of 1), phone (maximum binary similarity score of 1), and zip code (maximum binary similarity score of 1), all of them add up to a maximum score of 8. For the demo, we will set the threshold for a record pair match at 5.5. The threshold has to be low enough to account for some records missing data in various columns but also high enough to reach a reasonable level of prediction accuracy. For the production-level solutions, make sure to compute a suitable threshold for your use case by doing additional experimentation on your data. Table 7 depicts a simplified illustration of what the data should look like at this stage of the entity matching process, namely, with columns storing the computed and boosted similarity scores for each record identifier in a separate column and the summed up score of all similarity scores for all identifiers.

After filtering out all the record pairs with a total similarity score lower than our set threshold, we get 84 record pairs predicted as matches. Now we also need to make sure that we only allow for a one-to-one comparison; in other words, each record from the first source can only be matched to one record from the second source. Such a limitation is not always applicable, however, as there might be multiple records of the same entity within the same source. As such, the final application depends on the type of data being matched.

Let us extract one-to-one predicted pairs and evaluate the quality based on an annotated gold standard. Our threshold-based classifier reaches 82% precision and 78% recall on our dataset. Using a simple threshold-based approach, we have already achieved significant precision and recall levels without having to rely on an extensive training corpus of thousands of annotated record pairs. This is one of the main advantages of such simple approaches. Next, we will explore a neural-based entity matching system called DeepMatcher.

**Table 7** Data sample of hotel records with similarity scores

record_pair_id	ltable_name	ltable_phone	rtable_name	rtable_phone	name_similarity_score	phone_similarity_score	score_sum
1	Valley Country Inn	9224561178	Valley Country Inn Bed & Breakfast	9224561178	2.52	1	3.52
2	Valley Country Inn	9224561178	Beach Country Inn	9223453468	2.24	0	2.24

## 2.4 Record Pair Comparison (Neural-based)

A neural-based approach to entity matching has its pros and cons. It still requires the pre-processing and candidate generation steps. However, the computation of similarity scores between each record identifier is now a part of a fully automated process within the neural-based approach. The drawback of this approach is its dependency on annotated data with manually matched record pairs. This annotated data is required in order to train a model that will produce acceptable match classification results.

Compared to the threshold-based approach, the amount of code required is significantly lower as the system takes over most of the tasks related to similarity computation and classification. The focus shifts to selecting the right hyperparameters and classifier types for the data type at hand, which is achieved through extensive experimentation. Considering that pre-processing and candidate generation has already been done, it only takes a few lines of code to train and apply a DeepMatcher model:

```
import deepmatcher as dm

# load train, validation and test sets
train, validation, test = dm.data.process(
    path='./data/deep_matcher',
    train='train.csv',
    validation='validation.csv',
    test='test.csv')

# build an Entity Matching model
model = dm.MatchingModel(
    attr_summarizer=dm.attr_summarizers.Hybrid(
        word_contextualizer=dm.word_contextualizers.
        SelfAttention(heads=2))

# run training with appropriate hyperparameters
model.run_train(train, validation, epochs=5, batch_size=8,
    best_save_path='hybrid_model.pth',
    pos_neg_ratio=2
)

# predict matches on the test set
model.run_eval(test)
```

You can experiment with different types of models, explanations of which are given in the tutorial page of DeepMatcher<sup>11</sup> as well as in a paper by Mudgal et al. (2018). For our demo, we are using the Self-Attention Transformer encoder with two attention heads, and we will train the model for five epochs. Additionally,

<sup>11</sup><https://github.com/anhaidgroup/deepmatcher/tree/master/examples>

DeepMatcher allows for the fine-tuning of other various internal components, for example, how to tokenize the text input, what types of word embeddings to use, and more.

After evaluating the trained model, we reach 60% precision on a small test set of 25 samples. This is a good achievement considering that setting up the whole classification process for DeepMatcher is based on a few lines of code; however, higher precision can only be achieved through a larger training set.

### 3 Summary

The ease of use is what truly differentiates the neural-based entity matching approaches from a more rule-based and manual approach shown above. However, this approach needs a considerable amount of training data to perform well on a larger scale dataset. The threshold-based approach also gives one the opportunity to fine-tune the smallest details of how the similarity between record identifiers is computed, which is not easily available with DeepMatcher. The choice of which approach to use depends on the end goal, and the model's inference time, type of data, and availability of training data need to be taken into consideration. Do not discard any approach for its apparent simplicity or complexity and always rely on experimentation to define which one works best for your particular application.

#### **Service Section**

**Main Application Fields:** Entity matching is often used to consolidate records from various databases or data sources into one by joining records from various sources that refer to the same real-world entity. This approach is often applied when joining databases of people, for example, when matching census data from various sources. It is also used when joining data sources containing company entities of various types, for instance, hotels, amongst others. A subset of entity matching approaches is also used for data deduplication, for example, deduplication of hotel reviews within one database or deduplication of guest data between subsidiaries of a single hotel chain.

**Limitations and Pitfalls:** One of the major limitations of entity matching approaches is that there are almost no open-source datasets to work with. Since entity matching can be applied to various entity types and many of these include private information, finding any openly available datasets is extremely hard. The best approach is to usually annotate data internally, which requires a significant amount of time. Another issue is the computational complexity of many entity matching approaches. Comparing two databases, each containing millions of rows, and doing it record-by-record is usually computationally intangible. Much effort is needed to select the most suitable approach for

(continued)

pre-filtering in order to limit the number of record pairs that need to be compared.

**Similar Methods and Methods to Combine with:** Entity matching relies heavily on advances from other research fields such as string similarity matching, document classification, word embeddings, and many more.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-19-Entity-Matching>

## Further Readings and Other Sources

Book: Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection by Christen (2012).

Video: Deep learning for entity matching: A design space exploration <https://www.youtube.com/watch?v=plaONS-Lr8U>

## References

- Bayrak, A.T., Özbek, E.E., Kestepe, S., & Yildiz, O.T. (2019). Intelligent mapping for hotel records representing the same entity (pp. 560–563). In *2019 4th International conference on computer science and engineering (UBMK)*.
- Christen, P. (2012). *Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Publishing Company. Incorporated.
- Cohen, W. W., Ravikumar, P., & Fienberg, S. E. (2003, August). A comparison of string distance metrics for name-matching tasks. *IIWeb*, 3, 73–78.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Hyyrö, H. (2003). A bit-vector algorithm for computing Levenshtein and Damerau edit distances. *Nordic Journal of Botany*, 10(1), 29–39.
- Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). FastText.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.
- Keil, J. M. (2019). Efficient bounded Jaro-Winkler similarity based search. In T. Grust, F. Naumann, A. Böhm, W. Lehner, T. Härder, E. Rahm, A. Heuer, M. Klettke, & H. Meyer (Eds.), *BTW 2019*. Gesellschaft für Informatik.
- Kirsten, T., Kolb, L., Hartung, M., Groß, A., Köpcke, H., & Rahm, E. (2010). Data partitioning for parallel entity matching. arXiv preprint arXiv:1006.5309.
- Kong, C., Gao, M., Xu, C., Qian, W., & Zhou, A. (2016, April). Entity matching across multiple heterogeneous data sources. In *International conference on database systems for advanced applications* (pp. 133–146). Springer.
- Kozhevnikov, I., & Gorovoy, V. (2016). Comparison of different approaches for hotels deduplication. In A.-C. N. Ngomo & P. Křemen (Eds.), *Knowledge engineering and semantic web*. Springer Nature.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

- Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., . . . Raghavendra, V. (2018). *Deep learning for entity matching: A design space exploration*. *SIGMOD '18* (pp. 19–34). Association for Computing Machinery. <https://doi.org/10.1145/3183713.3196926>
- Tai, X. (2018). Record linkage and matching problems in forensics (pp. 510–517). In *2018 IEEE International conference on data mining workshops (ICDMW)*. IEEE. <https://doi.org/10.1109/ICDMW.2018.00081>.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., . . . Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (Vol. 30). Curran Associates.
- Zhao, C., & He, Y. (2019). *Auto-EM: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning* (pp. 2413–2424). Association for Computing Machinery. <https://doi.org/10.1145/3308558.3313578>



# Knowledge Graphs



## Constructing, Completing, and Effectively Applying Knowledge Graphs in Tourism

Mayank Kejriwal

### Learning Objectives

- Illustrate the fundamentals of knowledge graphs
- Appreciate the potential of knowledge graphs in relation to Artificial Intelligence and data mining
- Explain how knowledge graphs can be used for tourism applications
- Describe an example of how knowledge graphs can be used on a real tourism dataset using a combination of existing tools

## 1 Introduction and Theoretical Foundations

Although COVID-19 has caused much harm to global tourism, recovery seems to be underway, with researchers already discussing risk mitigation strategies (Fotiadis et al., 2020; Neuburger & Egger, 2020; Yeh, 2020). However, a major beneficial impact of COVID-19 across multiple industrial and educational sectors has been the global acceptance of novel technological solutions. Many of these solutions were initially estimated to first appear at least a decade in the future (McKinsey and Company, 2020). In no small part, these advances have been driven by decade-long advances in “Big Data” (Dong & Srivastava, 2013), calling for more public data releases, a growth of the open-source software ecosystem, and the development of better algorithms in Artificial Intelligence (AI) sub-categories such as computer vision and Natural Language Processing (NLP) (Turian et al., 2010). As such, in the previous decade, impressive breakthroughs in deep neural networks, including

---

M. Kejriwal (✉)

Information Science Institute, University of Southern California, Marina del Rey, CA, USA

e-mail: [kejriwal@isi.edu](mailto:kejriwal@isi.edu)

unsupervised and semi-supervised learning paradigms (Goodfellow et al., 2016), brought about a renewed focus on AI-driven acceleration toward knowledge-based economies (De la Fuente & Ciccone, 2003).

While both data science researchers and industry-based practitioners agree that there is great potential for data science in most disciplines, there has been less focus, in practice, on the non-medical and non-educational service sectors, which often dominate this field. Tourism, despite its economic significance, is an excellent example of an industry in which the importance of data sciences is underestimated and underexposed. In the United States, for example, the travel and tourism industry contributed to 10.4% of global GDP US dollars in 2019 (before the COVID-19 pandemic).

Due to the vast development of new algorithms and the increasing simplicity of applying data driven research approaches, the time is currently ripe to implement cutting-edge data science technologies in tourism in order to deliver better experiences for tourists and analytical solutions for tourism providers. Moreover, as the importance of information provision and transactions via the Internet has increased dramatically, the ever-increasing competitive pressure requires a data-based approach to be applied to the tourism industry as well (Hernández-Méndez & Muñoz-Leiva, 2015).

However, the availability of large, often unstructured, data and more powerful and complex technologies does not necessarily imply a better outcome. Rather, acquiring valuable insights into the data is a more pressing concern. Tourism providers must find, target, and engage their most profitable customers amidst extensive competition and unknown circumstances. For example, suppose that a customer wants to take an authentic countryside river tour in the heartland of the United States. Doing a simple keyword search (e.g., *heartland US river cruise*) may yield some relevant results, but it is very likely that the customer will have to do more considerable research to understand the provided choices. In contrast, when using simpler phrases such as *places to visit in San Jose*, the Google Knowledge Graph can produce a list of results that a user can browse through without even having to click (Singhal, 2012). This is also true for other common queries such as *list of current movies playing in theatres* or *best dystopian films on Netflix*. The question then remains: how can a seamless, low-cognitive/minimal effort search experience be enabled for a customer in the tourism domain?

Based on research and applications in other domains, including e-commerce, a new kind of technology known as a Knowledge Graph (KG) was discussed as a viable solution for solving the problem depicted in the example above. The rest of the chapter proceeds as follows: first, a definition and brief, but informative, theoretical overview of KGs will be provided, followed by a discussion on how KGs have become an important area of research within AI and data science over the past decade. This is evidenced by the launch of mainstream Web-scale efforts such as the Google Knowledge Graph and domain-specific knowledge graphs (Kejriwal, 2019; Paulheim, 2017; Singhal, 2012). Third, the methodology and workflow for constructing a tourism domain-specific KG will be presented. Some hints and tips

on how to facilitate practical implementation will follow, before the chapter finally concludes with a research case study involving KGs.

### 1.1 Fundamentals

A KG represents the world, or a domain of interest, primarily as a collection of discrete entities, attributes, and relations between entity–entity and entity–attribute pairs. Specifically, a KG can be defined in at least two different ways. The first definition, illustrated in Fig. 1, is inspired by graph theory and network science. According to this definition, a KG is a directed, labeled, multi-relational *graph* in which, visually, entities and attributes are represented using ovals and rectangles, respectively. Attributes tend to be literals, such as strings and numbers, but in some communities, such as the Semantic Web, entities are also technically allowed to be URLs.

Since the KG is, first and foremost, a *graph*, it primarily consists of *nodes*, also known as *vertices* or *edges*. Both nodes and edges are labeled. Although not explicitly stated in the definition, labels are expected to be human-readable and represent actual knowledge about the domain. For example, in Fig. 1, if the labels were to be replaced with arbitrary symbols, the formal definition would still be met, but the resulting data structure would not be referred to as a KG in practice. Hence, the labels provide *semantics* to the graph. The topic of semantics in itself is an interesting issue from a philosophical perspective, but a more detailed explanation thereof is beyond the scope of this chapter. Note that similar definitional issues also apply to many other concepts within AI, including questions as to how concepts like *intelligence* or *natural language* can be formally stated (Wang, 2019).

In the 1990s, it was not common practice to think of these KGs in a graph-based theoretical way since graph database research was not very popular yet. In addition, Web search engines were still in their infancy (Grishman & Sundheim, 1996). Over

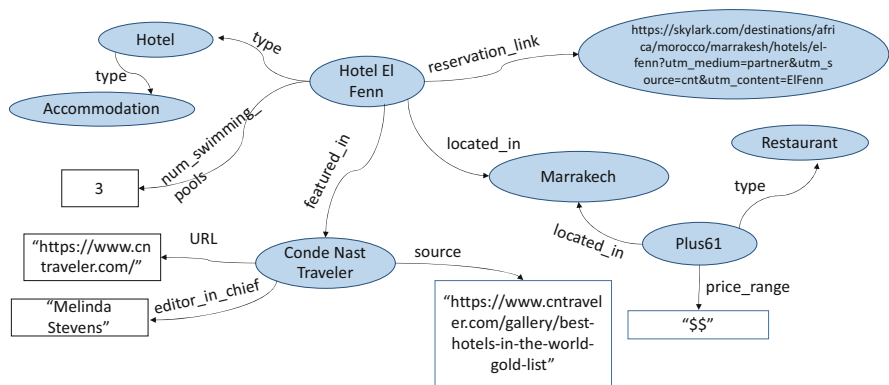


Fig. 1 Example of a KG in tourism. Source: Author’s own illustration

Subject	Predicate	Object
Hotel El Fenn	num_swimming_pools	3
Hotel El Fenn	type	Hotel
...		
Plus61	located_in	Marrakech
Conde Nast Traveler	editor_in_chief	"Melinda Stevens"
...		

**Fig. 2** A portion of the KG fragment in Fig. 1 represented as a set of triples. Source: Author’s own illustration

the next decade, however, the Web grew hyper-exponentially, and entirely new channels of commerce and communication emerged. As a result, Information Retrieval (IR), recommender systems research, and graph databases were all witness to significant interest and curiosity from researchers and practitioners alike (Angles, 2012). This cross-cutting research eventually coalesced into what is now an interdisciplinary KG-centric community (Tiddi et al., 2020). The second definition of a KG, which is more present in the other sub-categories of AI, especially NLP and the Semantic Web, is that a KG is *a set of triples*, i.e., a 3-tuple of the form (*subject, predicate, object*) (Ehrlinger & WöB, 2016). A similar definition is also applied to a so-called *Knowledge Base (KB)*, which, in some contexts, is still the traditional way of referring to what is now often known as a KG (Abburu & Golla, 2017).

While the two definitions above are not identical, they are nonetheless mutually compatible, and most KGs can be described using either definition. A triple in the second definition may be thought of as an edge in the first definition, with the *subject* and *object* serving as the nodes’ labels to which the edge is related. The *predicate* serves as the label of the edge itself, and the directionality of the edge is assumed to be from *subject* to *object*. For the sake of completeness, a portion of the same KG in Fig. 1 is expressed as a set of triples in Fig. 2.

## 1.2 Modeling the Domain

In the cases above, a KG is defined as a generic data structure, but how can a KG be modeled in a *domain-specific* manner? The usual approach is to first define an abstract model of the domain called an *ontology*. An ontology refers to a more general case of a *schema* that is popular in the database community. Within tourism, specifically, a number of ontologies and schemas have been proposed that could be used by others either as a starting point or even as a final model (Chaves et al., 2012; Kärle et al., 2017; Panasiuk et al., 2018).

While the notion and definition of an ontology are very broad, the treatment thereof is based on the practical conception primarily developed in the Semantic Web (SW). In the SW community, an ontology, often synonymous with a *T-Box*, is a set of concepts, predicates, and axioms. Concepts and predicates tend to dominate in real-world ontologies and represent, as the names suggest, the concepts and predicates in the domain of interest. For example, in the tourism domain, appropriate concepts include cruise, hotel, local guide, date, and so on. In Fig. 1, the concepts “Hotel,” “Restaurant,” and “Accommodation” would be considered part of the T-Box. Just as with KGs, predicates are declared between concepts to indicate relationships. Even though axioms and constraints are typically not visually illustrated, when it comes to an ontology, they are allowed. For example, the value constraint that a date should be within a specific range, or must obey a certain format, can be declared as a constraint-based axiom in an ontology.

The SW community has developed extensive literature on how to model ontologies and constraints. The predominant underlying language is the Resource Description Framework (RDF), which serves as the blueprint for more advanced modeling languages such as RDF Schema (RDFS) and the Web Ontology Language (OWL). Examples of pre-defined and commonly re-used ontologies defined in OWL include Dublin Core and SKOS (Miles & Bechhofer, 2009; Weibel et al., 1998). As models, RDF or OWL are quite sophisticated and designed for Web data identified via Uniform Resource Identifiers (URIs). Within SW, a typical domain model or ontology is usually defined using either RDFs or OWL, both of which are technically just built on top of RDF. The actual KG, or, in this case, the so-called *A-Box*, is defined using RDF.

While the nodes in the ontology are equal to *concepts*, the nodes in the KG are considered *entities*. Entities, however, have *types* that bridge the nodes in the A-Box and T-Box using a specific relationship called *is-a*. A triple such as (*Fairmont Hotel*, *is-a*, *hotel*) is an example where the subject is in the A-Box, and is part of the KG, while *hotel* is in the T-Box, and is part of the ontology. The *is-a* link can also be used *inside* the T-Box to declare sub-classes, e.g., the triple (*hotel*, *is-a*, *accommodation*).

In contrast to nodes, *predicates* are usually the same in the A-Box and the T-Box and are subject to constraints. For example, the ontology requires that a customer *stays* in an accommodation. A customer, however, *cannot* stay in a restaurant. Therefore, if the relationship *stays* is declared between a restaurant entity and a customer entity in the A-Box, it will technically violate an ontological constraint. Depending on the expressiveness of the reasoning engine (Sirin et al., 2007), the insertion of a triple may not be allowed into the KG, similar to how some forms only allow a certain range of values to be filled in for fields such as a phone number or zip code.

Although ontologies can be constructed automatically using large text corpora, this practice is non-standard in actual applications. Usually, domain experts define the ontology manually and compactly, using a maximum of a few hundred concepts and a much smaller set of predicates to link these concepts. In some applications, there are fewer than 50 concepts defined in the T-Box. More concepts allow for finer granularity but make an accurate KG harder to construct, as will be discussed in the

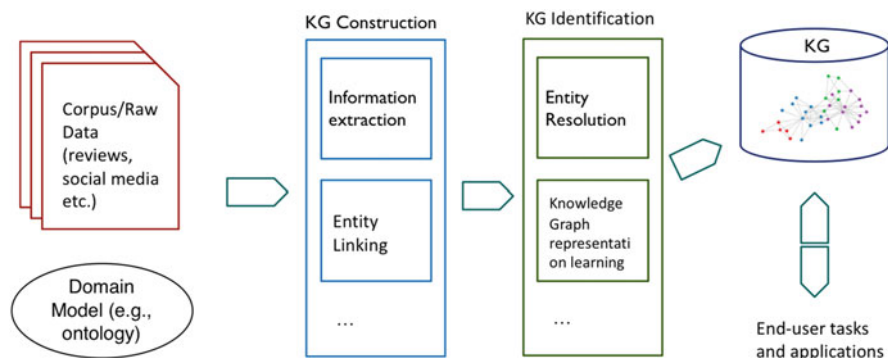
next section. Therefore, a balance is necessary: a more detailed ontology is not always the best strategy, but a model that is too coarse-grained may not provide any advantages over the simple keyword search. A process of trial-and-error is sometimes necessary before a model can be finalized. As such, a model should also always be subject to a periodic review by relevant domain experts. Depending on the application, such experts may range from academics and policy experts to business leaders. Earlier, examples of tourism-related domain models were cited as good starting points for interested users (Chaves et al., 2012; Kärle et al., 2017; Panasiuk et al., 2018).

## 2 Steps Toward Building a Tourism Knowledge Graph

Domain-specific Knowledge Graph Constructions (KGC), where both customers and providers have special needs, is a complex problem that has only recently found a feasible solution. Examples of domain-specific KGs are most evident in the burgeoning area of “AI for social good” (Kejriwal, 2019; Kejriwal & Szekely, 2017) as well as the medicine and government branches (Portisch et al., 2020). The author’s own work, for example, has involved developing KGs for social good, including for fighting human trafficking and isolating informative social media texts in the aftermath of natural disasters (Kejriwal et al., 2018; Strassel & Tracey, 2016; Zhang et al., 2019). Given such success stories in various academic and industrial contexts, tourism companies can make use of feasible and cost-effective KGs to provide deep insights and practical value to customers and internal departments like marketing.

Tourism providers, including small- and medium-sized enterprises, already have an online presence and access to reasonably large quantities of data. Therefore, the key question is how to build a KG from this existing data and more generic sources, such as Wikidata and GeoNames, so as to enable rich customer facing applications. This is not a trivial problem as the tourism industry is complex and interwoven. Besides the service providers, there are thousands of heterogeneous intermediaries that serve various purposes other than just distributing touristic services. The aim is to provide open data access, define data standards for sharing data, and analyze live data.

Figure 3 illustrates a high-level workflow for building a high-quality tourism knowledge graph (Kejriwal, 2019). Prior to describing the KG construction and identification process, selecting or building an appropriate domain model is essential, along with having access to a rich dataset. Concerning the former, an introduction was provided in the *Modeling the Domain* section in addition to existing examples of domain models within the tourism domain (Chaves et al., 2012; Kärle et al., 2017; Panasiuk et al., 2018). The latter depends on the task, application, and resources available. A web portal like Expedia, for example, already has excellent access to both third-party and proprietary data and would likely have a leg-up during this step. In contrast, a new startup looking to disrupt an incumbent or



**Fig. 3** A typical end-to-end workflow for building a domain-specific KG. Source: Author's own illustration

a Destination Management Organization (DMO) would need to either partner up with another source for data or crawl data from the Web. Potential partners can include non-traditional sources like blogs, magazine companies, and social media aggregators, to name but a few. Moreover, researchers also publish relevant datasets on occasion, such as the tourism trajectory datasets used by Wang et al. (2014).

The next and most fundamental step is the KG Construction (KGC). In this step, NLP research proves to play an important role, especially if the raw data is expressed through natural language, such as reviews or social media posts. However, similar tools also apply in cases where the raw data is semi-structured or involving a relational database (Pinto et al., 2003).

## 2.1 Knowledge Graph Construction

Information Extraction (IE) is the single most important step regarding the KGC component (Sarawagi, 2008) and involves the widely studied problem of automatically extracting *named entities*, *relations*, and, possibly, higher-order entities such as *events* from raw text. Only recently has the performance on named entity recognition (NER) achieved very high levels due to the introduction of transformer-based language representation models like BERT. Note that the definition of *what* constitutes a named entity is domain-dependent. In most domains, generic entity types such as *Person* and *Location* are clearly important; however, in the tourism domain, NER systems may have to be designed from scratch for much finer-grained *Accommodation* types (Kärle et al., 2017) in addition to types that are either specific to the domain or to the organization looking to construct the KG (Chantrapornchai & Tunsakul, 2019; Saputro et al., 2016).

Another step that is sometimes optional in other domains, but that should be considered valuable in the tourism domain, is *entity linking* (Ling et al., 2015). Entity linking assumes the existence of a *canonical KB*, such as Wikipedia, that contains

many entities. The issue here is to automatically link an extraction to its canonical entry in this KB. For example, an ideal entity linking algorithm should link “Wimbledon” (the tennis tournament) to the Wikipedia page for Wimbledon. In other words, the algorithm would deduce that the extraction refers to the *tennis tournament* and not the *location* by understanding the context in which the word was used. Yet, since the “surface” form of a word provides limited value for disambiguation, entity linking becomes difficult. A related difficulty is deciding when *not* to link, i.e., to figure out that the extraction does not have an entry in the KB. As such, ignoring this distinction can lead to incorrect links and noise in the subsequent KG. See Chapter “Entity Matching: Matching Entities Between Multiple Data Sources” for more details on entity matching. An additional similar problem, with its own body of literature within NLP, is the notion of *co-reference* or *anaphora resolution*, whereby pronouns and other non-named instances of entities must be linked to their named equivalents in the text, e.g., “He” would have to be linked to the actual named entity it refers to (Elango, 2005).

Concerning the performance of these steps, the most progress has been observed in NER. State-of-the-art deep neural networks, including recent models such as transformers, have achieved excellent performance that can be immediately applied to real-life applications and commercial settings (Gong et al., 2019; Yan et al., 2019). However, beyond NER, the best methods for more advanced NLP components, like relation extraction, event extraction, and, to a lesser extent, co-reference resolution, achieve lower relative performance (Piskorski & Yangarber, 2013). The performance also tends to be less robust, and publicly available, pre-trained resources are not as accessible for these other modules as they are for NER. The problem becomes even more acute in multilingual settings. Furthermore, performance on multimodal datasets, including text and other media such as videos and images, lags behind that of just text (Liu et al., 2019). All of these are ripe avenues for future algorithmic research.

For all of these reasons, a KG can be *noisy* and *incomplete*. By noisy, the implication is that there may be relations and entities in the KG that are somehow erroneous. For example, a relation was extracted between two entities that do not exist or were mistaken for some other relation. On the other hand, incompleteness refers to relations that *do* exist but may not get extracted. Similar problems arise with regard to the entity linking and co-reference resolution settings, e.g., a pronoun that is linkable to some named entity is either not linked to it, or worse, is linked to a different named entity (Ratinov & Roth, 2012).

An even more nebulous problem is that the noise is typically not random: rare, so-called long-tail entities and relations with few instances in the data have a greater probability of either not getting extracted at all or getting extracted by mistake (Nadeau & Sekine, 2007). To measure the extent of this problem, it is vital to use a suite of metrics to measure performance by, for example, having practitioners check both micro-averaged and macro-averaged level accuracy measures to control for label imbalances of any kind (Spyromitros et al., 2008).



## 2.2 Knowledge Graph Identification

Once an initial KG has been constructed, it should still only be considered a rough approximation of the true underlying KG. There are many reasons why this assumption holds true, both in theory and in practice. Perhaps the most important practical reason is the imperfection of KGC itself. As noted earlier, different KGC modules suffer from different degrees of noise and incompleteness. However, the noise and incompleteness introduced into the KG are neither random nor independent. Therefore, the goal is to detect these sources and to repair the actual underlying KG (Pujara et al., 2013). This process is called *knowledge graph identification*. As such, it systematically uses higher-level features from the initial knowledge graph, including structural features and domain knowledge encoded via probabilistic graphical models or representation learning (Wang et al., 2017). Below, different aspects of KG identification are briefly discussed.

Theoretically, even if all steps worked properly and did not contain noise and/or incompleteness, two additional problems may arise that warrant further processing of the initial KG. First, relations might be implicit. For example, the fact that Marrakech is in Morocco may never be exerted explicitly, yet could be *implied* by different pieces of data in the initial KG. This provides an opportunity for a reasonably powerful algorithm to discover the relation and insert it into the KG as an edge or triple (Shi & Weninger, 2018). Second, because of the complexity of IE and other steps like co-reference resolution, it is still typical to go through them on a *per-document* basis. Hence, it is virtually guaranteed that there will be redundancies in the KG, i.e., the same entity or fact may get extracted multiple times, sometimes with different surface forms, from different documents. Thus, it is important to de-duplicate and apply the process of canonicalization to these extractions.

Entity Resolution (ER) is another important step that can be undertaken for KG identification (Pujara et al., 2013). Simply put, ER is the problem of algorithmically identifying when two or more entities refer to the same *underlying* entity. A simple example is the pair of entities “UN” and “United Nations,” which clearly refers to the same underlying entity. While the problem seems simple, it has proven to be quite a complication for creating general solutions. Over the last 50 years, therefore, various supervised, semi-supervised, and unsupervised solutions have been proposed across communities (Kejriwal, 2016). As previously mentioned, more recently, the advent of language representation-learning models like BERT has led to significant performance enhancements, assuming the domain is not too esoteric (Devlin et al., 2018). While not much literature exists on applying such language models to ER on large, domain-specific KGs, algorithms in related problem domains can be adapted. With good ER, unnecessary and noisy redundancies in the KG can be removed, facilitating more accurate querying.

More advanced examples of KG identification include link prediction and triples classification. Concerning the former problem, this applies when predicting new edges between existing nodes in the KG and is an essential concern especially in social KGs. The latter problem is primarily relevant for removing noisy triples,

although it can be adapted to give “scores” to triples as well. In an age of misinformation, this may be an important application area for some domains. The state-of-the-art techniques used for solving these higher-order identification problems include some form of KG representation learning (Wang et al., 2017). Similar to word2vec and language representation-learning models (Pennington et al., 2014) (for more details on Text Representation and Word Embeddings, see Chapter “Text Representations and Word Embeddings”), the goal of KG representation learning is to embed nodes and relations in the KG into a dense, continuous vector space. Using vector operations, like translation, or by using the vectors as feature vectors in an ordinary machine learning pipeline, issues involving, for instance, link prediction and triples classification can be effectively addressed (Wang et al., 2017).

### 2.3 Storing, Querying, and Using the Knowledge Graph

Due to the growth and popularity of big datasets, a KG constructed over such datasets, using the workflow in Fig. 3, is also expected to be large. In this sense, a KG could, for example, contain hundreds of thousands, if not millions, of entities. The number of triples is usually an order of magnitude or is greater than the number of entities. Note that the number of unique relations tends to stay relatively small, typically fewer than one hundred, and, as noted earlier, are defined in the domain model (a.k.a. an ontology or T-Box). In turn, a pragmatic question arises as to how such a KG can be stored, queried, and used.

Knowledge graphs can be stored in *graph databases*, such as Neo4j,<sup>1</sup> or in *triple stores*, such as Apache Jena<sup>2</sup> and Amazon Neptune.<sup>3</sup> Several authors have already explored these options for tourism (Chareyron et al., 2020; Yochum et al., 2018), with triple stores having been widely investigated in the Semantic Web and being considered the default infrastructure. In contrast, Neo4j has been the subject of many studies in both industry and database communities, including the tourism domain. Primarily SPARQL is the language used to access and query triple stores, while the graph databases tend to have their own domain-specific languages, such as Cypher for Neo4j (Chareyron et al., 2020). If the KG is enormous, neither option may scale well, and practitioners may want to consider using Big Data architectures like Apache Hive or Cassandra (Chen & Zhang, 2014). It is rare, however, to encounter such big KGs that render those types of architectures necessary. Other key-value stores like MongoDB and Elasticsearch have also been used to store KGs, although representing KGs in such non-native formats is rather non-intuitive.

While choosing a storage and querying infrastructure, a practitioner must look at the size, query complexity, and latency, in addition to other factors, as needs arise.

---

<sup>1</sup><https://neo4j.com/>

<sup>2</sup><https://jena.apache.org/>

<sup>3</sup><https://aws.amazon.com/neptune/>

Queries that are very complex may require a triple store or graph database. In general, it is challenging to execute such queries using any of the other options, such as a key-value or Big Data store, without significant engineering and expertise. In contrast, if IR-style queries, such as those facilitated through keyword search, are all that the KG is expected to serve in an application, a key-value or document store is clearly the best option as it may offer intuitive horizontal scaling and indexing capabilities (Seeger & Ultra-Large-Sites, 2009). Yet, it must be emphasized that there is no one right or wrong infrastructure; the choice should be guided by the requirements of the end-application and the properties and quality of the KG. For example, if a KG is *low-quality*, a triple store should be *less* preferred over a key-value database.

Armed with fundamentals as well as KGC methodology, it is fruitful to revisit why a KG is expected to be useful in practical problem settings. Returning to the example scenario in the introduction about a customer searching for a *heartland US river cruise*—if the search engine in the website is optimized to return results from a KG *constructed* with tourism-specific data, the website would not have to rely on keyword matching, whether directly or in a multi-hop fashion. Instead, it should be able to recognize *heartland US* as referring to a *set* of places or, even more specifically, cities and counties in certain states. Furthermore, it should also recognize *river cruise* as a tourism product or service. At a minimum, the querying program should know not to interpret *river* and *cruise* as independent of one another since they imply different kinds of services when used separately. For example, *cruise*, on its own, would likely imply a more traditional ocean-based cruise rather than a river cruise. In this way, the key advantage of modeling and building a tourism-specific KG is that a query interface will be capable of making these distinctions because the KG has been modeled and constructed for that specific domain.

Once the querying algorithm has made these connections with the underlying KG, it can even retrieve relevant factual information about that entity. Such information includes *other* entities connected to the entity in the corpus, such as *countryside US packages with river cruise* as well as the relationships connecting these entities and their properties together (Liu et al., 2014). In the best-case scenario, the domain-specific search engine that powers the website or tourist provider may even end up persuading the customer to purchase a more profitable or comprehensive package that the customer had previously not even known existed. In short, KGs allow rich querying and search applications to be unlocked, which further enable greater monetization. E-commerce giants such as Amazon have realized the potential of this monetization as their inventory and choice of products continues to expand. Without good search technology, it becomes increasingly difficult for customers to find what they are looking for (Dong, 2018). In contrast, with such enabling technology, companies can continue to grow and tap into more significant and distinct markets without having a negative impact on customer experience. Search engines like Google presumably invest in this technology for similar reasons; if customers have an easier time finding what their looking for, it is possible to

monetize that intent more effectively, especially with targeted, personalized advertising (Singhal, 2012).

Although entity-centric search is the primary use case for most known KG applications, other use cases are also possible and are currently being explored in some domains such as scientific research mining, journalism, and recommender systems (Berven et al., 2018; Jiang & Shang, 2020; Lü et al., 2012). One such use case that has emerged is the use of KGs in question answering (QA) and digital personal assistants such as Siri and Alexa. While the technologies behind these personal assistants are primarily proprietary and many details are unknown, the use of KGs to boost performance on everyday QA tasks, an example being “is it going to rain today?”, has been well-established by now. Excellent performance has also been achieved on “commonsense” QA benchmarks that involve social and physical situations (Vollmers et al., 2021). In this space, the birth of large-scale language representation models such as BERT, GPT-3, and, more recently, Switch, which has more than a trillion parameters, has been an exciting advance (Devlin et al., 2018; Fedus et al., 2021). These models may lead to synergies with KG research that, even a few years ago, were considered futuristic. Researchers from Amazon have also spoken quite extensively about the Amazon Product Graph, which is related to search but may also facilitate other applications such as product categorization and clustering (Dong, 2018).

Overall, state-of-the-art KG technology is designed to be algorithmically scalable; another customer searching for a completely different product or service will be able to retrieve answers in real-time, assuming the software has been properly implemented, engineered, and configured. Astoundingly, modern research has even progressed to the point where it is possible to query a KG in natural language by using similar NLP technologies as those implemented in modern chatbots and personal digital assistants such as Siri rather than a formal query language (Maheshwari et al., 2019). Nonetheless, this is still very much an active area of research.

### 3 Practical Demonstration and How-To Guidelines

KG research has clearly come a long way, but practitioners are more frequently interested and concerned about how this research has been translated into usable tools. Unsurprisingly, although some components established in Fig. 3 have good toolkit implementations, others require customized coding and additional effort. “Gluing” together the different outputs or implementations, many of which have their own format and dependencies, is a non-trivial challenge that requires software engineering. While there has been some progress within the KG community, making it easier for a team of domain experts with some software engineering experience to build an end-to-end KG system, there is still much work to be done.

On that note, this section aims to cover some how-to implementation guidelines, with hints and tips provided in the next section. Moreover, the software referenced in

this section is then further described in the *Available Software/Solutions* section. Note that the actual usage of each software always depends on the context in which the KG is being implemented, including the end-application and available resource set. By way of example, consider the simple case where the raw data is publicly available and already well-structured, allowing a KG to be constructed without significant Natural Language Processing. The *Research-Case* section thus covers a piece of work that, for a recommendation problem, aimed to do precisely that. Here, a step-by-step overview of a more simplified pipeline is provided in order for a data scientist to get started on the problem.

**Step 1: Data Acquisition** Based on the workflow in Fig. 3, domain modeling and data acquisition are critical preliminary steps to take care of before a KG can be constructed. In the absence of proprietary data, it is best to use public datasets with a broad scope. Typically, these include datasets derived from either crowdsourced knowledge or an “encyclopedic” source such as Wikipedia. Examples include GeoNames, DBpedia, and Wikidata. GeoNames describes locations, administrative regions, and geographical entities, whereas DBpedia contains Wikipedia “infoboxes” and abstracts, and Wikidata is a crowdsourced KG.

In this section, DBpedia will be the primary focus (with links and further information also provided in the *Available Software/Solutions* section). While the DBpedia KG is available in multiple formats, the most straightforward format to work with is the *n-triples*, which has an *.nt* extension and is an RDF KG. This file may be downloaded from the website in the language of the user’s choice (<https://wiki.dbpedia.org/develop/datasets>). Thereafter, when an *.nt* file is opened or previewed in a text editor, each uncommented line in the file should be assumed to correspond to an “edge,” conforming to the definition of a KG as a set of triples.

Without even downloading the dataset, the code snippets below can show you how to get started. The full tutorial is available on the book’s Github-Profile and includes a “requirements” file containing the names of the packages that need to be installed for the code to work. The code below is from the *Building\_KGs.ipynb* notebook.

As a first step, we import the packages that we will need for the exercise.

```
# importing all the required packages
import spacy
from spacy.lang.en import English
from SPARQLWrapper import SPARQLWrapper, JSON
import networkx as nx
import matplotlib.pyplot as plt

nlp = spacy.load('en_core_web_lg')
%matplotlib inline
```

If all requirements have been installed correctly, the imports should be successful.

Next, we specify the city of the data we want to retrieve from DBpedia and execute the function below.

```
# Enter the City to construct the Knowledge Graph (ensure correct spelling)

city = 'Bangalore'

# This function currently retrieves ABSTRACT, POPULATION, COUNTRY and ALIASES for the city,
# it can be extended to fetch any other information from DBpedia that is required.

def fetchData(city):
    '''
    input: string : city name
    output: dict : dictionary containing the results of city information
    from DBpedia

    This function uses SPARQL endpoint to extract information about the
    given city
    from DBpedia in JSON format.
    '''

    sparql = SPARQLWrapper("http://dbpedia.org/sparql")
    sparql.setQuery(
        """
        PREFIX dbr: <http://dbpedia.org/resource/>
        PREFIX dbo: <http://dbpedia.org/ontology/>
        PREFIX foaf: <http://xmlns.com/foaf/0.1/>

        SELECT ?name, ?city, ?abstract, ?population, ?country, ?alias, ?
        timezone
        WHERE {{
        ?city a dbo:City .
        ?city rdfs:label ?name .
        ?city dbo:abstract ?abstract .
        ?city dbo:populationTotal ?population .
        ?city dbo:abstract ?abstract .
        ?city dbo:country ?country .
        ?city rdfs:label {0}@en .
        ?city foaf:name ?alias .
        ?city dbo:timeZone ?timezone .
        FILTER ( langMatches(lang(?abstract), "en") ) .
        FILTER ( langMatches(lang(?name), "en") ) .
        }}
        """ .format(city)
    )

    sparql.setReturnFormat(JSON)
    results = sparql.query().convert()

    return results
```

The rest of the tutorial in *Building\_KGs.ipynb* shows how to use this data to construct KG fragments. We provide more conceptual background in the steps below.

**Step 2: Domain Modeling** The next step is to download and use a domain model, or ontology, to which the KG will be constructed. Tourism-specific ontologies and schemas are currently available and can be applied by the more familiar user (Chaves et al., 2012; Kärle et al., 2017; Panasiuk et al., 2018), but a more convenient option may be to download the DBpedia ontology from the website. While this is a “generic” ontology, covering a broad set of concepts and predicates, including *Locations*, *Music*, *Films*, and so on, it is well documented and reasonably sized. The ontology, available as an OWL file, can be opened with a text editor, yet it is better to process it using a tool called Protégé. The tutorial on Protégé, also available on the website, is highly recommended as it contains powerful facilities including support for reasoning and ontology manipulation. Although the Protégé web interface can also be used, installing the software locally is best for those looking to play around with KGs and ontologies.

For beginners, only try to import the ontology and get familiar with it in Protégé, without modifying it. The advanced user could try to “refine” the ontology, but this should not be necessary as good ontologies already exist in tourism (cited above). An even more advanced user could try to import more than one ontology into Protégé and combine the best of both. However, this degree of merging is something that, in practice, is best left to a knowledge engineer or ontologist.

**Step 3: Knowledge Graph Construction (KGC)** When the raw data involves natural language data, the most important KGC step is information extraction. Although this exercise does not deal primarily with natural language data, interested practitioners can still play with it by downloading the *DBpedia Abstracts* corpus in the language of their choice. Next, the user should install SpaCy (see *Available Software/Solutions*). There is also a convenient demo interface available on the SpaCy website that allows the user to try out various NER models and use small pieces of example text to verify performance. Results obtained from the DBpedia abstract for the city of Leipzig, using the “en\_core\_web\_sm” model, are illustrated in Fig. 4. The reader is further referred to Chapter “Natural Language Processing (NLP): An Introduction” in which NER is described in greater detail.

For the user not wishing to delve deeper into NLP but prefers to continue working with the n-triples files downloaded earlier, an *entity linking* exercise is recommended instead. Recall that entity linking assumes the existence of a canonical KB. In this case, it is natural to think of DBpedia as the KB. The goal, then, is to link entities in GeoNames and Wikidata to entities in DBpedia. GeoNames is a valuable resource to start with as a ground-truth set of links is already available, as described in *Available Software/Solutions*. Figure 5 thus illustrates a fragment thereof. This ground-truth is useful because it helps a user evaluate different algorithms using metrics such as precision and recall. Given a set of labels and predictions, code for computing precision and recall can be found in a number of packages, including scikit-learn.

Leipzig **ORG** (ˈlɛʲˌlʌɛ̯ˈptsɛ̯ˈɛiː; [ˈlʌɛ̯ˈpt͡ɕisɛ̯ˈʌʂ]) is a city in the federal state of **Saxony** **GPE** , **Germany** **GPE** . It has a population of 551,871 inhabitants (1,001,220 residents in the larger urban zone). **Leipzig** **GPE** is located about 150 kilometers (93 miles) south of **Berlin** **GPE** at the confluence of **the White Elster** **ORG** , **Pleisse** **GPE** , and Parthe rivers at the southerly end of **the North German** **NORP** Plain. **Leipzig** **GPE** has been a trade city since at least the time of **the Holy Roman Empire** **GPE** . The city sits at the intersection of **the Via Regia** **ORG** and **Via Imperii** **ORG** , two important Medieval trade routes. **Leipzig** **ORG** was once one of the major **European** **NORP** centers of learning and culture in fields such as music and publishing. **Leipzig** **ORG** became a major urban center within **the German Democratic Republic (East Germany)** **GPE** ) after World War II, but its cultural and economic importance declined despite **East Germany** **GPE** being the richest economy in **the Soviet Bloc** **LOC** . **Leipzig** **GPE** later played a significant role in instigating the fall of communism in **Eastern Europe** **LOC** , through events which took place in and around **St. Nicholas Church** **ORG** . Since the reunification of **Germany** **GPE** , **Leipzig** **GPE** has undergone significant change with the restoration of some historical buildings, the demolition of others, and the development of a modern transport infrastructure. **Leipzig** **GPE** **today** **DATE** is an economic center, the most livable city in **Germany** **GPE** , according to the GfK marketing research institution and has a prominent opera house and one of the most modern zoos in **Europe** **LOC** . **Leipzig** **GPE** is currently listed as **Gamma World City** **ORG** .

Fig. 4 Performance of a small pre-trained SpaCy model on an abstract for the city of Leipzig. Source: Author’s own illustration

```

<http://dbpedia.org/resource/Neanderthal42C_Germany> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2866829/> .
<http://dbpedia.org/resource/Qingdao> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/1797929/> .
<http://dbpedia.org/resource/Molins> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/3081472/> .
<http://dbpedia.org/resource/Podlachian_Voivodeship> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/858789/> .
<http://dbpedia.org/resource/Pl1K31A4ming> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2926388/> .
<http://dbpedia.org/resource/Brusbane> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2174003/> .
<http://dbpedia.org/resource/Lommel> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2800931/> .
<http://dbpedia.org/resource/Birobidzhan> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2026643/> .
<http://dbpedia.org/resource/Utrecht_128province129> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2745912/> .
<http://dbpedia.org/resource/Utrecht_128city129> <http://www.w3.org/2002/07/owl#sameAs> <http://sws.geonames.org/2745912/> .

```

Fig. 5 A fragment of the DBpedia-GeoNames ground-truth links. Source: Author’s own illustration

Naturally, this still asks the question of how one would actually *complete* entity linking. For the beginner, simple string matching is recommended. The correct way to do this would be to use a Python program to parse each triple in the DBpedia and GeoNames n-triples files and collect the label of each entity in each file. Sometimes, it is possible to “hack” the URL, as in the case of DBpedia, to retrieve the label; for



example, it is easy to write a script to parse the label “Turku” from the URL “<http://dbpedia.org/resource/Turku>”. However, as the fragment in Fig. 5 shows, this kind of hacking is not possible for GeoNames. Hence, getting the label from the GeoNames n-triples file is a good exercise for working with KGs as a set of triples. It may take some practice, but, if done right, the actual code is quite simple.

In this regard, there are many good string matching packages available; a simple one, implemented in Python, is PyPi Edit Distance in which users can express their data as two sets of string labels, e.g., from GeoNames and DBpedia. To avoid scalability problems, they can start with sets containing, at the most, 10–15 strings each. The *Entity\_Resolution.ipynb* file in the tutorial linked earlier describes such an exercise using PyPi Edit Distance, but, for completeness, the following code snippet is also provided below.

```
import edit_distance
import pandas as pd
actual_names = ['Los Angeles', 'New York City', 'Bangalore', 'Mumbai',
                'Chennai', 'Kolkata', 'New Delhi', \ 'Saint Petersburg',
                'Melbourne', 'Gothenburg', 'Vienna', 'Barcelona',
                'Las Vegas']

input_names = ['City of Los Angeles', 'New York', 'Bengaluru', 'Bombay',
               'Madras', 'Calutta', 'Delhi', \ 'St. Petersburg',
               'Melborne', 'Goteborg', 'Wien', 'Barca', 'Las Vegas']
def edit_dist_metrics(actual_names, input_names, threshold):
    '''
    input: list : actual_names list
    list : input_names list
    float : threshold value for similarity score 0 <= threshold <= 1

    The function compares every string in actual_names with every string in
    input_names using edit_distance and provides a similarity score. If the
    score is more than or equal to a given threshold, the two strings are
    matched as the same entity and compared with ground truth results.
    The results, precision and recall is printed out.
    '''
    res = []
    for i, a_name in enumerate(actual_names):
        for j, i_name in enumerate(input_names):
            r = edit_distance.SequenceMatcher(a_name.lower(), i_name.lower()).
            ratio()
            if r >= threshold:
                res.append([i_name, a_name, r, i==j])

    df = pd.DataFrame(res, columns=['Input Name', 'Predicted
    Name', 'Similarity Score', 'Ground Truth'])
    precision = round(sum(df['Ground Truth'])/len(df), 3)
    recall = round(sum(df['Ground Truth'])/len(actual_names), 3)
    print(df, '\n')
    print("Precision: "+str(precision))
    print("Recall: "+str(recall))
    return
```

Following the initial setup and experimentation, the user can then play with the edit-distance outputs and try various “thresholds” in order to decide when two entities should be linked. Then, the results should be compared to the actual ground-truth using precision-recall metrics. The code snippet below shows what happens/what results appear when the function is executed with a threshold of 0.4:

```
edit_dist_metrics(actual_names, input_names, 0.4)
```

```

Input Name Predicted Name Similarity Score Ground Truth
0 City of Los Angeles Los Angeles 0.733333 True
1 Las Vegas Los Angeles 500000 False
2 New York New York City 0.761905 True
3 Bengaluru Bangalore 0.666667 True
4 Barca Bangalore 0.428571 False
5 Bombay Mumbai 0.500000 True
6 Calutta Kolkata 0.428571 True
7 New York New Delhi 0.470588 False
8 Delhi New Delhi 0.714286 True
9 St. Petersburg Saint Petersburg 0.800000 True
10 Goteborg Saint Petersburg 0.416667 False
11 Melbourne Melbourne 941176 True
12 St. Petersburg Gothenburg 0.416667 False
13 Goteborg Gothenburg 777778 True
14 Bengaluru Vienna 0.400000 False
15 Wien Vienna 0.600000 True
16 Melbourne arcelona .470588 False
17 Barca Barcelona 0.714286 True
18 Madras Las Vegas 0.400000 False
19 Las Vegas Las Vegas 1.000000 True

```

```
Precision: 0.6
```

```
Recall: 0.923
```

**Step 4: Knowledge Graph Identification** This is one of the more advanced steps, recommended for a user who, at this stage, is more comfortable with KGs. The user can use a software package such as OpenKE to “embed” the DBpedia KG, with or without entity linking, and visualize some vectors using t-SNE to get a sense of how deep learning can help with KG identification. More details are also provided in Chapter “Dimensionality Reduction.” For those who are more comfortable with deep learning, link prediction can be used to split the set of DBpedia triples into a training and testing set. The embeddings can be learned using the training set triples, and the testing set should be used to evaluate link prediction performance. The best metric for the beginner is *accuracy*. However, more advanced users can use Hits@10 and IR-based metrics like the Mean Reciprocal Rank. Following the embedding, these metrics can then assess whether the model, given just  $s$  and  $p$  from a test-set triple  $\langle s, r, p \rangle$ , is able to correctly predict  $r$  as the most probable relation.

**Step 5: Storage, Querying, and Use** This step is simple enough that it can be done on the DBpedia n-triples file without even completing Steps 2–4, although the overall purpose of this exercise is to do it on the “merged” KG that emerges from doing all steps above (by first linking GeoNames to DBpedia and, then, as an even more advanced exercise, linking Wikidata to DBpedia following an analogous set of steps). Rather than setting up a local querying infrastructure, a starting user can sign up for a free account on Amazon Web Services and use Amazon Neptune. There are excellent tutorials available that guide the user through the process of importing their KG to the cloud and querying it using SPARQL. At this point, the OWL ontology may also be helpful as it provides a summary of classes and properties in the KG. For users looking to become more familiar with SPARQL, *DBpedia Online Access* allows users to try SPARQL queries in relation to DBpedia and is a valuable resource, similar to diSplaCy, illustrated earlier in Fig. 4. More details are provided in *Available Software/Solutions*.

The *Research-Case* section provides more details on how the KG can also be used in recommender systems. While recommendation is an advanced application for KGs, the case provides practical guidance on enabling a recommender-based application. More broadly, because the individual steps described above are complex, there is an unmet need for end-to-end, easy-to-use frameworks for building KGs. Such frameworks are useful even if they only serve a limited set of applications. A possible solution called DIG (Kejriwal, 2021; Kejriwal & Szekely, 2019) is described in *Available Software/Solutions*, but prior to using such end-to-end solutions, it is recommended to attempt the simpler workflow illustrated above.

### 3.1 Hints and Tips

Based on the author’s experience with building domain-specific knowledge graphs, a set of hints and tips is provided below for practitioners looking to implement KGs for their own use cases and datasets:

<p><i>Have relatively simple, but effective, baselines for the more critical steps.</i></p>	<p>As discussed in the previous steps, there are many moving parts when it comes to knowledge graph construction, and many hundreds, if not thousands, of papers have been published on these steps. Therefore, it is tempting to spend time downloading and/or implementing (or, if already available as open-source, <i>tuning</i>) the latest tools in the hopes that they may provide an edge on the use case of interest. In practice, however, it has often been found that simple baselines can sometimes be harder to beat, if implemented properly. Only in some cases, and for specific domains, are significant and consistent improvements observable. This does not imply that the research is invalid, only</p>
---	---

(continued)

	<p>that it may or may not be appropriate for the domain of interest. Hence, practitioners are well-advised to adopt a <i>build-the-baseline-first</i> methodology and to incrementally and carefully improve each step. An advantage of this methodology is that it leads to faster development of an end-to-end system that can be progressively improved and evaluated over time.</p>
<p><b><i>Use multiple evaluation metrics and carefully identify the steps in the workflow from Fig. 3. Most likely to acquire annotations and/or algorithmic investments.</i></b></p>	<p>Since knowledge graphs, and the systems used to build them, are complex, there is no one metric to evaluate performance. This is true even for an individual component, such as information extraction. Often, the metric is task-dependent. For example, some tasks, such as a ranking-based search, require a more balanced tradeoff between precision and recall, whereas others are willing to tradeoff recall for higher precision. It is unclear, however, whether improvements in individual components lead to an overall improvement. It may very well be that even significant improvements in, for example, information extraction do not lead to <i>proportionate</i> improvements in the end-application. Nevertheless, the opposite may be observed for entity resolution, i.e., minor improvements may end up exhibiting increasing returns on end-application performance. These empirical observations can be used to guide investment in algorithmic tuning and annotation effort. In some cases, a best-of-both-worlds scenario may be possible by using frameworks like active learning and weak supervision.</p>
<p><b><i>Take advantage of open-source data and software.</i></b></p>	<p>One of the greater benefits of the knowledge graph ecosystem, both in research and in practice, has been the emergence of a large open-source community. This trend is independent of KGs, since, in the last ten years, there has been a general movement toward releasing and making more data and software available for various reasons. Today, many governments also make, at least a part of, their data public. This is in addition to companies and organizations that collect and sell data. Within the KG and SW community, linked open data (LOD), a 13+ year initiative, has led to the publication of hundreds of interlinked KG-like datasets on the web, including geographic knowledge bases like GeoNames, encyclopedic datasets like DBpedia and YAGO, and domain-specific datasets like PubChem. Some of these open datasets can be</p>

(continued)

	useful for the tourism domain (e.g., GeoNames); nevertheless, where possible, it is recommended to leverage this diversity of open-source data and software in any practical pipeline.
<b>Have straightforward end-user applications in mind.</b>	Before building a KG, the organization must understand <i>why</i> the building of a KG is necessary. Typically, it is best to have a primary <i>must-have</i> end-application in mind, along with a small set of secondary or <i>nice-to-have</i> use cases. A must-have application could be, for example, better recommendation capabilities on the organization’s website. Like any technology, it may well be that a secondary use case ends up superseding the primary use case over the course of the development. Therefore, uncertainty is not necessarily undesirable. However, project managers should have one or more strong applications or goals in mind that the KG is expected to play a major role in. Otherwise, there is a risk of the project derailing when resources become more constrained.

## 4 Research Case

Lu et al. (2016) describe how they performed “travel attraction recommendations” with knowledge graphs, using a more advanced version of the example given in the step-by-step description in the *practical demonstration*. First, they describe a semi-automatic method, not consisting of too much manual effort, to construct a “world-scale travel knowledge graph.” Rather than using natural language data, they use KGs, such as GeoNames, DBpedia, and Wikidata, as their primary sources. Second, they present a “city-dependent user profiling strategy” that uses the semantics in the constructed KG to understand travelers’ interests better and provide recommendations.

Before going into details, it is helpful to review the hints and tips mentioned earlier. One of the tips was to have straightforward end-user applications in mind. This is something that is clearly embodied in this paper. In other words, the authors are not just attempting to build a KG for the sake of it but, rather, have an explicit end goal in mind. As obvious as this may seem, it is an important consideration to keep in mind when investing time and money into building a KG.

The first step the authors took was to retrieve the pool of travel attractions. This corresponds to Steps 1 and 2 that were described earlier in the chapter; essentially, the authors used the DBpedia property *dct:subject* and a regular expression to decide

which categories to keep in DBpedia. An example SPARQL query that they ran as well as steps for further processing are described in Sect. 3.1 of their paper.

Similar to the step-by-step instructions provided earlier, Lu et al. (2016) did not use NLP modules like information extraction. Instead, they designed specific modules to extract relevant data from DBpedia, and, for Wikidata, they leveraged the “ground-truth” or *owl:sameAs* links from DBpedia to Wikidata, along with some minor post-processing adjustments. However, in many challenging cases, such links may not be available or may be noisy or incomplete, which would require steps such as entity linking and resolution. As this kind of research is advanced and includes many processing steps, it is difficult to find papers that describe the individual steps in detail, which, in turn, has a detrimental effect. Within the industry, full architectures exist, especially in e-commerce companies like Amazon, but, with very few exceptions, are typically not published or discussed openly (Lockard et al., 2019).

For storage and querying, the authors used Neo4j, which was presented as one of the many viable options in the *Storing, Querying, and Using the Knowledge Graph* section. Once the knowledge graph was stored in a Neo4j infrastructure, the authors described the recommender system in detail.

Finally, the authors evaluated their knowledge graph-based recommender using the Yahoo! Flickr Creative Commons 100 M (YFCC100M) dataset. Since the entire dataset contains 100 million Flickr photos and videos, the authors only used a subset thereof. Specifically, computer vision tasks like object recognition were not within the scope of the paper; therefore, instead, they retained geotagged photos and videos with the highest geo-location accuracy, and mapped each such photo/video to a point-of-interest (POI) entity in their travel KG. Their approach offers a great example of how publicly available resources can be used to construct ground-truths and evaluation datasets for assessing KG-based applications, even though the resources were not originally designed to support such evaluations. To evaluate the recommender system, they used multiple metrics (as also suggested as a hint in the previous section), namely, *precision*, *recall*, *F1-Score*, and the *Normalized Discounted Cumulative Gain (NDCG)*, the last of which is a crucial IR metric.

### Service Section

**Main Application Fields:** Knowledge Graphs (KGs) play an instrumental role in Web search as well as in developing web agents and the Semantic Web. More recently, applications have also included AI for social good, including building KG-based domain-specific search engines for crisis informatics and human trafficking, e-commerce, and, increasingly, manufacturing and services industries. More generally, because KGs represent data in a consistent, structured format, they may be applied in many modern information and knowledge retrieval applications, including tourism.

(continued)

**Limitations and Pitfalls:** A key limitation is the difficulty in building a full end-to-end system. As argued in *Hints and Tips*, it is tempting to focus too much on optimizing a single component at the cost of getting an end-to-end system ready and evaluating it (Kejriwal et al., 2021). Another problem is that some of the tools may not work as well with multilingual and multimodal data, although considerable advances have been established in the last decade. In addition, scale may also be a problem, but this can be addressed by a competent knowledge engineer. Finally, it is important to note that a KG is only as good as its underlying data sources. Hence, it is important to develop a good crawling system for data that is scraped from the Web. Novak (2004) provides a good overview of web crawling algorithms. For focused crawling, the interested reader is referred to the work by Talvensaaari et al. (2008).

**Similar Methods and Methods to Combine with:** Natural Language Processing (NLP) methods and applications, such as sentiment analysis, semantic role labeling, and question answering, are all complementary areas to be added to KGs. They can feed into a KG as additional sources of information and, conversely, be used to access KGs, e.g., in the case of question answering. Representation learning, as discussed earlier in *KG Identification*, is also closely connected to KG research.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-20-Knowledge-Graphs>

## Further Readings and Other Sources

2nd International Workshop on Knowledge Graphs on Travel and Tourism: <https://tourismkg.github.io/2019/>

Google Knowledge Graph: <https://www.youtube.com/watch?v=mmQl6VGvX-c>  
*Knowledge Graphs: Fundamentals, Techniques and Applications*. Kejriwal, Knoblock and Szekely (Kejriwal, Knoblock, and Szekely, 2021): <https://www.amazon.com/Knowledge-Graphs-Fundamentals-Applications-Computation-ebook/dp/B08C75T8JF>

Kejriwal, M. (2019). *Domain-specific knowledge graph construction*. Springer International Publishing.

For surveys on focused crawling and domain discovery works by (Novak, 2004) and (Talvensaaari et al., 2008) are recommended.

To learn more about OWL and RDF, the interested reader is referred to a number of classic references on the subject (Berners-Lee and Hendler, 2001; Antoniou and Van Harmelen, 2004).

Online source: <https://towardsdatascience.com/knowledge-graph-bb78055a7884>

The Semantic Web (Scientific American): <https://www.scientificamerican.com/article/the-semantic-web/>

Our group website has numerous examples of KG applications and projects: <https://usc-isi-i2.github.io/home/>

## References

- Abburu, S., & Golla, S. B. (2017, October). *Ontology and NLP support for building disaster knowledge base* (pp. 98–103). In 2017 2nd International conference on communication and electronics systems (ICCES). IEEE.
- Angles, R. (2012, April). *A comparison of current graph database models* (pp. 171–177). In 2012 IEEE 28th International Conference on Data Engineering Workshops. IEEE.
- Berven, A., Christensen, O. A., Moldeklev, S., Opdahl, A. L., & Villanger, K. J. (2018, September). News Hunter: building and mining knowledge graphs for newsroom systems. In *Norsk konferanse for organisasjoners bruk av IT* (Vol. 26, No. 1).
- Chantrapornchai, C., & Tunsakul, A. (2019, July). *Information extraction based on named entity for tourism corpus* (pp. 187–192). In 2019 16th International joint conference on computer science and software engineering (JCSSE). IEEE.
- Chareyron, G., Quelhas, U., & Travers, N. (2020, January). *Tourism analysis on graphs with Neo4Tourism* (pp. 37–44). In International conference on web information systems engineering. Springer.
- Chaves, M., Freitas, L., & Vieira, R. (2012). *Hontology: A multilingual ontology for the accommodation sector in the tourism industry*. SciTePress.
- Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275, 314–347.
- De la Fuente, A., & Ciccone, A. (2003). *Human capital in a global and knowledge-based economy* (Vol. 918). Office for Official Publications of the European Communities.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Dong, X. L. (2018, July). *Challenges and innovations in building a product knowledge graph* (pp. 2869–2869). In Proceedings of the 24th ACM SIGKDD International conference on knowledge discovery & data mining.
- Dong, X. L., & Srivastava, D. (2013, April). *Big data integration* (pp. 1245–1248). In 2013 IEEE 29th international conference on data engineering (ICDE). IEEE.
- Ehrlinger, L., & WöB, W. (2016). Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48, 1–4.
- Elango, P. (2005). *Coreference resolution: A survey*. University of Wisconsin.
- Fedus, W., Zoph, B., & Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.
- Fotiadis, A., Polyzos, S., & Huan, T. C. T. (2020). The good, the bad and the ugly on COVID-19 tourism recovery. *Annals of Tourism Research*, 87, 103117.
- Gong, C., Tang, J., Zhou, S., Hao, Z., & Wang, J. (2019). Chinese named entity recognition with BERT. *DEStech transactions on computer science and engineering*, (ciscnc).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* (Vol. 1, No. 2). : MIT Press.
- Grishman, R., & Sundheim, B. M. (1996). *Message understanding conference-6: A brief history*. In COLING 1996 Volume 1: The 16th International conference on computational linguistics.
- Hernández-Méndez, J., & Muñoz-Leiva, F. (2015). What type of online advertising is most effective for eTourism 2.0? An eye tracking study based on the characteristics of tourists. *Computers in Human Behavior*, 50, 618–625.
- Jiang, M., & Shang, J. (2020, August). *Scientific text mining and knowledge graphs* (pp. 3537–3538). In Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining.
- Kärle, E., Simsek, U., Akbar, Z., Hepp, M., & Fensel, D. (2017). Extending the schema.org vocabulary for more expressive accommodation annotations. In *Information and communication Technologies in Tourism 2017* (pp. 31–41). Springer.
- Kejriwal, M. (2016). *Populating a linked data entity name system: A big data solution to unsupervised instance matching* (Vol. 27). IOS Press.



- Kejriwal, M. (2019). *Domain-specific knowledge graph construction*. Springer International Publishing.
- Kejriwal, M. (2021). A meta-engine for building domain-specific search engines. *Software Impacts*, 7, 100052.
- Kejriwal, M., Gilley, D., Szekely, P., & Crisman, J. (2018, April). *Thor: Text-enabled analytics for humanitarian operations* (pp. 147–150). In Companion proceedings of the the web conference 2018.
- Kejriwal, M., Knoblock, C. A., & Szekely, P. (2021). *Knowledge graphs: Fundamentals, techniques, and applications*. MIT Press.
- Kejriwal, M., & Szekely, P. (2017). Knowledge graphs for social good: an entity-centric search engine for the human trafficking domain. *IEEE Transactions on Big Data*.
- Kejriwal, M., & Szekely, P. (2019). myDIG: Personalized illicit domain-specific knowledge discovery with no programming. *Future Internet*, 11(3), 59.
- Ling, X., Singh, S., & Weld, D. S. (2015). Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3, 315–328.
- Liu, X., Gao, F., Zhang, Q., & Zhao, H. (2019). Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279*.
- Liu, X., Yang, P., & Fang, H. (2014, April). *Entexpo: An interactive search system for entity-bearing queries* (pp. 784–788). In European conference on information retrieval. Springer.
- Lockard, C., Shiralkar, P., & Dong, X. L. (2019, June). *OpenCeres: When open information extraction meets the semi-structured web* (pp. 3047–3056, Vol. 1, Long and short papers). In Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies.
- Lu, C., Laublet, P., & Stankovic, M. (2016, November). Travel attractions recommendation with knowledge graphs. In *European knowledge acquisition workshop* (pp. 416–431). Springer.
- Lü, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K., & Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1), 1–49.
- Maheshwari, G., Trivedi, P., Lukovnikov, D., Chakraborty, N., Fischer, A., & Lehmann, J. (2019, October). *Learning to rank query graphs for complex question answering over knowledge graphs* (pp. 487–504). In International semantic web conference. Springer.
- McKinsey & Company. (2020, October). *How COVID-19 has pushed companies over the technology tipping point—and transformed business forever*. URL: <https://www.mckinsey.com/business-functions/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever>
- Miles, A., & Bechhofer, S. (2009). SKOS simple knowledge organization system reference. *W3C recommendation*.
- Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1), 3–26.
- Neuburger, L., & Egger, R. (2020). Travel risk perception and travel behaviour during the COVID-19 pandemic 2020: A case study of the DACH region. *Current Issues in Tourism*, 24, 1–14.
- Novak, B. (2004). A survey of focused web crawling algorithms. *Proceedings of SIKDD*, 5558, 55–58.
- Panasiuk, O., Akbar, Z., Gerrier, T., & Fensel, D. (2018, March). Representing GeoData for Tourism with Schema.org. In *GISTAM* (pp. 239–246).
- Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3), 489–508.
- Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation (pp. 1532–1543). In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP).
- Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003, July). *Table extraction using conditional random fields* (pp. 235–242). In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval.

- Piskorski, J., & Yangarber, R. (2013). Information extraction: Past, present and future. In *Multi-source, multilingual information extraction and summarization* (pp. 23–49). Springer.
- Portisch, J., Fallatah, O., Neumaier, S., Jaradeh, M. Y., & Polleres, A. (2020, September). *Challenges of linking organizational information in open government data to knowledge graphs* (pp. 271–286). In International conference on knowledge engineering and knowledge management. Springer.
- Pujara, J., Miao, H., Getoor, L., & Cohen, W. (2013, October). *Knowledge graph identification* (pp. 542–557). In International semantic web conference. Springer.
- Ratinov, L., & Roth, D. (2012, July). *Learning-based multi-sieve co-reference resolution with knowledge* (pp. 1234–1244). In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning.
- Saputro, K. E., Kusumawardani, S. S., & Fauziati, S. (2016, October). Development of semi-supervised named entity recognition to discover new tourism places (pp. 124–128). In 2016 2nd International conference on science and technology-computer (ICST). IEEE.
- Sarawagi, S. (2008). *Information extraction*. Now Publishers.
- Seeger, M., & Ultra-Large-Sites, S. (2009). *Key-value stores: A practical overview*. Computer Science and Media.
- Shi, B., & Weninger, T. (2018, April). Open-world knowledge graph completion (Vol. 32, No. 1). In Proceedings of the AAAI conference on artificial intelligence.
- Singhal, A. (2012). Introducing the knowledge graph: Things, not strings. *Official Google Blog*, 5, 16.
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2), 51–53.
- Spyromitros, E., Tsoumakos, G., & Vlahavas, I. (2008, October). *An empirical study of lazy multilabel classification algorithms* (pp. 401–406). In Hellenic conference on artificial intelligence. Springer.
- Strassel, S., & Tracey, J. (2016, May). *Lorelei language packs: Data, tools, and resources for technology development in low resource languages* (pp. 3273–3280). In Proceedings of the tenth international conference on language resources and evaluation (LREC'16).
- Talvensaari, T., Pirkola, A., Järvelin, K., Juhola, M., & Laurikkala, J. (2008). Focused web crawling in the acquisition of comparable corpora. *Information Retrieval*, 11(5), 427–445.
- Tiddi, I., Lécué, F., & Hitzler, P. (Eds.). (2020). *Knowledge graphs for explainable artificial intelligence: Foundations, applications and challenges* (Vol. 47). IOS Press.
- Turian, J., Ratinov, L., & Bengio, Y. (2010, July). *Word representations: a simple and general method for semi-supervised learning* (pp. 384–394). In Proceedings of the 48th annual meeting of the association for computational linguistics.
- Vollmers, D., Jalota, R., Moussallem, D., Topiwala, H., Ngomo, A. C. N., & Usbeck, R. (2021). Knowledge graph question answering using graph-pattern isomorphism. *arXiv preprint arXiv:2103.06752*.
- Wang, P. (2019). On defining artificial intelligence. *Journal of Artificial General Intelligence*, 10(2), 1–37.
- Wang, Y., Lee, K., & Lee, I. (2014). Visual analytics of topological higher order information for emergency management based on tourism trajectory datasets. *Procedia Computer Science*, 29, 683–691.
- Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12), 2724–2743.
- Weibel, S., Kunze, J., Lagoze, C., & Wolf, M. (1998). Dublin core metadata for resource discovery. *Internet Engineering Task Force RFC*, 2413(222), 132.

- Yan, H., Deng, B., Li, X., & Qiu, X. (2019). Tener: Adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.
- Yeh, S. S. (2020). Tourism recovery strategy against COVID-19 pandemic. *Tourism Recreation Research*, 46, 1–7.
- Yochum, P., Chang, L., Gu, T., Zhu, M., & Zhang, W. (2018, October). *Tourist attraction recommendation based on knowledge graph* (pp. 80–85). In International conference on intelligent information processing. Springer.
- Zhang, C., Fan, C., Yao, W., Hu, X., & Mostafavi, A. (2019). Social media for intelligent public information and warning in disasters: An interdisciplinary review. *International Journal of Information Management*, 49, 190–207.

**Part IV**  
**Additional Methods**

# Network Analysis



## Connecting the Dots to Understand Tourism

Rodolfo Baggio

### Learning Objectives

- Understand the fundamental concepts and methods of network analysis
- Explain the most important issues and become familiar with the basic notation and terminology used for network analysis
- Develop elementary and practical network analysis skills and be able to visualize and compute main measures
- Show how to apply the methods of network science to unravel real-life problems and analyze real-world networks

## 1 Introduction and Theoretical Foundations

Most known and studied systems and phenomena can be classified as complex systems. Complex, in the popular language, indicates something one has difficulty fully understanding or describing; here, however, complex refers to a specific class of elements. They are characterized by having a certain number of parts, often organized with a detectable structure (Brodu, 2009; Levin, 2003; Lewin, 1999). These elements are interconnected, and the relationships that bind them are of a non-linear nature. The system exhibits a number of peculiar features, the most relevant of which include the following:

---

R. Baggio (✉)

Dondena Center for Research on Social Dynamics and Public Policy, Bocconi University, Milan, Italy

Tomsk Polytechnic University, Tomsk, Russia

e-mail: [rodolfo.baggio@unibocconi.it](mailto:rodolfo.baggio@unibocconi.it)

- emergence: structures and behaviors seem to appear at a global level that cannot be easily derived from single elements;
- robustness and fragility: sudden events might be easily absorbed by the system, but some seemingly insignificant shocks might disrupt it;
- self-organization: the system seems to generate structures or hierarchies autonomously and without any central guidance;
- evolutionary dynamics (adaptiveness): systems have a continuous exchange with the environment they are embedded in, and they adapt to these evolving conditions.

The net result is a fundamental unpredictability of the detailed structural and dynamic characteristics in the long term. Examples of complex adaptive systems include the patterns of birds in flight or the interactions of various life forms in an ecosystem, the behavior of consumers in a retail environment, people and groups in a community, the economy, the stock-market, the weather, earthquakes, traffic jams, the immune system, river networks, zebra stripes, sea-shell patterns, and many others. As such, tourism and tourism systems (e.g., destinations) are, unquestionably, typical complex systems. Its basic composition (elements and relationships) naturally leads to the idea of a useful representation being that of a network. A network (graph) is an abstract model in which the elements of the system are represented as dots connected by lines. A further abstraction consists of describing the network with a matrix (a.k.a. an adjacency matrix) whose elements indicate whether two nodes are connected or not. This allows for the use of powerful methods of linear algebra to calculate a wide array of measures that provide the characterizing features of the network (Barabási, 2016; Coscia, 2021; Sayama et al., 2016).

The basic idea of network science involves mapping and analyzing the patterns of relations among the elements of a system to understand its structure and, given a strong existing link, to examine its functions and the dynamic processes that may be involved. Moreover, mathematical modeling makes it possible to employ a wide array of techniques in order to simulate phenomena in cases where a real-life experiment would not be feasible due to theoretical, ethical, or practical reasons (Baggio & Baggio, 2020).

Lastly, the topological approach, which, regardless of the nature of the elements at play, takes structural features into account, allows the consideration of ensembles of objects (networks) belonging to very diverse domains and studies the possible existence of universal features that can better help to understand specific systems via analogical means. This is done based on a strong theoretical background, that of statistical physics, from which network science borrows many techniques and methodological approaches.

### 1.1 Network Analysis in a Nutshell

Formally speaking, a graph (network) is a pair  $G = (V, E)$ , where  $V$  is a set of vertices (nodes), and  $E$  is a set of pairs of distinct vertices, which are members of  $V$ :  $E = \{(u,v) \mid u, v \in V\}$ . The elements of  $E$  are called arcs, ties, links, or edges. Links can be assigned different properties, such as direction or weight (cost, intensity, duration, etc.). The basic types shown in Fig. 1 which also contain their matrix representation (adjacency matrices).

A special type of network is that of bipartite (a.k.a. 2-mode or affiliation network) in which the nodes can be divided into two disjoint and independent sets such that the edges only connect members of different sets. Examples may include authors-papers, affiliates-groups, topics-documents, and so on. For those interested, a complete survey of this network type can be found in the works of Pavlopoulos et al. (2018) and Guillaume and Latapy (2006).

Recent literature has provided a wealth of measures that can be used to characterize a network (Barabási, 2016; da Fontoura Costa et al., 2007). Among these, the most relevant and often used are the following:

- *density*: the portion of potential connections in a network that are actually present;
- *degree*: the number of links each node has; and *degree distribution*, the statistical distribution of the number (and sometimes the type) of the linkages among the network elements;
- *assortativity*: the correlation between the degrees of neighboring nodes;
- *average path length*: the mean distance (number of links) between any two nodes and *diameter*, the maximal shortest path connecting any two nodes;
- *closeness*: the mean weighted distance (i.e. the shortest path) between a node and all other nodes reachable from it;

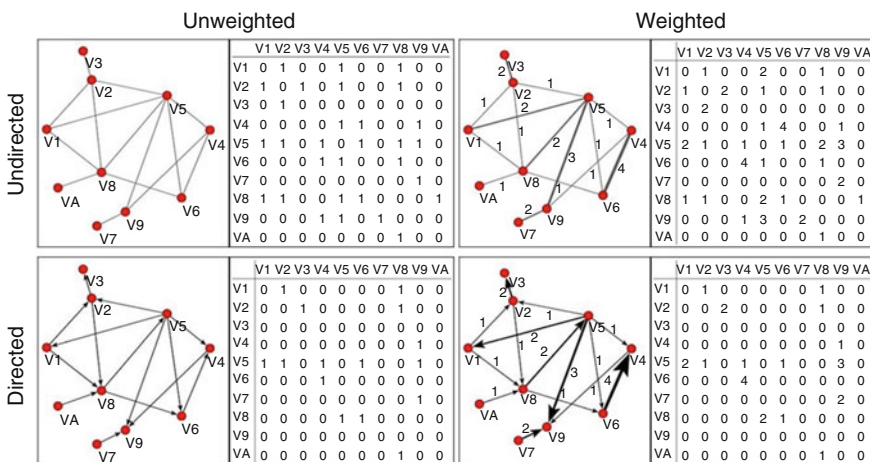
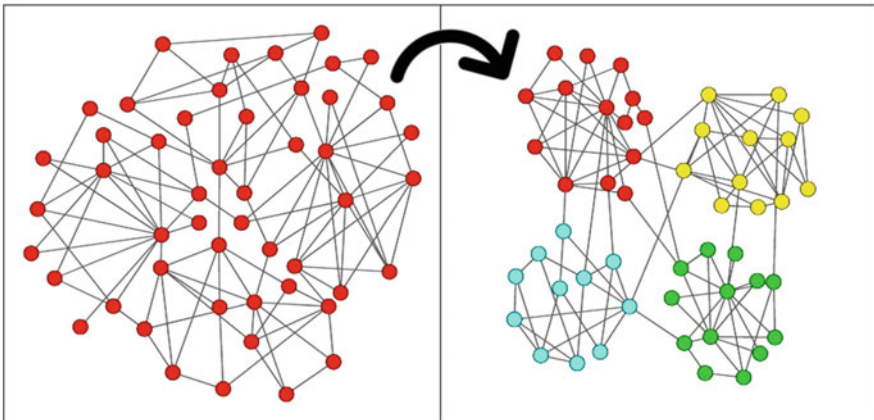


Fig. 1 Network types

- *betweenness*: the extent to which a node falls between others on the shortest paths connecting them;
- *clustering coefficient*: the concentration of connections of a node's neighbors—it provides a measure of the heterogeneity of the local density of links;
- *eigenvector*: calculated by using the matrix representation of the network and its principal eigenvector and based on the idea that a relationship to a more interconnected node contributes to its own centrality to a greater extent than a relationship with a less efficient interconnected node. One variation of this measure is the well-known *PageRank*.

A network study starts with collecting the necessary data (i.e., nodes and links). When tourism systems (e.g., destinations) are involved, the nodes typically represent the different stakeholders of the system (hotels, restaurants, service companies, travel agencies, public bodies, etc.). The links may be collected using different methods, for instance, surveys (explicitly asking participants for and about their connections), websites' hyperlinks between companies, listings from associations or consortia, official records on co-ownership, and so on. Frequency of connections or perceived importance can be used as weights for the edges. Once a network has been built, the study makes use of suitable software packages or libraries for some programming languages in order to derive the various measures that are typically used for assessing the system's features at three levels of analysis:

- *individual (microscopic) level*: refers to the specific nodal properties such as degree, betweenness, closeness, clustering coefficient, and so on. Normalized versions of these metrics are usually known as centralities (e.g., degree centrality, betweenness centrality, etc.);
- *intermediate (mesoscopic) level*: aims at highlighting the possible modular structure of a network. These modules (communities or clusters) are formed by nodes that are more densely connected between themselves than to the rest of the network (Fig. 2). The quality of the division into modules is measured by a

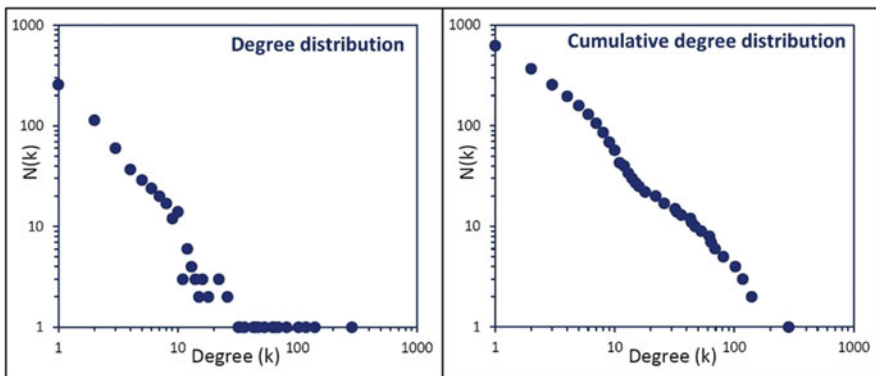


**Fig. 2** A network and its communities



modularity index. Several algorithms allow for the detection of these clusters (Fortunato, 2010); for instance, one of the most used and reliable algorithms is that proposed by Traag et al. (2019), also known as the *Leiden algorithm*. Built upon previous techniques, this iterative algorithm recursively assigns nodes to different groups until all elements are locally optimally assigned to a partition (i.e., the modularity index is maximized), providing communities that are guaranteed to be connected. Hierarchical structures can also be revealed by using similar algorithms;

- *global (macroscopic) level*: describes the overall structure operationalized by quantities such as density, average path length, diameter, etc. The most important and common measurement for describing the topology (structure) of a network is the probability distribution of the degrees,  $P(k)$  (degree distribution). Its mathematical form hints at the general features of the network, its complexity, and its behavior when subject to a dynamic process. For many real networks, typical degree distribution has a power-law shape ( $P(k) \sim k^{-\alpha}$ , see Fig. 3); that is, few nodes have many connections (i.e., hubs), while many others only have a few links. The degree distribution is also an indicator of possible mechanisms for the formation and evolution of the network (Coscia, 2021; Newman et al., 2011). Other measures used to describe macroscopic characteristics are the existing correlations between the distributions of different metrics as well as the average values of the microscopic metrics over the whole network.



**Fig. 3** A degree distribution with its cumulative version (axes are logarithmic to better show the long tailed shape of the distributions)

## 2 Practical Demonstration

Network analysis naturally leads the researcher to adopt a broad systemic view and to focus on general issues that concern the area or the problem under investigation. In many cases, the use of these methods has resulted in outcomes that can be considered counterintuitive or not sufficiently highlighted (see, e.g., Baggio, 2011). As for any other inquiry, it is important to start with a clear idea of the objectives of the work since these may affect the choices concerning the definition of the different elements, the techniques, and the metrics to be used. Normally, a good scan of the literature can provide valuable suggestions in this regard. From these preliminary explorations, a conceptual map can be drawn containing all the elements and steps needed (see, e.g., Baggio & Baggio, 2020).

Once a research question has been determined, the next step consists of defining the elements of the network: the nodes and links. Usually, this process is a relatively straightforward task for what concerns the nodes; they can be, for example, the stakeholders (firms, associations, public bodies, etc.) at a destination, employees in a company, papers published, or individuals. Together with the *names* of the entities identified as nodes, it is also recommended to collect some attributes that describe them (e.g., location, size, type of business, type of entity, etc.), which can then be used for later comparisons. Links are uncovered, as previously stated above, by using a survey in which respondents are asked to indicate their main connections, public records, listings for groups, and so on. Here, too, an evaluation of the relevance (e.g., importance, cost, speed, frequency of contacts, etc.) can then be used to weigh the links, if need be. These weights can then be rendered using a suitable scale, which also allows for the use of some *qualitative* features. All in all, it is essential to try to be as complete as possible. In fact, the distributions of practically all the network's characteristics (degree, closeness, betweenness, etc.) are strongly skewed, showing long tails, and, therefore, all the *usual* sampling considerations (designed for almost-normal distributions) do not apply. Verification of this completeness can be confirmed with good knowledge of the analyzed domain and by resorting to a few interviews with some knowledgeable informants.

Ideally, all the data points should be organized into a couple of tables, as shown in Fig. 4.

The use of a tool such as Excel allows for the organization of the data and can easily transform the data into the format requested by the software chosen for the analysis (often, as for Gephi, a csv file).

Once the network has been obtained, a suitable software (see “Available Software”) can calculate all the desired metrics. Software packages, such as Gephi, Unicet, Pajek, etc., provide functions for basic analyses and have little (or limited) support when special network features are involved (e.g., bipartite networks, link weights, directionality of links, etc.). More advanced methods, or full treatment of special cases, as well as dynamic simulations, need to be addressed using more efficient libraries such as those implemented in Python (NetworkX, igraph, etc.), R (igraph, sna, tnet, etc.), or MATLAB.

Nodes						Links					
ID	Name	Code	Type	Location	Size	Source	Target	SourceID	TargetID	Type	Weight
1	Acme Hotels	H001	Hotel	A	Medium	H001	P001	1	8	Undirected	1
2	Globex Hotel	H002	Hotel	A	Medium	H002	G002	2	7	Undirected	1
3	Soylent agency	A001	Travel agency	B	Small	H003	S001	4	10	Undirected	2
4	Blue Cat Hotel	H003	Hotel	B	Small	G001	H001	5	1	Undirected	1
5	Umbrella consortium	G001	Association	A	Medium	T001	H003	6	4	Undirected	3
6	Hooli Buses	T001	Transports	C	Small	G002	T001	7	6	Undirected	2
7	Veherent association	G002	Association	C	Medium	P001	T002	8	9	Undirected	1
8	Tourism board	P001	Public	A	Medium	T002	G001	9	5	Undirected	1
9	Relaxicab	T002	Transports	B	Large	H001	T002	1	9	Undirected	1
10	InGen	S001	Services	C	Small	H001	G002	1	7	Undirected	2
						A001	S001	3	10	Undirected	2
						P001	T001	8	6	Undirected	3
						P001	H003	8	4	Undirected	1
						G002	S001	7	10	Undirected	1

Fig. 4 Sample data collection table

### 3 A Worked Example

For this example, the software Gephi will be used, which is a good choice for an initial approach to network analysis. The reader is advised to follow the tutorials provided online in order to better understand and become familiar with the basic functioning thereof (<https://gephi.org/users/>) .<sup>1</sup>

Let us consider the network of a small tourism destination in which the nodes are the companies and associations in the area. They are assigned an attribute (Biz) that codes their main business activity (Association: ASS; Hotel: HOT; Other Accommodation: OTH; Restaurant: RES; Other services: SRV; Travel Agency: TVA). The links represent any type of collaboration or relation between two entities, and the network is symmetric and unweighted. The objective is thus to analyze this network and highlight the most important items and how they cluster together in order to assess the collaborative atmosphere at the destination.

The network is loaded and displayed in the main Gephi panel (*Overview*), and the main working environment (Fig. 5) provides the user with all the algorithms needed for analyzing the network and its visualization. In particular, the *Statistics* panel contains the functions needed to calculate all the metrics described above (Fig. 6).

Once done, all the results can be found in the *Data Laboratory* screen (Fig. 7). The table can be downloaded as a csv file, which can then be analyzed further with other tools, for example, Excel, if needed.

From this window, one can easily identify the most relevant elements by sorting the different metrics. Keep in mind that, depending on the extent to which *importance* is given, different measures may apply. Here, degree signals popularity, clustering coefficient refers to collaborative relationships with immediate neighbors, and betweenness indicates a broker or a bottleneck. If necessary, a global indicator can be derived from averaging all of these values.

<sup>1</sup>The Gephi file is available at the books Github profile: <https://github.com/DataScience-in-Tourism/DataScience-in-Tourism>

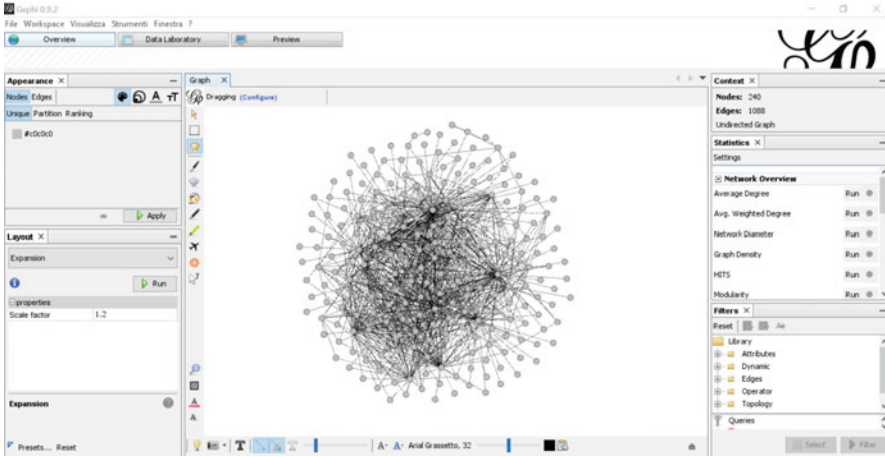
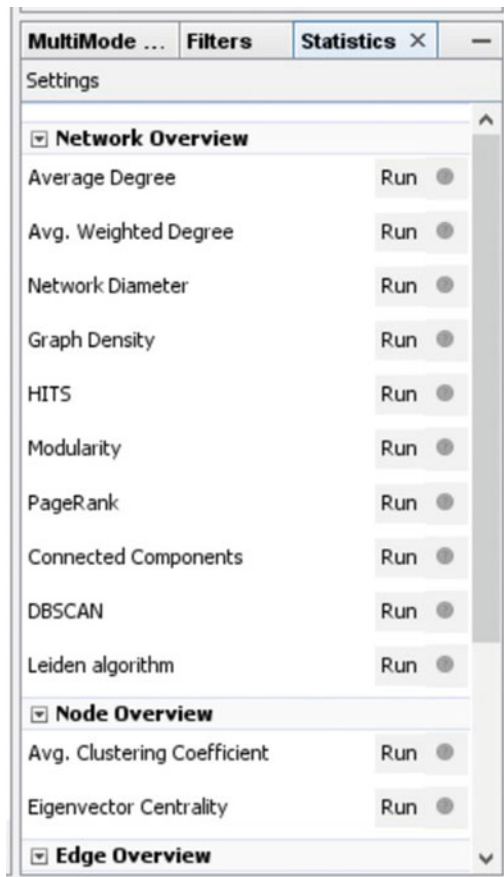


Fig. 5 Gephi main screen

Fig. 6 Gephi's Statistics functions



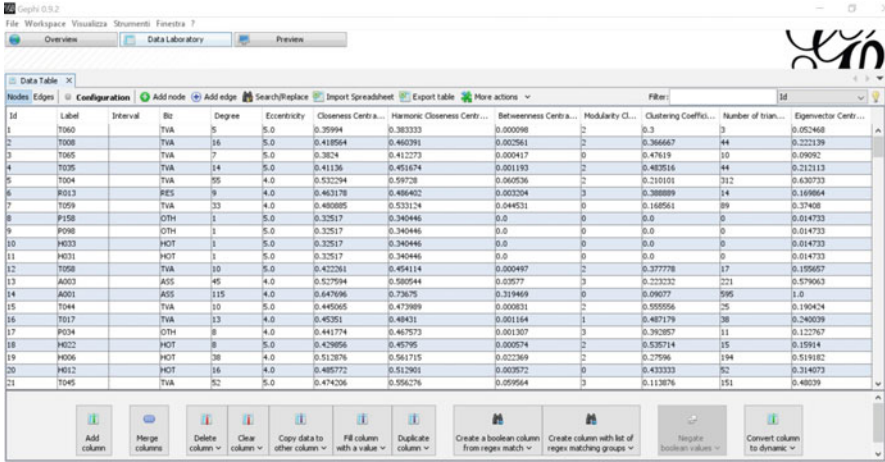


Fig. 7 Gephi’s Data Laboratory screen with an example of the metrics computed by the program

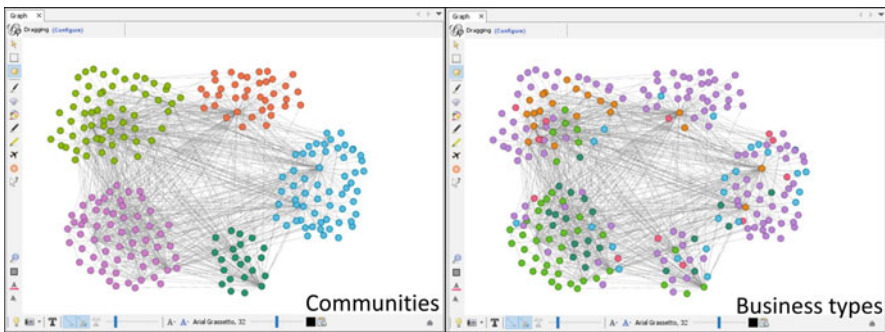


Fig. 8 A network laid out and colored by the communities uncovered and by the type of business inside the communities

The analysis reveals a network with low connectivity (density = 0.038) yet good compactness for the low value of the average path length and diameter (diameter = 6, average path length = 2.539). The good average clustering coefficient (0.5) shows good capabilities and willingness from the actors’ side to work with their immediate neighbors. Moreover, a modularity analysis identifies four communities (Fig. 8); but relatively badly defined (Q = 0.299, a low value). If, then, we consider the composition of these communities, for example, by coloring them by business type, we see that they are all “mixed,” that is, no cluster is formed by one single type of operator. It must be noted here that, unfortunately, Gephi does not provide functions for this type of node rearrangement. Therefore, the repositioning must be performed by hand, which is clearly only feasible in cases where the number of nodes is not too high. Thus, by highlighting the different communities (relatively poorly separated), a

relatively good tendency to collaborate could be emphasized. Yet, considering that they are formed by different types of tourism operators, a good possibility of imagining and designing multifaceted products and services exists. Overall, all of these considerations should be validated and verified by taking deeper knowledge of the specific destination into account.

Finally, the last Gephi panel (*Preview*) shows a picture of a network that can be personalized and exported into a variety of different formats.

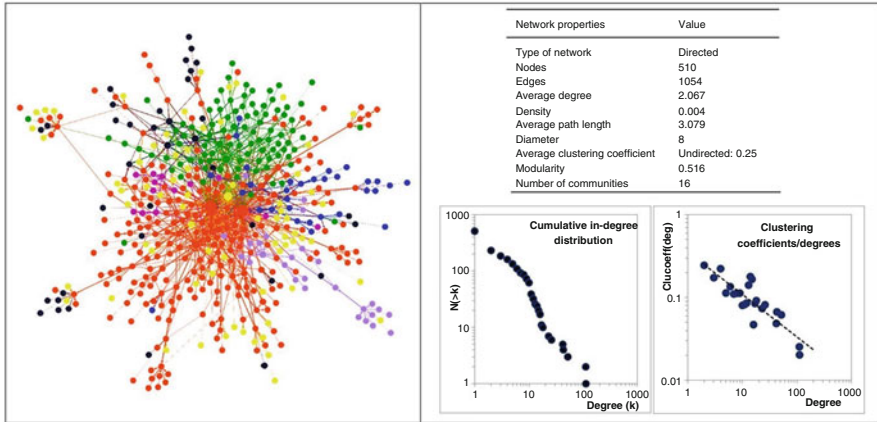
It is important to take into consideration that regardless of which package is selected, they all have their limitations in the fact that the metrics and the type of analyzable network (symmetric, weighted, directed, etc.) are given. Moreover, it is not always easy to understand what parameters are used for the calculations (algorithm, normalization factors, features considered, links' weights, etc.). When peculiar computations are needed, the only possibility is to use a programming language with the available libraries. The same can be said if some kind of simulation is desired or if modifications are to be implemented on the network as well as their effects assessed.

Lastly, it is worth underlining that this is a domain in which the *old* distinction between qualitative and quantitative approaches is not only meaningless but can also be viewed as dangerous. As exemplified and well-defined in Mariani and Baggio (2020), purely qualitative methods risk leading to contradictory or inaccurate results. On the other hand, good qualitative knowledge of the issues under study is of crucial importance for a correct and useful interpretation of the quantitative results. For this reason, and given the different competencies and expertise required, a meaningful study typically requires a well-mixed multidisciplinary team of researchers.

## 4 Research-Case

The case described in this section is the one discussed in Raisi et al.'s (2019) paper, "A network perspective of knowledge transfer in tourism." The objective is to investigate the characteristics of the inter-organizational knowledge transfer in Western Australian tourism by assessing the topological characteristics of the existing network at the destination. The relevance of this issue is evident since efficient, effective, and smooth flows are recognized to be of crucial importance for creating a sustained competitive advantage in both destinations and individual organizations and are a prerequisite for establishing the innovative atmosphere fundamental for maintaining a good level of competitiveness (Cooper, 2018; Hjalager, 2010).

The network was built by collecting data from companies, businesses, and organizations involved in the tourism industry in the area through an online questionnaire. Essentially, the question asked individuals to name ten entities, in order of importance, from which the participants receive information or knowledge and the (perceived) importance of these transfers for their own business. The resulting network is directed (direction being the citation of an existing relationship), and



**Fig. 9** The Western Australian tourism network and its main characteristics (adapted from Raisi et al., 2019)

the analysis used a combination of tools including UCINET (Borgatti et al., 1992), Gephi (Bastian et al., 2009), and the Networkx Python library (Hagberg et al., 2008). These resulted in the metrics summarized in Fig. 9.

The network is rather sparse (very low density) but relatively compact (small diameter and low average path length). Furthermore, the degree distribution has a clear power-law shape, meaning that a large number of organizations receive information from a few but highly central organizations. The clustering coefficient calculated on a symmetrized version of the network is relatively high, and the modularity index is similarly high as well. This indicates that a few (16) well-defined communities exist and that the actors tend to form small, closely related collaborative groups. Drawing the relationship between the average clustering coefficient of a node and its degree, a power-law relationship is obtained that suggests a hierarchical organization of the whole system (see Ravasz & Barabási, 2003).

From a microscopic point of view, the authors identify the most relevant actors in the network. Given the different meanings of “importance” attributed to the various nodal measures, a workable suggestion is to use, as a global indicator, the geometric mean of the normalized values of the four main centrality measures: in-degree, closeness, betweenness, and eigenvector (Sainaghi & Baggio, 2014). The completed ranking is shown in Fig. 10, which clearly renders regional public institutions and associations as important for disseminating information and knowledge.

Thus, in this particular case, the use of network analytic methods was able to provide a good picture of the situation and supply a series of outcomes that, most likely, would have remained blurred when using other methods. More importantly, however, this investigation has opened up an avenue of deeper and more interesting analyses. One possible development would be to refine the analysis by estimating, for each actor, the capabilities to absorb and transfer information and see how this

Rank	Importance index	Region	Sector
1	0.438	Experience Perth	Public tourism body
2	0.411	Experience Perth	Public tourism body
3	0.247	Experience Perth	Regional tourism organization
4	0.217	Australia's South West	Regional tourism organization
5	0.174	National	Public tourism body
6	0.141	Experience Perth	Tourism association
7	0.125	Australia's South West	Tourism association
8	0.121	Experience Perth	Tourism association
9	0.110	National	Public tourism body
10	0.107	Experience Perth	Tourism association
11	0.106	Experience Perth	Regional tourism organization
12	0.103	Experience Perth	Public tourism body
13	0.097	Experience Perth	Information services
14	0.092	Experience Perth	Tourism association
15	0.092	Experience Perth	Information services

**Fig. 10** The most relevant actors in the Western Australian tourism network (adapted from Raisi et al., 2019)

modifies the overall picture. Moreover, by implementing some suitable numerical simulation, it would be possible to examine the effects different modifications of the network's structure can have on the dynamic process of exchanging information as well as how to optimize the process.

### Service Section

**Main Application Fields:** Practically any domain in which a “relational” aspect is considered important and in which entities with the role of nodes and relationships between any two of them can be reasonably and meaningfully defined.

**Limitations and Pitfalls:** The initial data collection is a delicate matter (see above) as it is the use of specific software tools that might limit the cases in which they can be used and force the researcher (usually for reasons regarding lack of awareness or know-how) to resort to unnecessary modifications of the network (e.g., symmetrizing, projecting bipartite networks, dichotomizing links' weights, etc.), ultimately reducing the informative content of the data, without, at least, exploring the effects of various possible changes.

**Similar Methods and Methods to Combine with:** Any other method that is useful to fully answer the research questions. Often combined with some regression or correlation analysis.

**Code:** The Gephi-File is available at: <https://github.com/DataScience-in-Tourism/Chapter-21-Social-Network-Analysis>



## Further Readings and Other Sources

- Baggio, R. (2013). *Complexity, Network Science & Tourism*. IFITT Education Group Available at [http://www.iby.it/turismo/papers/rb\\_TourNetSci\(IFITT\).pdf](http://www.iby.it/turismo/papers/rb_TourNetSci(IFITT).pdf)
- Barabási, A. L. (2016). *Network science*. Cambridge University Press.
- Caldarelli, G., & Chessa, A. (2016). *Data science and complex networks: Real case studies with python*. Oxford University Press.
- Coscia, M. (2021). *The atlas for the aspiring network scientist*. IT University of Copenhagen.
- Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press.
- Mitchell, M. (2009). *Complexity: A guided tour*. Oxford University Press.

## References

- Baggio, R. (2011). Collaboration and cooperation in a tourism destination: A network science approach. *Current Issues in Tourism*, 14(2), 183–189.
- Baggio, J. A., & Baggio, R. (2020). *Modelling and simulations for tourism and hospitality*. Channel View.
- Barabási, A. L. (2016). *Network science*. Cambridge University Press.
- Bastian, M., Heymann, S., & Jacomy, M. (2009). *Gephi: an open source software for exploring and manipulating networks*. Paper presented at the 3rd International AAAI conference on weblogs and social media, San Jose, CA (May 17–20).
- Borgatti, S. P., Everett, M. G., & Freeman, L. C. (1992). *UCINET (Version 6.2 – 2009)*. Analytic Technologies. [www.analytictech.com](http://www.analytictech.com)
- Brodu, N. (2009). A synthesis and a practical approach to complex systems. *Complexity*, 15(1), 36–60.
- Cooper, C. (2018). Managing tourism knowledge: A review. *Tourism Review*, 73(4), 507–520.
- Coscia, M. (2021). *The atlas for the aspiring network scientist*. IT University of Copenhagen.
- da Fontoura Costa, L., Rodrigues, A., Travieso, G., & Villas Boas, P. R. (2007). Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1), 167–242.
- Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3–5), 75–174.
- Guillaume, J. L., & Latapy, M. (2006). Bipartite graphs as models of complex networks. *Physica A*, 371(2), 795–813.
- Hagberg, A. A., Swart, P. J., & Schult, D. A. (2008). *Exploring network structure, dynamics, and function using NetworkX*. Paper presented at the 7th Python in science conference (SciPy2008), Pasadena, CA (19–24 Aug).
- Hjalager, A. M. (2010). A review of innovation research in tourism. *Tourism Management*, 31(1), 1–12.
- Levin, S. A. (2003). Complex adaptive systems: Exploring the known, the unknown and the unknowable. *Bulletin of the American Mathematical Society*, 40(1), 3–19.
- Lewin, R. (1999). *Complexity, life on the edge of chaos* (2nd ed.). The University of Chicago Press.
- Mariani, M., & Baggio, R. (2020). The relevance of mixed methods for network analysis in tourism and hospitality research. *International Journal of Contemporary Hospitality Management*, 32(4), 1643–1673.
- Newman, M., Barabási, A. L., & Watts, D. J. (Eds.). (2011). *The structure and dynamics of networks*. Princeton University Press.

- Pavlopoulos, G. A., Kontou, P. I., Pavlopoulou, A., Bouyioukos, C., Markou, E., & Bagos, P. G. (2018). Bipartite graphs in systems biology and medicine: A survey of methods and applications. *GigaScience*, 7(4), art. giy014.
- Raisi, H., Baggio, R., Barratt-Pugh, L., & Willson, G. (2019). A network perspective of knowledge transfer in tourism. *Annals of Tourism Research*, 80, art. 102817.
- Ravasz, E., & Barabási, A.-L. (2003). Hierarchical organization in complex networks. *Physical Review E*, 67, 026112.
- Sainaghi, R., & Baggio, R. (2014). Structural social capital and hotel performance: Is there a link? *International Journal of Hospitality Management*, 37, 99–110.
- Sayama, H., Cramer, C., Porter, M. A., Sheetz, L., & Uzzo, S. (2016). What are essential concepts about networks? *Journal of Complex Networks*, 4, 457–474.
- Traag, V. A., Waltman, L., & Van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, 9(1), 1–12.

# Time Series Analysis



## Forecasting Tourism Demand with Time Series Analysis

Irem Onder and Wenqi Wei

### Learning Objectives

- Demonstrates how to conduct time series analysis for tourism demand
- Explains stationary data and unit root tests
- Illustrates univariate and combined forecasting methods
- Evaluates forecasting accuracy of different models

## 1 Introduction and Theoretical Foundations

The time series method involves a sequence of observations in chronological order in which the intervals between time periods are equally spaced and the sequence is successive (Agung, 2011; Frechtling, 2012; Montgomery et al., 2015). Unlike cross-sectional data, in which the sequence of the observations is considered unnatural, time series data feature a naturally chronological order. They are therefore critical for forecasting problems in various disciplines, including operations management, marketing, finance and risk management, economics, industrial process control, and demography (Agung, 2011; Frechtling, 2012; Montgomery et al., 2015). Time series analysis refers to a method of analyzing time series data for the purpose of identifying the behaviors, statistics, and other meaningful features of said data. Researchers usually aim to establish a time series forecasting model that clarifies the data pattern, describes the statistical relationships between the historical and

---

I. Onder (✉) · W. Wei

Department of Hospitality and Tourism Management, University of Massachusetts, Amherst, MA, USA

e-mail: [ionder@isenberg.umass.edu](mailto:ionder@isenberg.umass.edu)

ongoing data, and predicts the data pattern in the future (Agung, 2011; Frechtling, 2012; Montgomery et al., 2015).

Tourism demand forecasting is a dominant research theme in the tourism field, and tourist arrival remains the most popular measurement of tourism demand (Önder, 2017; Önder & Gunter, 2016; Song & Li, 2008). Tourist arrivals (i.e., the number of tourists arriving at a particular destination) may be examined in terms of travel purposes (Kulendran & Wong, 2005), types of transportation (Coshall, 2005), and/or country of origin (Lim & McAleer, 2002). Tourist expenditures in a particular destination (Li et al., 2006) as well as specific tourism products (Au & Law, 2002) and length of stay (Alegre & Pou, 2006) are also frequently used to measure tourism demand.

The nascent trend of time series studies has already merged with artificial intelligence (AI) techniques. For instance, Palmer et al. (2006), who developed the artificial neural networks (ANN) methodology for forecasting the time series data of tourism expenditure, confirmed that ANN generates better results than traditional time series models, including ARIMA models (Cho, 2003) and single exponential smoothing models (Burger et al., 2001). On the other hand, VAR models (vector autoregressive models), in which each variable is a linear function of past lags of itself and past lags of the other variables, are popular in multivariate forecasting. Song and Witt (2006) adopted VAR to forecast tourist flows to Macau and justified that VAR can incorporate the response of economic shocks variables in the forecasting (Pesaran & Shin, 1995; Rivera, 2016). This does not mean, however, that advanced methodologies always perform better than traditional models. Yu and Schwartz (2006) found that the accuracy of AI models is lower than the accuracy of simple time series models for forecasting yearly tourist arrivals in the USA, thus confirming the findings of Makridakis and Hibon (2000). This indicates that adopting AI techniques requires thorough consideration; only because a particular method outperformed another in one certain situation, does not necessarily mean it will do so again in another.

Big data is regarded as an influential and valid source of prediction, which can be incorporated into a given forecasting model to enhance accuracy (Pan & Yang, 2017). Thus, time series forecasting not only depends on traditional secondary data but also incorporates innovative data sources. In tourism and hospitality research, Xiang and Pan (2011) state that search volume data can be regarded as a direct predictor of the volume of the destination's travel industry. Some examples of research that employ search data include Choi and Varian (2012), Yang et al. (2015), and Gunter and Önder (2016). Meanwhile, several other data sources have received growing attention for forecasting tourism demand. For example, Önder et al. (2020) innovatively validated that Facebook likes can be regarded as a leading predictor for tourism demand in four destinations in Austria, while Li et al. (2020) demonstrated that a forecasting model incorporating big data from multiple sources, including search engines and online reviews, performed significantly better than a single-source model. These studies confirmed that big data is a powerful external predictor and is critical for increasing model accuracy. Although there are many approaches to forecasting and time series analysis, each model has its advantages and disadvantages. In addition, it is important to note that one forecasting method is

not suitable for all types of tourism destination data with different time horizons (Witt & Song, 2002).

Overall, tourism demand forecasting is difficult because tourism is a perishable product and, as such, is inseparable from the production and consumption process. Moreover, customer satisfaction depends on complementary services (e.g., public transport at the destination), demand is sensitive to natural and manmade disasters (e.g., terrorism, pandemic, etc.), and tourism supply requires a long lead-time investment in planning and infrastructure (Frechtling, 2012). These reasons, and many more, make having an accurate forecast essential for tourism businesses and destinations.

## 2 Practical Demonstration

This section discusses the steps and requirements needed to run a time series analysis. In order to run a time series analysis, the following conditions must be met: (1) past information (historical data) is available, (2) this information is numerical or can be quantified, and (3) some aspects of the past pattern will continue in the future (assumption of continuity) (Frechtling, 2012). Time series analysis does not attempt to discover factors causing the behavior as future predictions are based on the valuables' past values. The inclusion of any variable in time series analysis turns it into explanatory (causal) forecasting. If one wants to conduct explanatory forecasting, then one also needs to conduct a Granger causality test, which is used for testing whether one time series is useful for predicting another one (Granger, 1969).

Different approaches to time series analysis have different assumptions. One critical type of time series includes stationary time series. A strictly stationary time series indicates that the observations, along with mean and variance, are not influenced by a change in time; more specifically, the joint probability distribution of stationary processes is not affected by a time shift (Montgomery et al., 2015). The first step involving any type of time series analysis is to look at a time series graph so as to see the development of data over time. Tourism data are generally seasonal, meaning they increase and decrease based on seasonal effects and are, therefore, not stationary.

The presence of a unit root is a common cause of violating the stationary process. As such, a unit root signifies 1 as the root of the characteristic equation of a non-stationary process and is a characteristic of stochastic processes. Thus, it needs to be carefully tested in order to avoid a systematically unpredictable pattern across the time series data (Glen, 2016). Common unit root tests include the Dickey–Fuller, Phillips–Perron, KPSS, ADF–GLS, Breusch–Godfrey, Ljung–Box, and Durbin–Watson tests (Enders, 2008; Maddala & Kim, 1998).

Four representative univariate forecasting methods that fit our data characteristics were selected to forecast the time series. These methods have been proven to be successful in tourism demand forecasting studies (Li et al., 2005).

## 2.1 Research Case<sup>1</sup>

Berlin, the capital of Germany, was chosen for this research case. It is one of the most popular destinations, ranking third, European-wide, for the total number of bednights in 2018 (ECM, 2019). The time series data include the monthly number of arrivals from Germany, Italy, Spain, and the United Kingdom as well as the total market in Berlin at paid accommodations. “Total market” refers to the combination of tourists from both the German and all international markets. Italy, Spain, and the UK were chosen because they are among the top ten foreign markets in terms of arrivals and bednights. The arrivals dataset spans from the year 2005 to the end of 2019 and was retrieved from TourMIS ([www.tourmis.info](http://www.tourmis.info)), an online system that includes tourism statistics for European destinations (Fig. 1).

Table 1 shows the variables and definitions used for the UK market. The same procedure was applied to the other time series in this demonstration.

The following analysis aims to estimate the models by using the 2005–2014 data, measuring the forecast accuracy of the model by applying out-of-sample forecasts from 2015 to 2019. The results will reveal the best forecasting option among the models applied in the study.

In order to achieve stationarity, Augmented Dickey–Fuller (ADF) unit root tests were conducted for the arrivals data. All of them revealed the presence of unit roots,

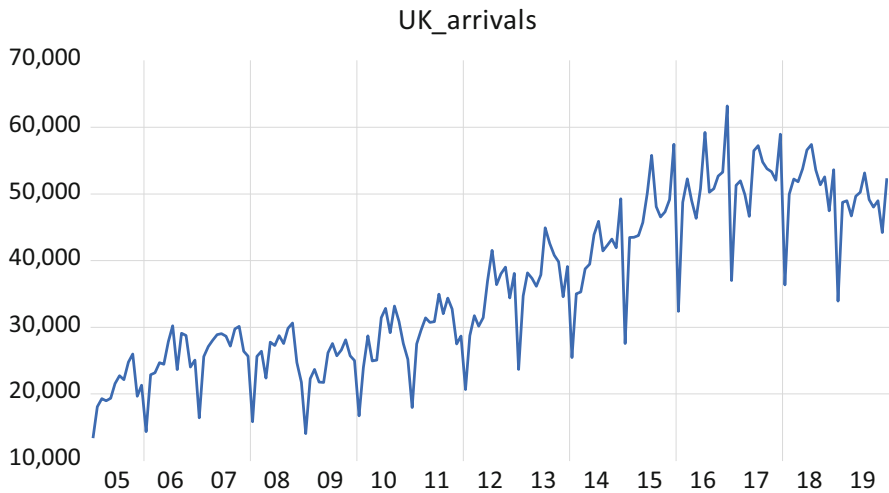
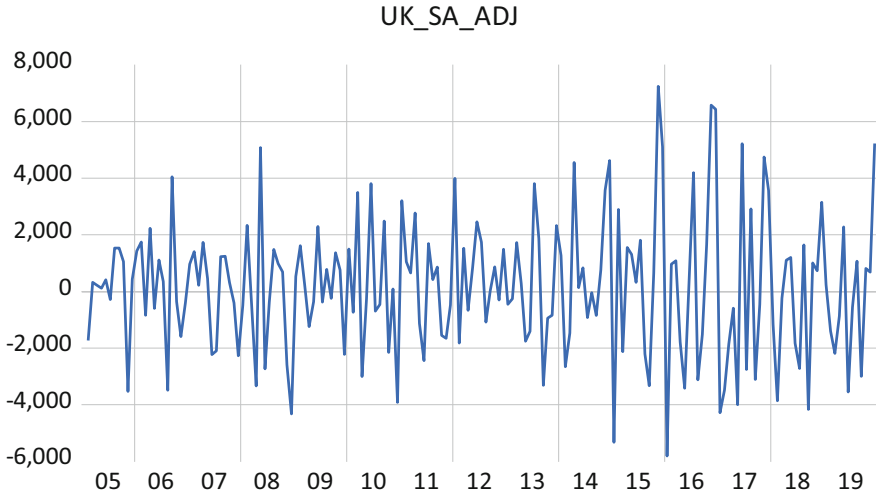


Fig. 1 UK arrivals to Berlin

<sup>1</sup>The full Eviews workflow example and data used in this chapter can be found here: [https://drive.google.com/drive/folders/1CMgNofpERihFqQ\\_GOmo3f3j1oT-pFL2G?usp=sharing](https://drive.google.com/drive/folders/1CMgNofpERihFqQ_GOmo3f3j1oT-pFL2G?usp=sharing)

**Table 1** Variables and definitions

Variable names	Definition
<i>uk</i>	Raw time series for arrivals from United Kingdom
<i>uk_sa</i>	Seasonality adjusted <i>uk</i> time series
<i>uk_sa_adj</i>	First differenced seasonality adjusted <i>uk</i> time series
<i>uk_sa_adj_sm</i>	First differenced seasonality adjusted and smoothed (SES) <i>uk</i> time series
<i>uk_ets</i>	ETS applied <i>uk</i> time series



**Fig. 2** First differenced seasonality adjusted time series

meaning the time series was not stationary. Thereafter, to reach stationarity, we applied seasonal differencing with a 12-month moving average to the variables and rechecked the unit root. This method was used because the Berlin arrivals data showed seasonality patterns. Results indicated that after taking the seasonal difference, a unit root still remained. Therefore, this process was followed by taking the first difference of the time series, which resulted in a stationary time series. If a time series is not stationary because it involves a trend (in an upward or downward direction), a seasonal effect (strong sales of heating oil in winter months), or both a trend and a seasonal effect, then a simple average fails to capture the data pattern. In such a case, one can conduct different forecasting methods such as the Holt–Winters triple exponential smoothing method or any other method that considers these assumptions (Fig. 2).

The forecast models have been assessed based on comparing their ex-post out-of-sample forecast accuracy in terms of the mean absolute percentage error (MAPE) and the root mean square error (RMSE). In addition, we calculated and assessed the simple average of combined forecasts based on the forecasts produced by the different models in the study. To solve the forecasting problem, we used the following time series approaches: seasonal naïve (no change), single exponential

smoothing (SES), error trend seasonal (ETS), and combined forecasts (simple average).

## 2.2 Forecasting Methods

### 2.2.1 Seasonal Naïve

The seasonal naïve method is often used with highly seasonal data in which the forecast value is set to be equal to the last observed value of the corresponding season of the previous year. As our dataset was seasonal, we used the actual time series (i.e., no seasonal adjustment or differencing) for this method. For instance, with the monthly data, the forecast value of February 2021 was equal to the actual value of February 2020. Similar rules can be applied not only to monthly data but also to quarterly and daily data (only when data is seasonal). Seasonal naïve is calculated as follows:

$$f_t = a_{t-12}$$

where  $a_t$  is the actual arrivals value and  $f_t$  is the forecast value.

### 2.2.2 Single Exponential Smoothing (SES)

Single exponential smoothing (SES) is very common when using forecasting data without noticeable trends or seasonal features. Exponential smoothing means that the weight of observations is exponentially decreasing over the time series. Thus, the more recent observation has higher weight and the older observation has lower weight. It is calculated as follows:

$$f_{t+1} = f_t + \alpha(a_t - f_t)$$

where  $a_t$  is the actual arrivals value,  $f_t$  is the forecast value, and  $\alpha$  is the smoothing constant between 0 and 1.

### 2.2.3 Error Trend Seasonal (ETS)

Error trend seasonal (ETS), which is regarded as an extended type of exponential smoothing (ES), can forecast time series data, based on state-space likelihood calculations, along with model selection and calculation of forecast standard errors. This model uses three parameters, Error, Trend, and Seasonal, in which each parameter can have different values, A/Ad (Additive or Additive Damped), M/Md (Multiplicative or Multiplicative Damped), or N (none). For instance, a model designated as “ETS(A,M,N)” contains an additive error, a multiplicative trend, and

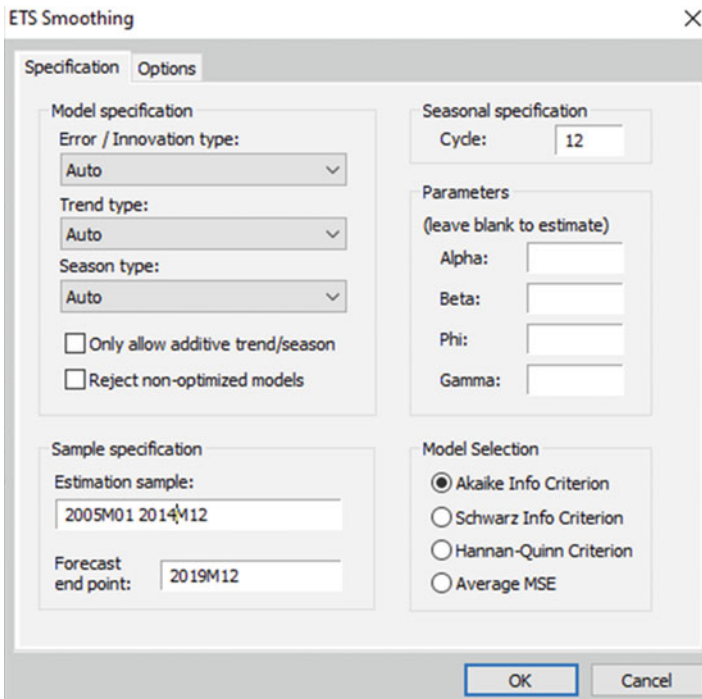


no seasonality. The best ETS model can be chosen based on Akaike’s Information Criterion (AIC), Akaike’s Information Criterion correction (AICc), and the Bayesian Information Criterion (BIC). AIC estimates the quality of each possible ETS model relative to each of the other models. AICc, on the other hand, is used when the sample size is small as AIC tends to select models with too many parameters, which may lead to overfitting of the model. BIC is similar to AIC, yet the penalty for parameters in BIC is  $\ln(n)k$  and in AIC is  $2k$ .

In EViews time series analysis software, these ETS models can be chosen automatically. Therefore, since this analysis also includes seasonal and trend terms, we used the actual (raw) data for the analysis instead of the smoothed and differenced time series. The following Table 2 shows the ETS models for each market (Fig. 3).

**Table 2** ETS models for each market

Market	ETS model
Germany	(M,M,M)
Italy	(A,Ad,M)
Spain	(A,N,A)
UK	(M,M,M)
USA	(A,N,A)
Total	(M,A,M)



**Fig. 3** ETS automatic selection in Eviews

### 2.2.4 Forecasting Combination Method

The forecasting combination method was introduced by Bates and Granger (1969) to improve forecasting accuracy. There are various ways of combining forecasts, including the simple average, weighted average combination, principal component, and Bayesian-based combination methods. Makridakis et al. (1982) compared 24 methods/models in 1001 series and concluded that the simple average composite is the most accurate method. Likewise, Makridakis and Winkler (1983) indicate that the simple average combination method is better than weighted combination methods because simple forecast combination results may be more robust. Their study has been widely cited, and, thus, in the following section, the average combined forecasting value of four methods was calculated for accuracy. The combination forecasts are based on the following formula:

$$f_c = \sum_{i=1}^n \frac{1}{n} f_i$$

where  $f_i$  is the  $i$ th single forecast,  $f_c$  is the combined forecast generated by the  $n$  single forecasts  $f_i$ , and  $\frac{1}{n}$  is the combined weight assigned to  $f_i$ .

### 2.3 Measures of Forecasting Accuracy

There are different accuracy measures for forecasting analysis; mean squared error (MSE), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are the most commonly used ones. In our study, for all four models, the out-of-sample forecasting performance of the monthly number of arrivals was assessed in terms of root mean square error (RMSE) and mean absolute percentage error (MAPE) as error measures. RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (a_t - f_t)^2}$$

where  $a_t$  is the value of the actual arrivals value,  $f_t$  is the forecast value, and  $n$  is the length of the forecast horizon. MAPE is calculated as:

$$MAPE = \frac{\sum_{t=1}^n \frac{|e_t|}{f_t}}{n}$$

where  $e_t$  is the forecast error,  $f_t$  is the forecast value, and  $n$  is the length of the forecast horizon.

### 3 Results

When deciding which forecasting method to use, more than one should be applied to the data in order to gauge the differences between them, as we also did in this example. The interpretation of the results is then based on the error terms. In our case, the smaller the error term, the closer we were to the actual values. The results of all the forecasting models and markets can be seen in Table 3. Overall, combined forecasts performed best with the lowest MAPE for each market. In terms of RMSE, combined forecasts performed the best across all markets except Germany (seasonal naïve) and the United Kingdom (SES). These results confirm that in terms of forecasting Berlin arrivals, combined forecasts were better than the individual models. These results are in line with Makridakis and Winkler (1983) insofar as simple average combined forecasts were shown to outperform other forecasting models.

The most accurate forecast was for German arrivals to Berlin. In general, domestic markets show more continuity and habitual travel patterns; therefore, in our case, it had the smallest forecast errors. Contrarily, Italy, Spain, and the United Kingdom had higher errors than domestic and total arrivals to Berlin.

As an example, the visual representation of model comparison for the UK market is shown in Fig. 4. Due to space limitations, we did not include all the graphs for the other markets.

#### Discussion and Limitations

Forecasting is not an exact science, but it can be a powerful tool for businesses and destinations as it may be necessary for different situations, such as scheduling aircraft staff, stocking inventory at a restaurant, or building a new airport. These types of decisions require an idea of the future, which can be accomplished via forecasting. Especially in tourism and hospitality, there often seems to be a time lag between awareness of a need and implementation of a solution (e.g., building a new airport). This lead time is the main reason for forecasting; hence its use for scheduling, acquiring resources, and determining resource requirements (Frechtling, 2012). Therefore, it is essential for tourism and hospitality businesses and destinations to conduct accurate forecasting analyses.

This study indicates that combined forecasts yield better results than the individual models that were applied. In this example, we used univariate time series analysis, which takes only past data into account. Nonetheless, the models can be enhanced even further through the addition of other variables (e.g., GDP, inflation rate, unemployment rate, prices, etc.) that affect travel behavior. The more variables one adds to the model, the more complex the model gets. This does not, however, automatically signify better forecasting results. Simple univariate models, especially in tourism demand forecasting, can indeed perform better than more complex

**Table 3** Out-of-sample forecasting performance

	Seasonal naïve (no change)		Error trend seasonal (ETS)		Single exponential smoothing (SES)		Combined forecast	
	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE
Total	3.607	46060.09	5.194	77716.013	5.500	67842.108	<b>3.528</b>	<b>44772.220</b>
Germany	3.893	<b>30283.119</b>	10.047	82929.883	5.037	39480.540	<b>4.177</b>	34091.003
Italy	11.118	3738.436	10.695	3818.9962	12.061	3897.270	<b>9.636</b>	<b>3091.162</b>
Spain	14.615	4595.656	27.196	8418.989	16.500	5436.320	<b>9.961</b>	<b>3293.684</b>
UK	7.688	4501.337	17.628	11683.875	7.300	<b>4447.080</b>	<b>8.535</b>	5462.356

\*Bold figures indicate the smallest errors

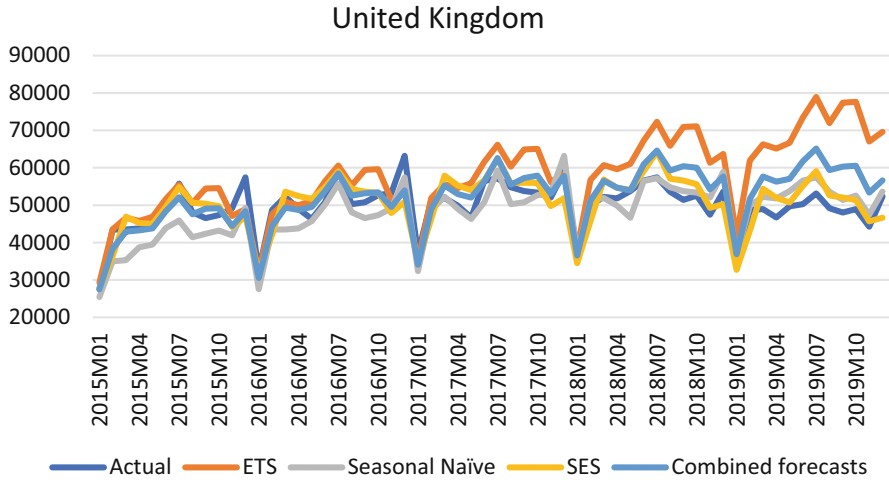


Fig. 4 Comparison of four models for the United Kingdom

models. The results may differ based on other destinations, markets, and time periods. Therefore, we cannot generalize our results.

In this study, EViews, a statistical software designed for time series forecasting and modeling, was used to conduct time series analysis. Nevertheless, there are numerous other software packages for time series analysis as well. For example, R and Python both feature free productive libraries and packages. R has the “ts,” “forecast,” and “smooth” packages, while Python has “fbprophet,” “pmdarima,” and “tsfresh,” all of which can be implemented for the application of forecasting. Additionally, SAS is an advanced statistical analytics software designed to accomplish time series analysis effectively and provide excellent visualization of time series analysis results (Brocklebank & Dickey, 2003), and MATLAB provides time series functions that enable users to explore the dynamics of an arbitrarily large set of time series with a variety of measurement and numerical computations (Kugiumtzis & Tsimpiris, 2010). Univariate time series analysis can even be done using Microsoft Excel, which is a practical solution for organizations lacking any extra budget for forecasting software. Lastly, Knime, Rapidminer, and Orange also have time series extension with windowing operator or data-emitting widgets, which not only allow for the exploration of data patterns but can also forecast by means of standard machine learning techniques.

### Service Section

**Main Application Fields:** Forecasting

**Limitations and Pitfalls:** Time series analysis is a univariate analysis using historical data of a variable to predict the future. Thus, it assumes history will repeat itself. It does not explain the causal relationships that affect the variable. Moreover, it also requires consistent historical data.

**Similar Methods and Methods to Combine with:** In addition to what was explained in this chapter, there are alternative time series methods, such as Double exponential smoothing, Holt–Winters exponential smoothing, ARIMA, etc. Regardless of the method used, combined forecasts result in better forecasts than individual models.

**Additional Data:** The dataset and a detailed EViews workflow is available at: <https://github.com/DataScience-in-Tourism/Chapter-22-Time-Series-Analysis>

## Further Readings and Other Sources

Eviews online help. <http://www.eviews.com/help/helpintro.html>

Eviews tutorials. <http://www.eviews.com/Learning/index.html>

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. Monarsh University <https://otexts.com/fpp2/>

## References

- Agung, I. G. N. (2011). *Time series data analysis using EViews*. Wiley.
- Alegre, J., & Pou, L. (2006). The length of stay in the demand for tourism. *Tourism Management*, 27(6), 1343–1355.
- Au, N., & Law, R. (2002). Categorical classification of tourism dining. *Annals of Tourism Research*, 29(3), 819–833.
- Bates, J. M., & Granger, C. W. J. (1969). The combination of forecasts. *Operational Research Quarterly*, 20, 451–468.
- Brocklebank, J. C., & Dickey, D. A. (2003). *SAS for forecasting time series*. Wiley.
- Burger, C. J. S. C., Dohnal, M., Kathrada, M., & Law, R. (2001). A practitioner's guide to time-series methods for tourism demand forecasting—A case study of Durban, South Africa. *Tourism Management*, 22(4), 403–409.
- Cho, V. (2003). A comparison of three different approaches to tourist arrival forecasting. *Tourism Management*, 24(3), 323–330.
- Choi, H., & Varian, H. (2012). Predicting the present with Google trends. *Economic Record*, 88, 2–9.
- Coshall, J. T. (2005). A selection strategy for modelling UK tourism flows by air to European destinations. *Tourism Economics*, 11(2), 141–158.

- Enders, W. (2008). *Applied econometric time series*. Wiley.
- European Cities Marketing. (2019). The European cities benchmarking report. Dijon, France.
- Frechting, D. (2012). *Forecasting tourism demand*. Routledge.
- Glen, S. (2016). *Unit root: Simple definition, unit root tests*. From [StatisticsHowTo.com](https://www.statisticshowto.com/unit-root/): Elementary Statistics for the rest of us! <https://www.statisticshowto.com/unit-root/>
- Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3), 424–438.
- Gunter, U., & Önder, I. (2016). Forecasting city arrivals with Google analytics. *Annals of Tourism Research*, 61, 199–212.
- Kugiumtzis, D., & Tsimpiris, A. (2010). Measures of analysis of time series (MATS): A MATLAB toolkit for computation of multiple measures on time series data bases. *arXiv preprint:1002.1940*.
- Kulendran, N., & Wong, K. K. (2005). Modeling seasonality in tourism forecasting. *Journal of Travel Research*, 44(2), 163–170.
- Li, H., Hu, M., & Li, G. (2020). Forecasting tourism demand with multisource big data. *Annals of Tourism Research*, 83, 102912.
- Li, G., Song, H., & Witt, S. F. (2005). Recent developments in econometric modeling and forecasting. *Journal of Travel Research*, 44, 82–99.
- Li, G., Wong, K. K., Song, H., & Witt, S. F. (2006). Tourism demand forecasting: A time varying parameter error correction model. *Journal of Travel Research*, 45(2), 175–185.
- Lim, C., & McAleer, M. (2002). Time series forecasts of international travel demand for Australia. *Tourism Management*, 23(4), 389–396.
- Maddala, G. S., & Kim, I. M. (1998). *Unit roots, cointegration, and structural change*. Cambridge University Press.
- Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., . . . Winkler, R. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2), 111–153.
- Makridakis, S., & Hibon, M. (2000). The M3-competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4), 451–476.
- Makridakis, S., & Winkler, R. L. (1983). Averages of forecasts: Some empirical results. *Management Science*, 29, 987–996.
- Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. Wiley.
- Önder, I. (2017). Forecasting tourism demand with Google trends: Accuracy comparison of countries versus cities. *International Journal of Tourism Research*, 19(6), 648–660.
- Önder, I., & Gunter, U. (2016). Forecasting tourism demand with Google trends for a major European city destination. *Tourism Analysis*, 21(2–3), 203–220.
- Önder, I., Gunter, U., & Gindl, S. (2020). Utilizing Facebook statistics in tourism demand modeling and destination marketing. *Journal of Travel Research*, 59(2), 195–208.
- Palmer, A., Montano, J. J., & Sesé, A. (2006). Designing an artificial neural network for forecasting tourism time series. *Tourism Management*, 27(5), 781–790.
- Pan, B., & Yang, Y. (2017). Forecasting destination weekly hotel occupancy with big data. *Journal of Travel Research*, 56(7), 957–970.
- Pesaran, M. H., & Shin, Y. (1995). *An autoregressive distributed lag modelling approach to cointegration analysis* (Cambridge Working Papers in Economics 9514).
- Rivera, R. (2016). A dynamic linear model to forecast hotel registrations in Puerto Rico using Google trends data. *Tourism Management*, 57, 12–20.
- Song, H., & Li, G. (2008). Tourism demand modelling and forecasting—A review of recent research. *Tourism Management*, 29(2), 203–220.
- Song, H., & Witt, S. F. (2006). Forecasting international tourist flows to Macau. *Tourism Management*, 27(2), 214–224.

- Witt, S. F., & Song, H. (2002). Forecasting tourism flows. In A. Lockwood & S. Medlik (Eds.), *Tourism and hospitality in the 21st century* (pp. 106–118). Elsevier Butterworth-Heinemann.
- Xiang, Z., & Pan, B. (2011). Travel queries on cities in the United States: Implications for search engine marketing for tourist destinations. *Tourism Management*, 32(1), 88–97.
- Yang, X., Pan, B., Evans, J. A., & Lv, B. (2015). Forecasting Chinese tourist volume with search engine data. *Tourism Management*, 46, 386–397.
- Yu, G., & Schwartz, Z. (2006). Forecasting short time-series tourism demand with artificial intelligence models. *Journal of Travel Research*, 45(2), 194–203.



# Agent-Based Modelling



## An Approach to Integrate Emerging Tourism Interactions with the Environment

Jillian Student

### Learning Objectives

- Illustrate the opportunities provided by agent-based modelling to study complexity, interactions, heterogeneity, nonlinearity, and uncertainty
- Explain how to set up an agent-based model based on a research question
- Appreciate the limits and challenges tourism researchers and practitioners face when applying agent-based modelling
- Demonstrate how to apply agent-based modelling in a tourism context

## 1 Introduction and Theoretical Foundations

When people book a vacation, they want to experience a certain location, particular activities at affordable prices, desirable weather conditions, no queues or other hassles, etc., and they plan accordingly to make these wishes come true. This seems relatively straightforward. However, there are many factors beyond an individual's control and their booking decisions that can affect the vacation and the individual's satisfaction towards their vacation. For instance, other people at the attraction may block the view of the exhibit, prices may skyrocket, a hurricane may make activities unavailable and render the location dangerous, or a person may have to queue an unbearably long time to get on the gondola at a ski lift. These events range in their seriousness with regard to health and safety.

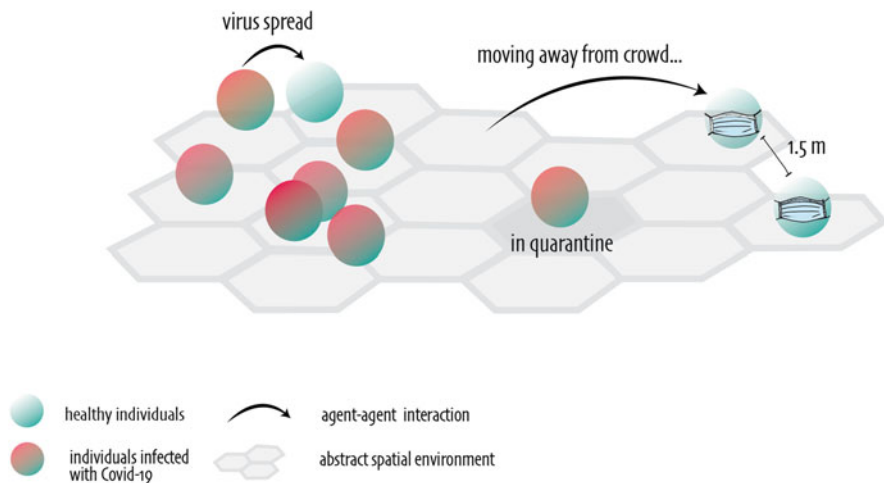
---

J. Student (✉)

Environmental Policy Group and the Wageningen Institute for Environment and Climate Research (WIMEK), Wageningen University and Research, Wageningen, The Netherlands  
e-mail: [jillian.student@wur.nl](mailto:jillian.student@wur.nl)

In our current times, dramatic, unexpected changes to tourism have occurred due to the COVID-19 pandemic, where travelers contributed to the global spread of the virus (Neuburger & Egger, 2021). Consequently, local and global tourist movement was brought to a halt or, at least, extremely limited. In 2020, there were multiple attempts to reopen the borders to help tourism-dependent economies recover. However, the potential threat of the virus spreading remained. This desire for increased mobility while limiting the spread of the virus relies on human behaviour, creates uncertainties, and generates trade-offs for decision-makers. With little information, decision-makers need to decide how *much* to open or close their borders and deliberate on whether sweeping decisions or localised lockdowns would be more effective. In this respect, Assaf et al. (2021) highlight the importance of understanding changing consumer behaviour to be one of the key issues in post-pandemic research.

Agent-based modelling (ABM) can help researchers to understand how individual choices and actions lead to systemwide consequences. In this chapter, the acronym ABM is used to describe both “agent-based modelling” and an “agent-based model”. In its basic form, ABM is a system containing agents, which can be individuals or entities, that make decisions autonomously by following simple rules and interact with each other and/or the spatial system (Bonabeau, 2002). For example, Dignum et al. (2020) applied ABM to include diverse human behaviours and explore the consequences of different policy interventions on the spread of the virus in order to complement the epidemiological models used in informing decision-makers during the pandemic. Figure 1 depicts a simple agent-based representation of infected and healthy individuals and their interactions in an abstract spatial setting.



**Fig. 1** Visualisation of interactions between agents who are infected with COVID-19 and those who are healthy in a simple spatial setting

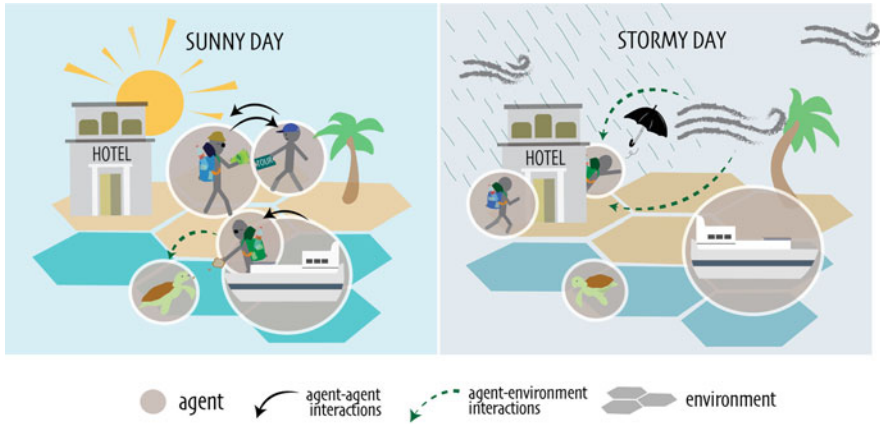
Overall, this chapter aims to introduce ABM concepts and underline how ABM can improve the understanding of real-life tourism challenges. Moreover, this chapter provides tourism-related examples of this method and one in-depth practical demonstration.

## 1.1 *Tourism as a Complex System*

Complexity is a recognised characteristic of tourism due to the many levels of interactions, heterogeneity, nonlinearity, and the resulting uncertainty (Baggio, 2008; Baggio & Sainaghi, 2011; Boavida-Portugal et al., 2017; Johnson et al., 2016; Student et al., 2016b; 2020). In a similar vein, systems thinking is a holistic means to analysing interactions (e.g. Amelung et al., 2016). In Amelung et al. (2016), ABM is argued to be a tool that is particularly useful in systems thinking and integrating the socio-economic aspects of tourism with anthropogenic environmental change. Agent-based modelling (ABM) is a form of modelling that helps to understand the complexity of interactions and tries to make them transparent and interpretable. Furthermore, ABM studied systems are often classified as complex adaptive systems, which are systems characterised by nonlinear interactions and feedbacks, emergent properties, and where agents can adapt individually or at varying macro levels (Macal & North, 2010; van Dam et al., 2013). This next section unpacks the characteristics of interactions, heterogeneity, nonlinearity, uncertainty, and mobility further.

Interactions are actions that lead to responses or feedbacks, which, in the tourism system, are ongoing, can involve many individuals, and are part of a dynamic analysis (Student et al., 2020). Ongoing interactions lead to changes in both the individual agents and the system over time. To illustrate this with an example, interactions can be seen as direct exchanges between two individuals: with one individual, a tourist, purchasing an excursion (such as swimming with sea turtles) from the other individual, an excursion seller (see Fig. 2). Yet, interactions can also take place indirectly since one change can set off a chain reaction of other events through feedbacks. Placed in the context of a wider system, this one transaction of buying a sea turtle excursion can, in turn, motivate other companies to offer the same excursions in order to earn money, which would increase congestion of the sea turtle swimming areas due to more boats with tourists visiting at the same time. Figure 2 shows examples of direct interactions, while Fig. 3 shows how direct interactions such as excursion transactions indirectly affect the pollution level at the destination.

Moreover, the increased sales of excursions could contribute to the sea turtles' weight gain as more tourists and operators feed them so as to attract the turtles to the tourists. In the described context, the buying of sea turtle excursions collectively and indirectly influences sea turtles' diet and weight. However, the relationship between a tourist buying a swimming excursion does not necessarily have a linear relationship with the increased weight of sea turtles. Some operators could decide that they



**Fig. 2** Visualisation of two states of a tourism destination ABM with a sea turtle attraction – agents, environment, agents and environmental characteristics, different interactions (agent-agent, agent-environment), and changes to the system can be noted

do not feed the turtles or only visit sea turtle swimming areas during off-peak periods. Conversely, sea turtles could have a consistent weight and health up until a certain number of excursion boats with tourists and food arrive. Above this threshold, sea turtle weight could exponentially increase, and health decrease, because of excess food, the stress of being surrounded by boats, and limited access to natural food sources due to the interference of tourists. This would create a positive feedback of sea turtles accepting food provided by tourism operators and tourists.

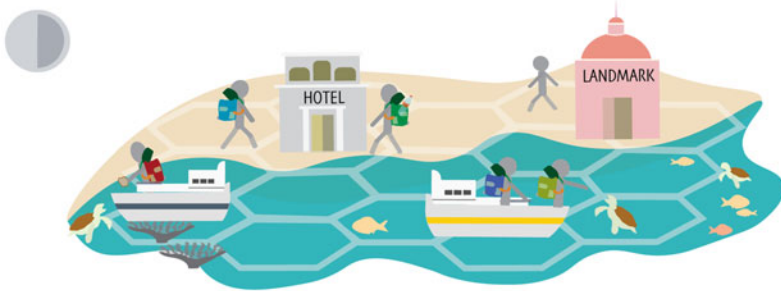
Nonlinear relationships or feedbacks makes it more challenging to plan what kind of activities and services to offer and what kind of changes to prepare for (Baggio & Sainaghi, 2011; Johnson et al., 2016; Levin et al., 2013; Student et al., 2020). For example, governments and researchers want to better understand how tourism will develop in the Antarctic (Student et al., 2016a), but, currently, tourism is self-regulated in the Antarctic through the member organisation International Association Antarctica Tour Operators (IAATO). Membership is voluntary. As such, the ABM explored the implications of new operators entering the Antarctic cruise tourism without the same long-term commitment of established tourism operators. Moreover, it looked at different motivations for collaboration with IAATO (commitment to the environment and importance of public image) to explore possible tipping points of tourism operators' ability to self-regulate.

Heterogeneity is another important characteristic of tourism that contributes to complexity as there is a considerable difference between tourists and other integrated stakeholders (Johnson et al., 2016). There are even differences within stakeholder types. Some examples within the tourist type include differences in terms of their activities, destination preferences, willingness to try new things, contribution to waste, and spending capacity. Moreover, as in the previous example, heterogeneity impacts linearity of feedbacks, i.e., since different tourism operators have distinct

Starting system state



Agent changes in system



Agent & environment changes in system



**Fig. 3** Visualisation of the changing state of a tourism destination ABM with sea turtles and growing tourism over time

strategies, the pressure on the sea turtles and their weight may be mitigated by those who decide not to overfeed or, in contrast, those who do feed more because operators are competing for the attention of a limited number of sea turtles. ABM is capable of simulating agents, interactions, and spatial heterogeneity, which ultimately helps to capture the real system and the emerging changes (Bonabeau, 2002; Macal, 2016; Student et al., 2020).

Tourism is also characterised by a high degree of movement with regard to people and goods. Environmental mobility is a concept that focuses on movements of people, entities, and environmental factors as well as their associated impacts (e.g. Boas et al., 2018; Urry, 2000). Mobilities further contribute to tourism complexity in contexts such as understanding consumer demand (Gössling et al., 2012), overtourism (Milano et al., 2019), mitigating CO<sub>2</sub> emissions (Peeters & Dubois, 2010), crisis recovery (Assaf et al., 2021), and destination or event planning (Lew & McKercher, 2006). Tourism activities occur at a destination, but tourists tend to come from outside of the destination and are not limited to one location at the destination. At the same time, food and materials to build infrastructure are often imported to the destination to support tourism activities such as skiing, safaris, beach visits, and cultural events. Figure 3 illustrates sea turtle movement with the tourists and tourism operators following their movement. These factors limit decision-makers' control and create uncertainty. Moreover, movement and interactions are difficult to express in models that lack a spatial context. ABM, however, can include the movement of individual agents (humans or other entities) over time in the analysis of individual changes, patterns, or macro-level changes (Heppenstall et al., 2012).

As illustrated in the previous paragraphs, many tourism-related research questions have to consider interactions, nonlinear feedbacks, heterogeneity, uncertainty, and mobility in complex systems (Baggio & Sainaghi, 2011; Johnson et al., 2016; Student et al., 2020). These system characteristics have far-reaching consequences, such as the spread of COVID-19 during the pandemic. However, these tourism system characteristics are difficult to simulate when applying many modelling approaches (Johnson et al., 2016). For instance, simple linear or reductionist models can obscure crucial interactions that lead to inconsistent system-level change (Levin et al., 2013). Although aggregating heterogeneities does give information on general trends, it does not easily show what happens to whom or to specific parts of the system and how they can adapt. Therefore, agent-based modelling (ABM) is a possible integrative solution for researchers wanting to study these challenges in complex adaptive systems.

## 1.2 *ABM Benefits*

ABM lends itself to complex adaptive systems research as it helps conceptualise and investigate complexity, interactions, heterogeneity, nonlinearity, movement, and emergence of unexpected phenomena from individual interactions in a spatial setting measured across a simulated time frame (see, for example, Bonabeau, 2002; Heppenstall et al., 2012; Levin et al., 2013; van Dam et al., 2013). In tourism contexts, this method is not only beneficial for capturing heterogeneous interacting agents (e.g. tourists from different areas, tourism operators providing different activities, etc.), but it can also incorporate nonlinear relations and complexity aspects useful for investigating human-environmental systems (Amelung et al., 2016; Balbi

& Giupponi, 2010; Boavida-Portugal et al., 2017; Macal & North, 2010; Müller et al., 2014; Student et al., 2020). In addition, ABM helps capture agency, such as actions, reactions, and interactions, that are missing or more difficult to capture in many other types of descriptive system models (Bonabeau, 2002; Heppenstall & Crooks, 2019; Macal, 2016). Moreover, the ABM approach is very flexible in defining and constructing use cases, which, in turn, makes it well-suited for interdisciplinary and transdisciplinary research topics (Amelung et al., 2016; Macal, 2016; Student et al., 2020).

A few of the commonly stated benefits of ABM include the following (Amelung et al., 2016; Bonabeau, 2002; Étienne et al., 2011; Macal, 2016; Macal & North, 2010; Nicholls et al., 2017):

1. Bottom-up approach, which can be more intuitive to model
2. Study of emergence (aggregating patterns stemming from individual interactions)
3. Ability to test different scientific theories
4. Flexible research approach that can be combined with participatory methods such as serious gaming
5. Supports decision-making processes by exploring multiple future scenarios

An overview provided by Nicholls et al. (2017) details potential applications of agent-based modelling in the context of tourism. More specifically, they highlight its use for (1) testing tourism theories and/or social theories applied in a tourism context, (2) simulating visitor flows for destinations or events, (3) experimenting with potential changes to destinations over time, and (4) supporting decision-making for marketing and planning. The following sections describe the conceptual background of this modelling approach, its main features, and its applications in tourism. In addition to these benefits, ABM can help simulate not only visitor flow, but also the movement of tourism operators, environmental factors, such as the sea turtles mentioned in the previous examples, and goods in a spatial setting. It can include randomness in the system set-up, the order of procedures, and possible outcomes of interactions. As such, ABM can support decision-making for marketing, general destination management, and planning under conditions of uncertainty. Since tourism is a challenging phenomenon to study in a lab environment, ABM offers a suitable space to experiment with different social and environmental challenges without harming people and the environment.

### ***1.3 Background of ABM***

ABM is rooted in systems thinking, complexity studies, complex adaptive systems, and ecological and societal adaptation, being a form of modelling that integrates human–environment interactions in the context of a system (e.g. van Dam et al., 2013). In itself, it is technically not new and evolved from other modelling processes such as cellular automata (Macal & North, 2010). However, unlike cellular automata, ABM distinguishes itself in that agents can move and carry their individual information with them. This is particularly interesting for tourism because movement, as discussed in the previous section, is an integral characteristic of tourism.

Since the 1970s, the application of ABM has spread to multiple fields, such as ecology, economics, and the social sciences, and has been applied to policy-oriented topics to support decision-making (see, for example, Macal, 2016; van Dam et al., 2013). As ABM can study phenomena emerging from individual interactions and decisions (Heppenstall et al., 2012; Macal, 2016; Student et al., 2020; van Dam et al., 2013) and is not limited to studying aggregates but, rather, is designed from the bottom up (e.g. Bonabeau, 2002; Macal, 2016), this sets it apart from other forms of simulation. In addition, this also means that it includes information about individual parts of a system, taking an agent perspective and including social processes, and how they interact with the natural environment (Macal, 2016).

A classic, early example of ABM concepts and its ability to look at individual decisions and how patterns emerge is the Schelling model (Schelling, 1971). This model investigated how residential preferences influenced the formation of racially segregated neighbourhoods. By applying very simple rules for the households (either to remain in their location or move), Schelling looked at how differences in household preferences influenced the general locations of the two distinct household types. As such, this simple model showed the power of ABM because it revealed to what extent small differences in individual preferences can lead to unexpected macro-level trends, such as segregation.

In the context of tourism, modelling different preferences can be relevant to many research/practical questions including destination preference (Boavida-Portugal et al., 2017) and the willingness of tourism operators to collaborate, act alone, or avoid action in the face of environmental challenges (Student et al., 2020b). ABM applications first emerged in tourism in the 2000s (e.g. Chhetri & Arrowsmith, 2008; O'Connor et al., 2005; Yin, 2007), and by the 2010s, these had continued to increase and appear in more academic tourism journals (e.g. Alvarez & Brida, 2019; Balbi et al., 2013; Boavida-Portugal et al., 2017; Johnson & Sieber, 2010, 2011; Li et al., 2015; Pirota et al., 2014; Pizzitutti et al., 2014; Pons et al., 2012, 2014; Soboll & Dingeldej, 2012; Soboll & Schmude, 2011; Student et al., 2016a; Zhai et al., 2019). As time continues to pass, the number and types of ABM applications also continue to expand; thus, the following section aims to describe ABM's basic components in order to help new ABM researchers further develop this approach within the tourism domain.

## ***1.4 Key ABM Features***

The main features of agent-based models are agents, variable types, and the concept of ongoing interactions in a spatial setting over time. There are three main types of variables: agent, environmental, and system-level, which can either be static (stay the same throughout the simulation) or dynamic (change through interactions and successive events) (e.g. Heppenstall et al., 2012; Macal & North, 2010; van Dam et al., 2013). An example of a static feature of an agent could be that the agent is classified as a tourist throughout the simulation, while an example of a dynamic feature could be the agent's spending capacity. Figure 2 depicts the agent, the agent's



surrounding environment, and the interactions amongst agents and between agent and the environment.

### 1.4.1 Agents

Naturally, there are different definitions of agents (e.g. Macal, 2016); however, in a tourism context, agents can be thought of as individuals or entities who are distinguishable, having their own outward features and internal rules for making decisions and interacting with other agents and/or their surrounding environment. Agents, and in some cases the environment, can possess memory and can learn, change strategies, and have changing adaptive capacities, which can be influenced by successive events (Bonabeau, 2002). The ability to study agents that have memory and can learn is useful for studying complex adaptive systems. Moreover, ABM does not need to assume perfect knowledge for agents as agents often have limited knowledge about other agents and the wider simulated system, i.e., they know more about those in close proximity, of the same type, or with whom they have previously encountered or worked with (e.g. Macal, 2016; van Dam et al., 2013).

The modeller needs to determine who (which agents) should be included in the system, their key characteristics, and what rules govern their actions. Further questions to consider are with whom and what they interact with directly. Do they interact directly with other agents and the environment? It is also important to consider whether the agents simply react or if they adapt their behaviours during the simulation, i.e., learn from previous interactions and change their strategy or alter the decision rules. One of the benefits of ABM is that the agents are not limited to “if this, then do this” rules as they can be derived from mathematical algorithms, behaviour theories, empirical data, statistics, rules of logic, and/or incorporated randomness (Macal & North, 2010).

### 1.4.2 Environment

In ABM, the agent is typically connected to a spatial environment (depicted in Figs. 1, 2, and 3). The environment can be an abstract concept like in Fig. 1 and show location relative to others, or it can have features that represent a particular setting such as a landmark or destination, as depicted in Fig. 2. Alternatively, the spatial environment can be spatially realistic and derived from GIS maps. The different parts of the environment depend on what needs to be represented and the level of detail required, and, like agents, the environment can also record changes or have a memory (Bonabeau, 2002). How the spatial environment is set up can limit agents’ interactions and their access to information (Macal & North, 2010). For example, in Fig. 1, the spatial environment indicates how close agents are to each other, and the agent’s range of sight may only be to that of its neighbouring cell. One further example, in Fig. 3, involves the fact that only certain boats are close enough for the tourists to see and feed the sea turtles. Another important consideration to

take into account is the simulated environment's spatial boundaries and what these boundaries represent (Macal, 2016; van Dam et al., 2013). Boundaries can be explicit limits to the agents' movement, or they can be open for agents entering and exiting the system. Alternatively, they can also wrap around in torus geometries, i.e., an agent going beyond the right border re-enters the system from the left border.

### 1.4.3 System-level

System-level variables are settings that apply to the whole system and can be accessed by both the agents and the environment for different model mechanisms. Some examples of system-level variables could be transportation costs, visitation number quotas, the temperature, and the rate of sea-level rise.

### 1.4.4 Interactions

Interactions are the links between agents, the environment, and the system, which shape the model mechanisms and are ongoing throughout the simulated time (Bonabeau, 2002). In ABM, the range of interactions can vary; for example, they can be localised and spatially close by to other agents who/that are spatially close or agents with specific characteristics (e.g. the sale of an excursion to a specific tourist with an interest in sea turtles). However, interactions can also include everyone in the system, such as the effect of weather conditions on tourism activities depicted in Fig. 2. Another key feature of interactions is how they change both the agents and the environment over time. In the example of buying a sea turtle excursion, the exchange of funds for a ticket is between two agents: one agent, the tourism business, will receive money, while the other agent, the tourist, will receive a ticket. The accumulation of multiple individual interactions affects the system over time, which is illustrated in Fig. 3. Here, the interactions of multiple people joining the excursions to visit sea turtles increases congestion in the waters surrounding the sea turtles, eventually contributing to more waste ending up in the water and the waters becoming more polluted.

The mechanisms of agent-agent and agent-environment interactions do not have to be deterministic algorithms, rather, they can include randomness (Heppenstall et al., 2012; Student et al., 2020b). This helps simulate the stochasticity and uncertainty present in the real world. In the ticket-buying example, an agent has decision rules that determine the excursion choice. This could indicate that agents usually buy tickets from the cheapest seller, for instance. However, occasionally, an agent may buy a ticket from the seller closest to their current location because of convenience. Figure 3 illustrates changes over time, both at the individual level (e.g. local agents not going to landmarks, the location of individual boat excursion operators) and the system-level (e.g. increasingly congested and polluted waters). This figure also shows how some of the operators' location choice does not match with the sea turtles' presence in the water, representing the randomness and uncertainty of movement.

## 1.5 Challenges When Applying ABM

Although ABM can be a useful method for tourism and other research domains, it is still considered more of a niche method when compared to other modelling approaches. General challenges with ABM include deciding on the amount/level of details to incorporate, drawing the environment's boundaries, considering time for model scheduling, model verification, and validation, in addition to other tourism research-related challenges.

As agent-based modelling deals with complexity, it makes it challenging to keep the model simple, and it can be tempting to include more details than necessary. Figure 1 depicts a simpler ABM simulation than Figs. 2 and 3 in terms of agents' characteristics, spatial environment, and interactions. Yet, the detail provided in Figs. 2 and 3 may not be relevant or necessary to answer the question of the effects of agent mobility, location, and mask-wearing on the percentage of the population that is healthy. It is true that representing behaviour and decision-making often include parts of mechanisms that are not observable; nonetheless, having more details requires more processing power and increases the chance of making errors. As such, Heppenstall and Crooks (2019) summarise some of the current challenges related to ABM in the spatial sciences, for instance, the availability and quality of data as well as translating vast amounts of geo-referenced data to specific agent behaviours. This "dance" between empirical data details and theoretical abstraction creates challenges in model development and analysis.

In the same vein, drawing the boundaries around a system, and a tourism system in particular, is challenging since boundaries are often ill-defined, even when looking at a tourism destination or venue, due to tourism-related agents' in and outbound movements. This necessitates a decision on what should be included in the model, what is considered external, who and/or what can enter or exit the modelled system, and what this means for tourism mobility.

Temporal boundaries also require specification. While in real life, events happen concurrently, in agent-based modelling, the timing or scheduling of events or agent actions needs to be itemised (Heppenstall et al., 2012; van Dam et al., 2013). This is not a trivial task as the order in which things occur can have a large effect on the outcome. For example, if all agents first go to the ticket office and then go on the excursion, there will be a queue, and a certain number would get tickets and then join the same excursion. Alternatively, the model could be scheduled in a way that agent "A" first completes both steps before Agent "B" does. In this case, agent "A" would first go to the ticket and encounter no one else and then go on an excursion alone before agent "B" goes to the ticket office to get a ticket and go on a separate excursion. The latter case seems less plausible in this instance, but, for other modelling queries, scheduling based on all agents completing a step might be necessary to mirror concurrent decisions.

Incongruent time scales further complicate this issue of scheduling. A model defines discrete time steps in which the schedule of events or agent actions occurs (van Dam et al., 2013). However, many scientific domains study time at different

scales, and the rate of change for a phenomenon being studied determines the relevant time scale. In tourism, this is complicated as questions often draw on insights from multiple scientific domains. Figure 2, for instance, includes the weather, which can change rapidly, while, in addition, arrival numbers change daily, and decisions can be made instantaneously or can be deliberated over time. On this note, the timing of different model interactions needs to be calibrated to the selected time step.

The previously mentioned challenges complicate the verification and validation processes as well. Some parts of an agent-based model can be challenging to verify, depending on the availability of observable data (e.g. Macal, 2016). Similarly, verification and validation are particularly difficult as ABM has various forms based on the modelling goal, such as theory testing vs. exploration or prediction (e.g. Ligmann-Zielinska et al., 2020; van Dam et al., 2013). One consideration when it comes to validation is whether empirical data can even be used to compare model behaviours with model mechanisms and outcomes (Heppenstall et al., 2012). If empirical data is unavailable, then experts can help validate parts of the model (e.g. Bonabeau, 2002). Lastly, another challenge relates to communicating the model's mechanisms and results with stakeholders (Johnson et al., 2016). Acceptance of the model and their findings is not guaranteed, and transparency and trust in ABM processes and analysis are ongoing dilemmas (e.g. Heppenstall & Crooks, 2019; Macal, 2016; van Dam et al., 2013).

## **1.6 Tourism-related ABM**

ABM continues to develop in many scientific fields, in spite of the challenges detailed in the previous section. Nevertheless, there are still relatively few ABM publications present in tourism journals. Johnson et al. (2016) identified three main challenges that tourism researchers and tourism practitioners face when considering the application of ABM:

1. The technical abilities of tourism researchers to translate the conceptual understanding of the problem into an agent-based model.
2. Communicating the model's mechanisms and findings with other tourism researchers and practitioners.
3. Having examples to build upon.

With regard to technical abilities, although ABM is also suitable for novice modellers (Macal, 2016), many researchers are unaware of where to start or have not undergone a formal training as part of their tourism studies (Johnson et al., 2016). The following sections of this chapter thus provide steps and references that can help inexperienced modellers to conceptualise and set up their projects. Furthermore, the resources at the end of the chapter provide the reader with online platforms and an in-depth description of the modelling process.

The second challenge of communicating ABM processes and results is difficult and often haphazard. However, there has been growing support to follow standardised forms in order to successfully communicate the model process, outcomes, and limitations with other researchers and stakeholders. Additionally, the model description is also becoming more standardised so that describing an ABM model’s purpose, its conceptual framework, and its mechanisms comes with more ease (e.g. Grimm et al., 2020; Müller et al., 2013). When communicating a model’s findings with practitioners and other stakeholders, it is vital to be clear about the model’s purpose and explain, in terms of outputs, whether it is exploratory or predictive.

Examples help to accelerate the uptake of ABM; although ABM is still a relatively niche approach in tourism studies, there is a growing body of projects and publications in general as well as in the tourism context specifically. It is suggested to use existing examples from repositories and forums, similar to the one included at the end of this chapter. Nicholls et al. (2017) provide a summary of some earlier examples, and Table 1 gives an overview of some of the models developed for the tourism industry not included in Nicholls et al.'s (2017) summary.

**Table 1** Examples of tourism-related agent-based models

Topic	Focus	Reference
Housing-market regulation and Airbnb growth	Tourism destination management	Vinogradov et al. (2020)
Destination choice	Tourist preferences	Alvarez and Brida (2019)
Destination preference change	Tourist preferences	Boavida-Portugal et al. (2017)
Recreation potential of nature areas	Tourism destination management	Chhetri and Arrowsmith (2008)
Tourism activities on Galapagos; scenarios of environmental change affecting markets	Tourism market/destination management	Pizzitutti et al. (2014)
User-generated content and strategic destination management	Destination management	Zhang et al. (2020)
Social media response to human-induced crises	Destination management	Zhai et al. (2019)
Evaluating the potential disturbance of tourism on dolphin activities and shark predation risk	Tourism–environment interactions	Pirotta et al. (2014)
Exploring emerging vulnerabilities in a coastal tourism setting	Tourism–environment interactions, destination management	Student et al. (2020b)
Individual visitors’ multi-destination travel patterns and spill-over effect to other destinations	Visitor flow	Li et al. (2021)
Determinants of forest visitor spatial patterns	Visitor flow	Li et al. (2015)

## 2 Practical Demonstration

This section provides a step-by-step guide and presents some tools for applying ABM to tourism research as well as what to consider when communicating the agent-based model to others. In the **research case section**, an example of a tourism-focused agent-based model is described in detail. The basic steps, defining the model purpose, developing the model's conceptual set-up, writing the model description, determining model components, selecting the appropriate software, and analysing the mechanisms and outputs are discussed.

### 2.1 *Defining the Model Purpose*

First, it is important to check whether ABM is the right tool for the problem at hand. If there are sufficient aggregate variables or averages of the agents, then another modelling tool, such as systems dynamics modelling, may be more appropriate. Examples of such a case would include considering the number of arrivals as a function of adding or removing different flight routes from a destination. As mentioned in the introduction, ABM is a tool that helps deal with complexity, heterogeneity, nonlinearity, bottom-up and top-down emergence, and interactions between multiple (individual) agents and their environment. Moreover, in cases where studying the movement of people or entities in combination with decision-making is critical, ABM can provide more user-friendly ways of establishing movement rules for individuals as compared to mathematical equation-based modelling (e.g. Heppenstall et al., 2012; Macal & North, 2010). In addition, ABM can enrich GIS models when individual decision-making is important to consider.

Thus, to determine whether ABM is appropriate for researching the question at hand, the following questions can be used as a good starting point:

1. What is the question I am trying to answer with my model?
2. Does the model need to take individual agents, heterogeneity, and complex interactions into consideration?
3. Are mobility and the movement of people, information, or goods essential for answering the research question?
4. Are the desired outputs only at an aggregate level or are other types of outputs required (by type, at different moments in time, for parts of the system)?

### 2.2 *Conceptual Model Set-up*

If ABM is appropriate for answering the above-mentioned questions, then the modelling goal also needs to be taken into account:

1. Is the model based on empirical data and attempting to predict a specific output? Is it explorative with regard to a problem, a (theoretical) concept, or a situation?
2. What kind of output is needed to answer the modelling question?
3. What types of inputs are needed and for which of these is data available?

These questions help determine the level of precision required for the simulated agents, environment, and interactions. While exploratory and theory testing models may require little to no necessary empirical data, prediction models often require robust data for verification and validation.

## 2.3 *Model Description*

One good measure for conceptually and technically setting up the model is having a blueprint or model description. Although there are ongoing debates on which format is the most appropriate type of model description (Müller et al., 2014), one format that is often applied to agent-based models is the Overview, Design concepts and Details (ODD) protocol (Grimm et al., 2006, 2010, 2020). In an updated version, Müller et al. (2013) proposes the ODD + Decision-making protocol (ODD+D) to emphasise human decision-making. Either way, the ODD or ODD+D is a useful starting point to organise one's thoughts as the protocols help to clarify what the model's goal and mechanisms are. Moreover, a clear ODD (+D) increases model transparency, enables other researchers to reproduce the model, and is often required if the findings are submitted to a scientific journal.

## 2.4 *Model Components*

*CoMSES OpenABM* (see section **Further Readings & other Sources**) provides an extensive model library and is a great resource for education and research purposes and a platform to ask questions.

### 2.4.1 **Agents**

In relation to agents and their characteristics, some key questions to ask include:

1. Which tourism stakeholders will be included (e.g. tourism operators, tourists, airlines, etc.)? What level of heterogeneity is necessary (e.g. tourists with different spending power, operators offering different activities)? What types of other stakeholders are included (e.g. local communities, government bodies, the police)?
2. What kind of behaviours need to be included (e.g. the choice of location, decisions on whether to invest and what to invest in)?

3. What type of behavioural information is available? Which behaviour or decision rules need to draw upon theory, uncertainty (stochasticity), and/or assumptions? (Often times, some part of the decision-making needs a proxy based on theory, randomness, or assumptions).
4. Which characteristics are static and which are subject to change (e.g. an agent is a tourist throughout the simulation, but his/her preference of activity can change)?

### 2.4.2 Environment

Next, the tourism-relevant environmental features need to be defined.

1. What is the scale of the model (e.g. a venue, a destination, a country, a region)?
2. How is the environmental context characterised (e.g. geospatial type (such as coastline, beach, etc.) or location of attractions, the number of people a location can accommodate, pollution levels)?
3. What data is available?
4. What is the level of spatial specificity (e.g. abstract, reality-based, specified to a certain context)? If it is spatially explicit, are latitude and longitude values needed? (Refer to the section on **software** for literature and suggestions of possible software for GIS-specific simulations.)
5. What level of detail is necessary (e.g. how many characteristics are included, for instance, attractiveness, geospatial features, elevation, carrying capacity, pollution level, and so on)?
6. What types of heterogeneities are related to the questions that need answering (e.g. attractiveness, elevation, geospatial type)? Are the differences expressed in numerical values or in types?
7. Which characteristics of the spatial environment are static and which can change? What can influence these changes?

### 2.4.3 System-Level Variables

In addition to agents and the environmental variables specific to parts of the spatial setting, the model may have parameters that apply to the whole system. These global variables, as they are called, could include things like sea level, temperature, and exchange rates, and can be either static or dynamic.

1. Is the global variable static or dynamic?
2. If the variable is dynamic, what mechanisms influence the changes (i.e. are there external inputs or is it a result of emergence)?
3. Is there external data that initialises the global variable?
4. Is there data outside the model that is entered after initialisation? If so, how is the external data's timing determined?



#### 2.4.4 Simulated Time

Time is an important consideration in ABM, and the modeller's question determines the total simulated time that they are interested in. The time step, sometimes referred to as ticks, is the smallest relevant time unit for model processes. The greater the difference between a time step and the total simulated time, the more time for interactions to occur. However, this also requires more processing power and leaves more room for errors. Therefore, the goal is to have the smallest *relevant* time unit, not the smallest possible time unit.

Another challenge is simulating interdependent events. In real life, complex interactions occur simultaneously in continuous real time, but, in a computer model, time steps are discrete and have to be explicitly determined (e.g. van Dam et al., 2013). This distinction requires careful attention when model scheduling in order to represent parallel processes, such as a group decision of whether to go on a sea turtle excursion. For tourism, the following points are relevant:

1. What is the total simulated time (e.g. xx days, xx months, xx years)?
2. What does each individual time step represent (e.g. a second, minute, day, week, months, years)?
3. Does the simulation run until a certain threshold has been reached (all visitors have been evacuated from the venue), or does it automatically end at a predetermined time?
4. How are the events ordered in a time step? Does an event occur in every time step?
5. What is the scheduling of events? Is there a specific order in which the interactions should occur? Which interactions need to be in parallel? Does the order in which an agent performs an action matter for the model outcomes? If so, does the order of the agent's actions follow a specific order, is it randomly determined, or determined by something else?

#### 2.4.5 Interactions

Once an idea of who the key agents are and what environmental features need to be considered have been established, the mechanisms that determine how they interact and influence each other need to be developed. The ARDI (Actors, Resources, Dynamics, and Interactions) method is a series of questions that are useful for categorising the different states that the key agents and environmental features have, their potential actions, and the effects on the environment's and/or other agents' states (Étienne et al., 2011). An agent's or environment's state is the current condition of a particular variable. In the COVID-19 example from Fig. 1, a state would be whether an agent is infected or in quarantine. Categorising these system features in ARDI helps specify in what ways Agent "A" is connected to Agent "B", and how Agent "A"'s actions change Agent "B"'s state. For example, agents "A" and "B" have the variable "vacation satisfaction level"; Agent "A" buys the last

ticket available for a tour, which prevents Agent “B” from joining. Since Agent “B” cannot join the excursion, the state of “B”’s vacation satisfaction level decreases. The following questions help specify interactions and provide context:

1. What are the direct interactions with other agents? With the environment? How do the interactions affect the state of agent and/or environment variables?
2. Are the agents mobile? If so, what conditions determine their movement?
3. In what ways does the spatial setting limit the extent of the interactions (actions localised, system-level)?
4. To what degree is randomness incorporated in the interactions?
5. Does the agent hold a memory of interactions? If so, is it perfect, does it deteriorate over time, or is it influenced by other factors? How does memory influence future interactions?

## 2.5 Software

Abar et al. (2017) provide a detailed overview of different software platforms for developing agent-based models, which include the following: information on the software’s source code (e.g. *C++*, *Java*, *Python*, *Microsoft.net*), the coding language, the licence agreement (i.e. open, limited, or closed source), level of skill required, the visual interface, and typical types of model applications. The following questions can further help to determine the right software solution.

1. What level of experience does the researcher have with modelling?
2. What are the model’s objectives?
3. What is the spatial and temporal scope needed to reach the model’s objectives?
4. Is this specificity based on empirical data in relation to input data or the model’s mechanisms?

Experience in (general) coding is a key determinant for which software is appropriate and whether external assistance in developing the model may be required. *NetLogo* programming software (Wilensky, 1999) is an entry-level open-source platform for model development and was used for the model described in the **research case section**. It is a relatively easy-to-use tool for those with limited coding experience and provides a graphical user interface, several modules, and code libraries.

In addition to modelling experience, which platform to select also depends on the type of research question and the kind of inputs and outputs needed. While *NetLogo* may be useful in many tourism-related studies, its scalability and ability to handle complex input files is limited compared to other software platforms. For example, if extensive GIS is required, another software tool such as *REPASt* or modelling space in *ArcGIS* may be more suitable.

Model development and analysis do not necessarily need to be performed using the same software. One platform might be needed to develop a model and another platform to perform the analysis of the outputs, as was the case for the tourism

example presented in the next section. *NetLogo's* built-in analysis tool called *BehaviorSpace* has limited capabilities for sophisticated analysis of complex models. Therefore, the model presented in the next section was exploratory and developed in *NetLogo*, but the analysis itself was conducted in *Java* and *Python* using *PyNetLogo* and the *Exploratory Modelling and Analysis (EMA) Workbench*. *These analysis tools can perform many types of analyses and generate a variety of visualisations including interactive plots and multiplot graphs. NetLogo can be integrated with Java and Python analyses.*

## 2.6 Analysis

After defining the model's purpose, designing the conceptual model, and determining model components, modellers need to consider how they will verify or test if the code is doing what it is intended to (van Dam et al., 2013). Model processes need to be evaluated to check the extent to which the developed model can answer the modeller's question. Many analysis formats exist to address the various ABM process and output types (e.g. Hahn, 2013; Heppenstall et al., 2012; Herman & Usher, 2017; Kwakkel & Pruyt, 2013; Ligmann-Zielinska et al., 2020; Ngo & See, 2012; ten Broeke et al., 2016; Troitzsch, 2014; van Dam et al., 2013). Therefore, it is important to first consider what input types are needed and what output(s) is expected. Ideally, both the model process and outputs should be evaluated. However, due to the complexity involved, some processes cannot be validated with real-world behaviour and the modeller needs to determine whether or not the processes and/or results are plausible and convincing (Heppenstall et al., 2012; van Dam et al., 2013). As ABM often explores the uncertainty level of how interactions lead to observed outputs, sensitivity analysis (SA) is a popular analysis approach for both validating model processes and outputs. This section gives a short overview of considerations to take into account when verifying, validating, and analysing model processes and findings. For more details on specific considerations for various types of analysis, please refer to the references in the subsequent sections.

### 2.6.1 Verification

Verification is the process of testing the model logic to check that the model is doing what it is supposed to (Heppenstall et al., 2012). Yet, calibration is challenging in the ABM context due to a lack of empirical data to compare findings, and, in addition, it involves identifying the range of input values that are appropriate for the model's mechanisms (Ngo & See, 2012). Although ABM helps to explore complex systems, a more detailed model is more prone to technical errors and more computing power is required to analyse the data. Moreover, the random number generators used to integrate randomness into interactions makes every model run unique (Abdou et al., 2012). As a result, verification looks at anticipated result ranges instead of exact

outcomes and requires more runs to test model behaviour. Hence, it is advisable to take a layering approach and to test out different mechanisms separately before adding them to the main model. In this way, it is easier to troubleshoot and to ensure that each mechanism is performing as it should. For example, in the turtle-feeding case from Figs. 2 and 3, the movement of pollution mechanism could first be developed autonomously to observe whether it moves according to the expected patterns.

### 2.6.2 Validation

Validating ABM is an ongoing challenge and the subject of ongoing debates regarding what it is and what it should show (Hahn, 2013; Heppenstall et al., 2012; Heppenstall & Crooks, 2019). While verification tests model logic, validation is applied to check whether the design fulfils its purpose (van Dam et al., 2013). In other words, validation ensures that a model is actually representing what it has originally been set out to represent (Hahn, 2013). There are many forms of validation, with some examples being empirical validation, statistical validation, conceptual validation, process validation, and output validation (Ngo & See, 2012). As such, validation of agent-based models requires validating agent behaviours and emergent phenomena in addition to system-level model outcomes (Hahn, 2013). When available, model processes and outcomes can be compared with empirical data; alternatively, model outputs can be validated through consultation with experts, simulating past events where data is available, literature validation, and/or by comparing outcomes with another model applying a different modelling technique (such as systems dynamics modelling) (van Dam et al., 2013).

SA evaluates the influence of one or more inputs on specified model outputs and indicates the strength of this link (Ligmann-Zielinska et al., 2020). Furthermore, it can be local or global: local SA focuses on the effects of small changes to one or a few inputs, while global SA explores changes to all inputs within a specified parameter (input space) and can look at variability derived from single inputs or interactions among inputs (Ligmann-Zielinska et al., 2020; Saltelli et al., 2008; Saltelli et al., 2004). ten Broeke et al. (2016) describe three types of sensitivity analysis in the context of ABM and the considerations for selection: one-factor-at-a-time, model-free output variance decomposition, and model-based output variance decomposition. Ligmann-Zielinska et al. (2020) further advise reflecting on the model's purpose in order to choose the appropriate sensitivity analysis and recommend global SA whenever possible. SALib (sensitivity analysis library), for instance, is a Python-based tool, which provides multiple types of global sensitivity analysis (Herman & Usher, 2017).

### 2.6.3 Analysing Findings

Findings analysis depends on the outputs of interest. The outputs could be agent or system-level change, patterns of movement, processes, value of outputs at the end of the simulated time, the changes to output values over time, the amount of time to reach a particular outcome, or the system or agents' characteristics when a defined equilibrium has been reached (see Heppenstall et al., 2012; van Dam et al., 2013). Statistical analysis can evaluate model findings, to which Troitzsch (2014) provides guidelines for appropriate conditions to apply such statistical analyses. Moreover, SA is a statistical analysis to explore emerging model behaviours and guide decision-making (Kwakkel & Pruyt, 2013; Student et al., 2020b; van Dam et al., 2013).

In many forms of sensitivity analysis, the inputs are defined and their influence on the various outputs is observed (see the validation subsection for more details). However, for scenario discovery, the opposite is done. Scenario discovery is a form of analysis similar to backcasting, where a threshold for one or more outputs is defined and the inputs are observed (Kwakkel & Pruyt, 2013; Steinmann et al., 2020). The threshold can be defined by modellers or decision-makers and indicates a situation that one either wants to attain or avoid. The analysis then determines which input variable combinations in which range have the most influence on exceeding this threshold. In this case, the *EMA Workbench* is a Python-based open-source toolbox used for scenario discovery analysis and other exploratory analyses (Kwakkel & Pruyt, 2013). Referring back to the transmission of the COVID-19 example depicted in Fig. 1, the threshold could be that less than 20% of the population becomes infected. The analysis could show that mask-wearing and social distancing are the most influential inputs for attaining the defined goal, when more than 65% of the simulated individuals wear masks and a social distance of 1.5 metres is maintained, or that other inputs are more influential for reaching the desired infection percentage of less than 20%. Defining the outputs instead of the inputs makes scenario discovery useful for decision-support under conditions of extreme uncertainty (Lempert, 2019). The decision-making goals are predefined, and the analysis reveals the factors that should be placed under focus in order to prevent an undesirable situation or attain the goal, rather than investing in factors that have little influence on the desired outcome (Kwakkel & Pruyt, 2013).

## 3 Research Case

This section provides a walkthrough of an agent-based model developed for the tourism domain. The agent-based model “*Coasting*” was developed as part of a dynamic vulnerability approach (Student et al., 2020) with the aim of assessing emerging socio-economic and ecological vulnerabilities in relation to climate change. *Coasting*, as the name implies, simulates a coastal tourism destination



**Fig. 4** Visual description of main agents (tourism operators), environmental features, and interaction characteristics. Author image published in Student et al. (2020b)

context and is applied to the destination Curaçao (Student et al., 2020b). The image in Fig. 4 depicts some of the main agents, environment, and interactions featured in this model.

This model was first developed as a serious game with stakeholders (see Student et al., 2020), and parameters were determined based on literature and empirical insights from simulation game sessions with tourism stakeholders. Wherever information was unavailable, randomness was introduced and/or a large parameter range was explored to account for some of the uncertainty.

The actual model has many components, as can be observed in Table 2. Figure 5 depicts the interface at the set-up of a simulation run using *NetLogo* software. The model's description, which follows the ODD+D protocol format, is available in Student et al. (2020a). It includes details of the model design and mechanisms as well as the tools used for the analysis. The research question for this case was “*what are the main (interacting) factors leading to socio-economic and/or ecological vulnerabilities?*” The main output indicators of socio-economic vulnerability were tourism operators going bankrupt, as this indicated that their capacity to deal with changing economic and environmental factors was too great. Furthermore, the main system-level ecological indicator of vulnerability was the aggregated environmental attractiveness of three different parts of the coastal system. These three parts were the land-based area (where hotels and beach operators were based), the coastal area (the spaces directly bordering the distinction between water and land, depicted in light yellow and light blue, in Fig. 5), and the nearshore waters (where diving and boating activities took place). Pollution, biodiversity, and environmental degradation levels were the main dynamic characteristics of each spatial unit that contributed to each spot's attractiveness and that could change over time.

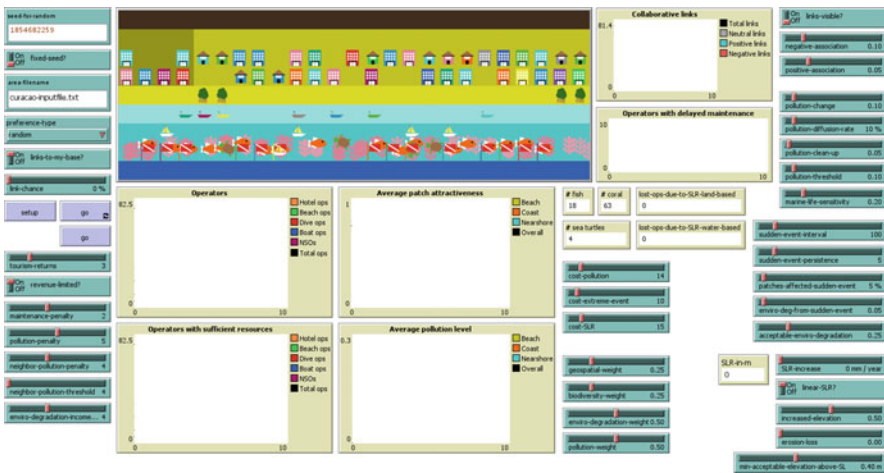
**Table 2** “Coasting” model components

Modelling goal	Exploratory	Understanding emerging socio-ecological vulnerabilities in a coastal tourism setting.
Software	Model development:	NetLogo
	Analysis: Global sensitivity analysis Scenario discovery	Java, Python, SALib PyNetLogo, Exploratory Modelling and Analysis (EMA) Workbench
Agents	Coastal tourism operators	<i>Five operator types:</i> hoteliers, beach vendors (e.g. cafes and beach activities), nearshore operators (e.g. surfing, jet ski and kayak businesses), day excursion boat operators, and dive operators. <i>Example of characteristics:</i> location selection and preference, resources. <i>Heterogeneities:</i> inputs necessary for business, location preferences. <i>Static variable example:</i> operator type, whether mobile (yes, no). <i>Dynamic variable example:</i> level of resources, links with other operators, location if water-based operator.
	Marine life and coastal resources	<i>Four types:</i> sea turtles, coral reef, reef fish, and mangroves. <i>Heterogeneities:</i> location preference. <i>Static variable example:</i> operator type, whether mobile (yes, no), sea turtle yes, mangroves no. <i>Dynamic variable example:</i> health level.
Environment	Key features simulated represent a simple depiction of a coastal area in Curaçao	<i>Static variable examples:</i> geospatial characteristics (e.g. inshore, beach*, coastal waters, nearshore waters, deep sea). *static except in the case of sea-level rise <i>Dynamic variables:</i> attractiveness of each part of the location, elevation, pollution level, environmental degradation level.
Inputs	Example of some of the input	<i>Decision-influencing inputs:</i> resources gained, income penalty for not doing maintenance, positive association with others if collaboration goes through. <i>Environmental change inputs:</i> rate and height of sea-level rise, probability, impact, and duration of sudden event affecting a part of the spatial setting, costs of addressing impacts of either pollution, sea-level rise, or environmental degradation through a sudden event.
Outputs	Vulnerability indicators focus on a proxy indicating ecological and socio-economic vulnerabilities	<i>Ecological vulnerabilities:</i> environmental attractiveness of beach, coastal, nearshore waters, and destination. <i>Socio-economic vulnerabilities:</i> number of operators bankrupt throughout the

(continued)

**Table 2** (continued)

Modelling goal	Exploratory	Understanding emerging socio-ecological vulnerabilities in a coastal tourism setting.
		scenario; this was aggregated as well as investigated for each operator type.
Analysis	Global sensitivity analysis Time analysis & scenario analysis	Completed at the end of the 30 years of simulated time. All time steps for ecological and socio-economic vulnerability indicators.
Special considerations	Model design	Designed to enable adaptation to other coastal tourism settings and add other environmental challenges.
	Data	Data collected at multiple time steps instead of only the final step.



**Fig. 5** Interface of “Coasting” simulation model set-up in NetLogo software

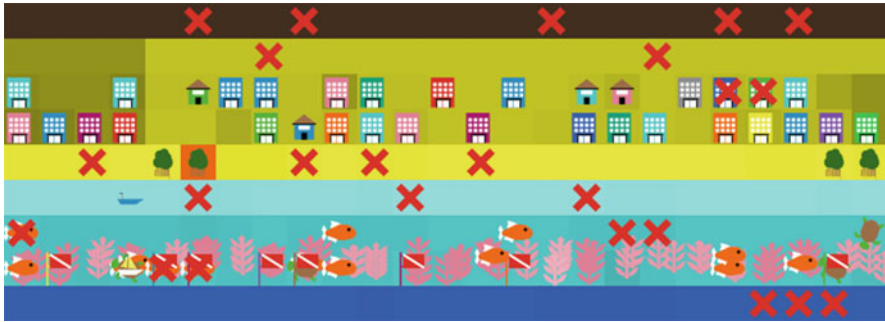
### 3.1 Contribution of Method

The *Coasting* game was developed first by applying the ARDI (Actors, Resources, Dynamics, and Interactions) method in order to generate a game that could be translated into a model (Étienne et al., 2011; Student et al., 2020).

### 3.2 Analysis

In this study, the goal was to look at how multiple human–environment interactions at a coastal tourism destination lead to vulnerability outputs. Specifically, the model





**Fig. 6** View of “Coasting” simulation with areas affected by sudden events (red crosses), pollution (squares in a relatively darker shade), and land that has been artificially raised to mitigate sea-level rise (red square)

looked at the number of actions to address environmental challenges (individual and collaborative), socio-economic vulnerabilities (bankruptcy), and ecological vulnerabilities (decreased environmental attractiveness). This study was explorative and looked at a wide combination of inputs using global SA and scenario discovery so as to assess the different number of inputs across different ranges of interaction and see the cumulative effects these inputs have on socio-economic and ecological vulnerabilities. As such, this analysis diverged from other tourism-related agent-based models.

Figure 6 shows a simulation run of a scenario with conditions of sea-level rise and sudden events. The global SA looked at first order (direct sensitivity), second order (sensitivity resulting from two interacting variables) and total sensitivity for number of actions, number of operators that stay in business, and the level of environmental attractiveness. Moreover, scenario discovery was used to identify interacting parameters that led to undesirable vulnerabilities of business loss (a proxy for socio-economic vulnerability) and environmental attractiveness (a proxy for ecological vulnerability). The results indicated that the ratio of input to revenues and how much tourism contributed to pollution were the most influential factors in socio-economic and ecological vulnerability scenarios.

**Service Section**

**Main Application Fields:** ABM is applied to many scientific fields with main applications taking place in ecology, the social sciences, land use planning, and transport logistics. In the context of tourism, climate change (human-environment interactions), visitor flow, disaster planning, and tourist/operator decision-making support are the main types of application to date.

(continued)

**Pros:** The basic concepts of ABM, agents, a spatial setting, and interactions are easy to grasp. This form of modelling incorporates complexity, interactions, heterogeneity, nonlinear feedback, and uncertainty. Moreover, ABM can include movement of individual agents, which is of interest in many tourism-related studies.

**Limitations and Pitfalls:** There are several limitations to ABM. Although the basic ideas thereof are easy, modelling human decisions and other processes can be quite complicated and conceptually deep. Complex models can mask processing and input errors as well as require extensive processing power to analysis. Moreover, ABM's predictive ability is disputed and difficult to verify and validate because of the dearth of empirical data to verify and validate model processes and outputs.

**Similar Methods and Methods to Combine with:** ABM can be combined with many modelling approaches and analysis types. For instance, GIS is a common addition in cases where specified spatial details are important. On the other hand, systems dynamics modelling is a top-down approach to modelling the system, and ABM and systems dynamics models can be used on the same research questions to compare results. Artificial intelligence could also be applied to help provide richness and depth to simulating behaviours. Lastly, big data can be used in ABM to provide input parameters, improve predictive capabilities, and for verification and validation of model processes and results.

**Code:** The NetLogo Model for the Coasting Model (Student et al., 2020a) is accessible at: <https://github.com/DataScience-in-Tourism/Chapter-23-Agent-based-Modeling>

**Acknowledgements** Thanks to Emily Liang, creative science communicator at Wageningen University & Research's Environmental Policy Group, for designing Figs. 1, 2, and 3.

## Further Readings and Other Sources

### *Agent-Based Modelling*

- Railsback, S., & Grimm, V. (2011). *Agent-based and individual-based modeling: A practical introduction*. Princeton University Press. <https://doi.org/10.2307/j.ctt7sns7>
- Van Dam, K. H., Nikolic, I., & Lukszo, Z. (Eds.). (2012). *Agent-based modelling of socio-technical systems* (Vol. 9). Springer Science & Business Media.
- Heppenstall, A., Crooks, A., See, L., & Batty, M. (2012). *Agent-based models of geographical systems*. *Journal of chemical information and modeling* (Vol. 53). Springer. <https://doi.org/10.1007/978-90-481-8927-4>

- Janssen, M.A. (2020) *Introduction to agent-based modeling: With applications to social, ecological, and social-ecological systems*. E-book. <https://intro2abm.com/>
- Macal, C. M. (2016). Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10(2), 144–156.
- Wilensky, U. (1999). *NetLogo*. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

### ***Overview, Design Concepts, and Details (ODD) + Decision-making (ODD+D) Protocols***

- Grimm, V., Railsback, S. F., Vincenot, C. E., Berger, U., Gallagher, C., Deangelis, D. L., & Ayllón, D. (2020). The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2) <https://doi.org/10.18564/jasss.4259>
- Müller, B., Bohn, F., Dreßler, G., Groeneveld, J., Klassert, C., Martin, R., Schwarz, N. (2013). Describing human decisions in agent-based models - ODD+D, an extension of the ODD protocol. *Environmental Modelling and Software*, 48, 37–48. <https://doi.org/https://doi.org/10.1016/j.envsoft.2013.06.003>

### ***Software Selection***

- Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. P. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>

### ***GIS***

- Heppenstall, A., Crooks, A., See, L., & Batty, M. (2012). Agent-based models of geographical systems. *Journal of Chemical Information and Modeling*, 53. Dordrecht: Springer. <https://doi.org/10.1007/978-90-481-8927-4>

## Analysis

- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., & Tarantola, S. (2008). *Global sensitivity analysis: The primer*. Wiley.
- Bryant, B. P., & Lempert, R. J. (2010). Thinking inside the box: A participatory, computer-assisted approach to scenario discovery. *Technological Forecasting and Social Change*, 77(1), 34–49. <https://doi.org/10.1016/j.techfore.2009.08.002>

## References

- Abar, S., Theodoropoulos, G. K., Lemariniere, P., & O'Hare, G. M. P. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33. <https://doi.org/10.1016/j.cosrev.2017.03.001>
- Abdou, M., Hamill, L., & Gilbert, N. (2012). Designing and building an agent-based model. In A. Heppenstall, A. Crooks, L. See, & M. Batty (Eds.), *Agent-based models of geographical systems* (pp. 141–165). Springer. [https://doi.org/10.1007/978-90-481-8927-4\\_8](https://doi.org/10.1007/978-90-481-8927-4_8)
- Alvarez, E., & Brida, J. G. (2019). An agent-based model of tourism destinations choice. *International Journal of Tourism Research*, 21(2), 145–155. <https://doi.org/10.1002/jtr.2248>
- Amelung, B., Student, J., Nicholls, S., Lamers, M., Baggio, R., Boavida-Portugal, I., . . . Balbi, S. (2016). The value of agent-based modelling for assessing tourism–environment interactions in the Anthropocene. *Current Opinion in Environmental Sustainability*, 23, 46–53. <https://doi.org/10.1016/j.cosust.2016.11.015>
- Assaf, A. G., Kock, F., & Tsionas, M. (2021). Tourism during and after COVID-19: An expert-informed agenda for future research. *Journal of Travel Research*, 524, 004728752110172. <https://doi.org/10.1177/00472875211017237>
- Baggio, R. (2008). Symptoms of complexity in a tourism system. *Tourism Analysis*, 13(1), 1–20. <https://doi.org/10.3727/108354208784548797>
- Baggio, R., & Sainaghi, R. (2011). Complex and chaotic tourism systems: Towards a quantitative approach. *International Journal of Contemporary Hospitality Management*, 23(6), 840–861. <https://doi.org/10.1108/09596111111153501>
- Balbi, S., & Giupponi, C. (2010). Modelling of socio-ecosystems: A methodology for the analysis of adaptation to climate change. *International Journal of Agent Technologies and Systems*, 2(4), 17–38. <https://doi.org/10.4018/jats.2010100103>
- Balbi, S., Giupponi, C., Perez, P., & Alberti, M. (2013). A spatial agent-based model for assessing strategies of adaptation to climate and tourism demand changes in an Alpine tourism destination. *Environmental Modelling and Software*, 45, 29–51. <https://doi.org/10.1016/j.envsoft.2012.10.004>
- Boas, I., Kloppenburg, S., van Leeuwen, J., & Lamers, M. (2018). Environmental mobilities: An alternative lens to global environmental governance. *Global Environmental Politics*, 18(4), 107–126. [https://doi.org/10.1162/glep\\_a\\_00482](https://doi.org/10.1162/glep_a_00482)
- Boavida-Portugal, I., Ferreira, C. C., & Rocha, J. (2017). Where to vacation? An agent-based approach to modelling tourist decision-making process. *Current Issues in Tourism*, 20(15), 1557–1574. <https://doi.org/10.1080/13683500.2015.1041880>
- Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3), 7280–7287. <https://doi.org/10.1073/pnas.082080899>
- Chhetri, P., & Arrowsmith, C. (2008). GIS-based modelling of recreational potential of nature-based tourist destinations. *Tourism Geographies*, 10(2), 233–257. <https://doi.org/10.1080/14616680802000089>

- Dignum, F., Dignum, V., Davidsson, P., Ghorbani, A., van der Hurk, M., Jensen, M., . . . Verhagen, H. (2020). Analysing the combined health, social and economic impacts of the Coronavirus pandemic using agent-based social simulation. *Minds and Machines*, 30, 177–194. <https://doi.org/10.1007/s11023-020-09527-6>
- Étienne, M., du Toit, D. R., & Pollard, S. (2011). ARDI: A co-construction method for participatory modeling in natural resources management. *Ecology and Society*, 16(1), 44. <https://doi.org/10.5751/es-03748-160144>
- Gössling, S., Scott, D., Hall, C. M., Ceron, J. P., & Dubois, G. (2012). Consumer behaviour and demand response of tourists to climate change. *Annals of Tourism Research*, 39(1), 36–58. <https://doi.org/10.1016/j.annals.2011.11.002>
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., . . . DeAngelis, D. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, 198(1–2), 115–126. <https://doi.org/10.1016/j.ecolmodel.2006.04.023>
- Grimm, V., Berger, U., DeAngelis, D., Polhill, J., Giske, J., & Railsback, S. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, 221(23), 2760–2768. <https://doi.org/10.1016/j.ecolmodel.2010.08.019>
- Grimm, V., Railsback, S. F., Vincenot, C. E., Berger, U., Gallagher, C., Deangelis, D. L., . . . Ayllón, D. (2020). The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 1. <https://doi.org/10.18564/jasss.4259>
- Hahn, H. A. (2013). The conundrum of verification and validation of social science-based models. *Procedia Computer Science*, 16, 878–887. <https://doi.org/10.1016/j.procs.2013.01.092>
- Heppenstall, A., & Crooks, A. (2019). Guest editorial for spatial agent-based models: Current practices and future trends. *GeoInformatica*, 23(2), 163–167. <https://doi.org/10.1007/s10707-019-00349-y>
- Heppenstall, A. A., Crooks, A. T., See, L. M., & Batty, M. (Eds.). (2012). *Agent-based models of geographical systems*. Springer. <https://doi.org/10.1007/978-90-481-8927-4>
- Herman, J., & Usher, W. (2017). SALib: An open-source Python library for sensitivity analysis. *Journal of Open Source Software*, 2(9), 97. <https://doi.org/10.21105/joss.00097>
- Johnson, P. A., Nicholls, S., Student, J., Amelung, B., Baggio, R., Balbi, S., . . . Steiger, R. (2016). Easing the adoption of agent-based modelling (ABM) in tourism research. *Current Issues in Tourism*, 20(8), 801–808. <https://doi.org/10.1080/13683500.2016.1209165>
- Johnson, P. A., & Sieber, R. E. (2010). An individual-based approach to modeling tourism dynamics. *Tourism Analysis*, 15(5), 517–530. <https://doi.org/10.3727/108354210X12889831783198>
- Johnson, P. A., & Sieber, R. (2011). An agent-based approach to providing tourism planning support. *Environment and Planning B: Planning and Design*, 38(3), 486–504. <https://doi.org/10.1068/b35148>
- Kwakkel, J. H., & Pruyt, E. (2013). Exploratory modeling and analysis, an approach for model-based foresight under deep uncertainty. *Technological Forecasting and Social Change*, 80(3), 419–431. <https://doi.org/10.1016/j.techfore.2012.10.005>
- Lempert, R. J. (2019). Robust decision making (RDM). In V. Marchau, W. Walker, P. Bloemen, & S. Popper (Eds.), *Decision making under deep uncertainty* (pp. 23–51). Springer. [https://doi.org/10.1007/978-3-030-05252-2\\_2](https://doi.org/10.1007/978-3-030-05252-2_2)
- Levin, S., Xepapadeas, T., Crépin, A. S., Norberg, J., De Zeeuw, A., Folke, C., . . . Walker, B. (2013). Social-ecological systems as complex adaptive systems: Modeling and policy implications. *Environment and Development Economics*, 18, 111–132. <https://doi.org/10.1017/s1355770x12000460>
- Lew, A., & McKercher, B. (2006). Modeling tourist movements: A local destination analysis. *Annals of Tourism Research*, 33(2), 403–423. <https://doi.org/10.1016/j.annals.2005.12.002>
- Li, S., Colson, V., Lejeune, P., Speybroeck, N., & Vanwambeke, S. O. (2015). Agent-based modelling of the spatial pattern of leisure visitation in forests: A case study in Wallonia,

- South Belgium. *Environmental Modelling and Software*, 71, 111–125. <https://doi.org/10.1016/j.envsoft.2015.06.001>
- Li, S., Yang, Y., Zhong, Z., & Tang, X. (2021). Agent-based modeling of spatial spillover effects in visitor flows. *Journal of Travel Research*, 60(3), 546–563. <https://doi.org/10.1177/0047287520930105>
- Ligmann-Zielinska, A., Siebers, P. O., Magliocchia, N., Parker, D., Grimm, V., Du, E. J., . . . Ye, X. (2020). ‘One size does not fit all’: A roadmap of purpose-driven mixed-method pathways for sensitivity analysis of agent-based models. *Journal of Artificial Societies and Social Simulation*, 23(1). <https://doi.org/10.18564/jasss.4201>
- Macal, C. M. (2016). Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10(2), 144–156. <https://doi.org/10.1057/jos.2016.7>
- Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, 4(3), 151–162. <https://doi.org/10.1057/jos.2010.3>
- Milano, C., Novelli, M., & Cheer, J. M. (2019). Overtourism and degrowth: A social movements perspective. *Journal of Sustainable Tourism*, 27(12), 1857–1875. <https://doi.org/10.1080/09669582.2019.1650054>
- Müller, B., Balbi, S., Buchmann, C. M., de Sousa, L., Dressler, G., Groeneveld, J., & Weise, H. (2014). Standardised and transparent model descriptions for agent-based models: Current status and prospects. *Environmental Modelling & Software*, 55, 156–163. <https://doi.org/10.1016/j.envsoft.2014.01.029>
- Müller, B., Bohn, F., Dreßler, G., Groeneveld, J., Klassert, C., Martin, R., . . . Schwarz, N. (2013). Describing human decisions in agent-based models - ODD+D, an extension of the ODD protocol. *Environmental Modelling and Software*, 48, 37–48. <https://doi.org/10.1016/j.envsoft.2013.06.003>
- Neuburger, L., & Egger, R. (2021). Travel risk perception and travel behaviour during the COVID-19 pandemic 2020: A case study of the DACH region. *Current Issues in Tourism*, 24(7), 1003–1016. <https://doi.org/10.1080/13683500.2020.1803807>
- Ngo, T. A., & See, L. (2012). Calibration and validation of agent-based models of land cover change. In A. Heppenstall, A. Crooks, L. See, & M. Batty (Eds.), *Agent-based models of geographical systems* (pp. 181–197). Springer. [https://doi.org/10.1007/978-90-481-8927-4\\_10](https://doi.org/10.1007/978-90-481-8927-4_10)
- Nicholls, S., Amelung, B., & Student, J. (2017). Agent-based modeling: A powerful tool for tourism researchers. *Journal of Travel Research*, 56(1), 3–15. <https://doi.org/10.1177/0047287515620490>
- O’Connor, A., Zenger, A., & Itami, B. (2005). Geo-temporal tracking and analysis of tourist movement. *Mathematics and Computers in Simulation*, 69(1–2), 135–150. <https://doi.org/10.1016/j.matcom.2005.02.036>
- Peeters, P., & Dubois, G. (2010). Tourism travel under climate change mitigation constraints. *Journal of Transport Geography*, 18(3), 447–457. <https://doi.org/10.1016/j.jtrangeo.2009.09.003>
- Pirotta, E., New, L., Harwood, J., & Lusseau, D. (2014). Activities, motivations and disturbance: An agent-based model of bottlenose dolphin behavioral dynamics and interactions with tourism in Doubtful Sound, New Zealand. *Ecological Modelling*, 282, 44–58. <https://doi.org/10.1016/j.ecolmodel.2014.03.009>
- Pizzitutti, F., Mena, C. F., & Walsh, S. J. (2014). Modelling tourism in the Galapagos Islands: An agent-based model approach. *Journal of Artificial Societies and Social Simulation*, 17(1), 14. <https://doi.org/10.18564/jasss.2389>
- Pons, M., Johnson, P. A., Rosas, M., & Jover, E. (2014). A georeferenced agent-based model to analyze the climate change impacts on ski tourism at a regional scale. *International Journal of Geographical Information Science*, 28(12), 2474–2494. <https://doi.org/10.1080/13658816.2014.933481>
- Pons, M., Johnson, P. A., Rosas-Casals, M., Sureda, B., & Jover, È. (2012). Modeling climate change effects on winter ski tourism in Andorra. *Climate Research*, 54(3), 197–207. <https://doi.org/10.3354/cr01117>

- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., & Tarantola, S. (2008). *Global sensitivity analysis: The primer*. Wiley. <https://doi.org/10.1002/9780470725184>
- Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity analysis in practice: A guide to assessing scientific models*. Wiley. <https://doi.org/10.1002/0470870958>
- Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology*, 1, 143–186. <https://doi.org/10.1080/0022250X.1971.9989794>
- Soboll, A., & Dinkeldey, A. (2012). The future impact of climate change on Alpine winter tourism: A high-resolution simulation system in the German and Austrian Alps. *Journal of Sustainable Tourism*, 20(1), 101–120. <https://doi.org/10.1080/09669582.2011.610895>
- Soboll, A., & Schmude, J. (2011). Simulating tourism water consumption under climate change conditions using agent-based modeling: The example of ski areas. *Annals of the Association of American Geographers*, 101(5), 1049–1066. <https://doi.org/10.1080/00045608.2011.561126>
- Steinmann, P., Atuping, W. L., & Kwakkel, J. H. (2020). Behavior-based scenario discovery using time series clustering. *Technological Forecasting and Social Change*, 156, 1–9. <https://doi.org/10.1016/j.techfore.2020.120052>
- Student, J., Amelung, B., & Lamers, M. (2016a). Towards a tipping point? Exploring the capacity to self-regulate Antarctic tourism using agent-based modelling. *Journal of Sustainable Tourism*, 24(3). <https://doi.org/10.1080/09669582.2015.1107079>
- Student, J., Amelung, B., & Lamers, M. (2016b). Vulnerability is dynamic! Conceptualising a dynamic approach to coastal tourism destinations' vulnerability. In W. L. Filho (Ed.), *Innovation in climate change adaptation* (pp. 31–42). Springer. <https://doi.org/10.1007/978-3-319-25814-0>
- Student, J., Kramer, M. R., & Steinmann, P. (2020a). Coasting: Model description, global sensitivity analysis, and scenario discovery. *MethodsX*, 7, 101145. <https://doi.org/10.1016/j.mex.2020.101145>
- Student, J., Kramer, M. R., & Steinmann, P. (2020b). Simulating emerging coastal tourism vulnerabilities: An agent-based modelling approach. *Annals of Tourism Research*, 85. <https://doi.org/10.1016/j.annals.2020.103034>
- Student, J., Lamers, M., & Amelung, B. (2020). A dynamic vulnerability approach for tourism destinations. *Journal of Sustainable Tourism*, 28(3), 475–496. <https://doi.org/10.1080/09669582.2019.1682593>
- ten Broeke, G., van Voorn, G., & Ligtenberg, A. (2016). Which sensitivity analysis method should I use for my agent-based model? *Journal of Artificial Societies and Social Simulation*, 19(1), 5. <https://doi.org/10.18564/jasss.2857>
- Troitzsch, K. G. (2014). Analysing simulation results statistically: Does significance matter? In D. Adamatti, G. Dimuro, & H. Coelho (Eds.), *Interdisciplinary applications of agent-based social simulation and modeling*. IGI Global. <https://doi.org/10.4018/978-1-4666-5954-4.ch006>
- Urry, J. (2000). *Sociology beyond societies: Mobilities for the twenty-first century*. Routledge.
- van Dam, K. H., Nikolic, I., & Lukszo, Z. (2013). *Agent-based modelling of socio-technical systems* (Vol. 9, p. Springer). <https://doi.org/10.1007/978-94-007-4933-7>
- Vinogradov, E., Leick, B., & Kivedal, B. K. (2020). An agent-based modelling approach to housing market regulations and Airbnb-induced tourism. *Tourism Management*, 77, 104004. <https://doi.org/10.1016/j.tourman.2019.104004>
- Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling.
- Yin, L. (2007). Assessing indirect spatial effects of mountain tourism development: An application of agent-based spatial modeling. *Journal of Regional Analysis and Policy*, 37, 257–265.
- Zhai, X., Zhong, D., & Luo, Q. (2019). Turn it around in crisis communication: An ABM approach. *Annals of Tourism Research*, 79, 102807. <https://doi.org/10.1016/j.annals.2019.102807>
- Zhang, Y., Gao, J., Cole, S., & Ricci, P. (2020). How the spread of user-generated contents (UGC) shapes international tourism distribution: Using agent-based modeling to inform strategic UGC marketing. *Journal of Travel Research*, 60(7), 1469–1491. <https://doi.org/10.1177/0047287520951639>

# Geographic Information System (GIS)



## Making Sense of Geospatial Data

Andrei P. Kirilenko

### Learning Objectives

- Define GIS
- Describe main functions of GIS
- Introduce data representations
- Define main concepts of spatial analysis
- Indicate popular GIS software

## 1 Introduction

The majority of data collected in the real world contain a spatial component. For instance, surveys are conducted in multiple locations, industry data reflect business environments in different cities, destinations and points of interest are spatially distributed, visitors come from diverse places, the environment changes over space, and so on. A researcher may decide to ignore this spatial component of the collected data or may represent it with a nominal variable, for example, by introducing a variable “place” to record a tourist’s visitation points. Essentially, the approach replaces the ratio level of measurement locational data (e.g., geographical coordinates of the visitation points) with nominal data (e.g., visitation point IDs). This greatly simplifies the analysis but makes considering the distance between the visitation points impossible.

---

A. P. Kirilenko (✉)

Department of Tourism, Hospitality and Event Management, University of Florida, Gainesville, United States

e-mail: [andrei.kirilenko@ufl.edu](mailto:andrei.kirilenko@ufl.edu)



Philosophically speaking, the need for spatially explicit data analysis originally ensued from the First Law of Geography, formulated by Waldo Tobler (1970), stating that “everything is related to everything else, but near things are more related than distant things” (p. 236). One research, widely cited as the first scientific application of this principle in a rigorous study, is the 1854 London cholera outbreak analysis by Dr. John Snow. In this pioneering study, Dr. Snow carefully mapped cholera incidents on a commercial-grade map of London, which allowed him to trace the source of the disease to a particular water well contaminated with graywater from a leaking sewer line (for more details see Koch, 2004). Notably, this study earned Dr. Snow recognition as the father of two sciences: Geographic Information Systems (GIS) and modern epidemiology.

Despite the early origins of the ideas that formed the foundation of GIS, practical applications of GIS had only really begun during the quantitative revolution of the 1960s, when access to mainframe computers started becoming widely available to the scientific community. The initial development of computer-based GIS can be attributed to the efforts of Roger Tomlinson and Michael Goodchild, amongst others, and their establishment of a computerized inventory of Canada’s natural resources (Goodchild, 2018). Many of those early decisions regarding GIS’ design have rarely changed over the past half-century and, notably, that includes the organization of data centered around a map (Goodchild, 2018).

The Geographic Information System (GIS) is a set of methods and technologies for capturing, storing, analyzing, and visualizing spatially explicit data. The wide scope of these tasks is reflected in three distinct views of GIS, namely, database-centric, spatial analysis-centric, and map-centric (Maguire, 1991). The core of the GIS is a geodatabase, which provides functionality for the storage and management of spatially explicit data. Hence, the database-centric view accentuates the development of an efficient and reliable system to store, update, and query the data with a locational component. Meanwhile, the spatial analysis-centric view stresses GIS’ ability to extend traditional methods of statistics to spatial data, introducing methods such as spatial descriptive statistics, spatial regression, spatial autocorrelation, and many others. Finally, the map-centric view places an emphasis on the data visualization and map-making functions of GIS. For the time being, a GIS analysis includes elements from all three views; that is, it starts with the collection and organization of spatial data, continues with spatial data analysis, and, lastly, visualizes the results with a series of maps. The next section will provide details on these three aspects of GIS together with sample methods and procedures.

## 2 Theoretical Foundations

GIS includes important concepts and methods from data management, visualization, statistics and data mining, collaboration and dissemination, decision-making support, and many other areas. In our selection of topics for this section, we limited

ourselves to choosing a small subset of methods that we deem relevant for starting academic research in the fields of tourism and hospitality.

## 2.1 Data

Non-spatial data lacks spatial characteristics, while spatial data is characterized by location, size, and shape. This trivial observation results in a fundamental difference between non-spatial and spatial data in terms of storage, processing, and representation. When it comes to storage, the standard table-oriented storage used for non-spatial data is inadequate for spatial data as it does not allow the representation of linkages between stored objects. Instead, a spatial database optimized for the representation of points, lines, polygons, and other spatial structures is used. In addition to the common database queries, the spatial database permits geographic queries, spatial overlay of the datasets, computation of the distances and areas, spatial modification of the stored objects (e.g., making a buffer around a river), prediction of a spatial relationship between objects (e.g., is the intended development outside a 100 m buffer of a body), and many other means of processing that are impossible for non-spatial data.

The two types of spatial data an academic researcher usually deals with are vectors and rasters, which conceptualize the world in a discreet or continuous way, respectively. For example, lakes, rivers, or boat ramps, or, in other words, objects with clearly defined boundaries, are better represented in a vector format, more specifically, as polygons (also called area), lines (also called arc), and point features, respectively. Contrarily, naturally continuous data, such as temperature distribution, is better represented as a raster using a continuous grid of square cells, for example,  $100 \times 100$  km or  $1 \times 1$  degree in latitude and longitude. Note that there are several modifications and many formats relating to the vector and raster data models. Yet, a researcher's preferred data model and format should depend not only on the object of interest but also on the way data was collected (e.g., remote sensing data would be raster), the intended analysis (some methods assume vector data model while others require rasters), and the capabilities of a specific GIS.

Raster data is usually represented as a regular grid. For example, the representation of the mean temperature of Florida (in terms of land) in the ArcGIS ASCII raster format could be organized in the following simple way (Table 1 (A)): the number of columns and rows; the coordinates of the lower-left center of the grid, the size of each cell, and the value used for missing data. Vector data, on the other hand, has two components, geometry and attributes. Geometry defines the shape of the objects, and attributes describe the objects' features. Using the previous example, geometry could describe the geographical locations of meteorological stations across the state of Florida, while features could include temperature and precipitation measurements at these locations. Hence, the easiest way to add a spatial component to collected data is by adding geographical coordinates to each data collection point (Table 1 (B)).

**Table 1** Example of temperature representation for Florida using raster and vector data formats. Format examples denote ASCII raster format of ArcGIS and point .dat format of GRASS GIS. The XLLCORNER and YLLCORNER rows denote the longitude and latitude of the lower left corner of the grid while the CELLSIZE denotes the size of one grid cell (1 geographical degree in this example)

A. Raster format	B. Vector format (point data)
NCOLS 8	ID Latitude Longitude Name Temp.
NROWS 7	1 +25.788 -080.317 Miami 24.0
XLLCORNER -87.55	2 +26.378 -080.108 Boca Raton 22.5
YLLCORNER 24.78	3 +28.290 -081.437 Kissimmee 21.5
CELLSIZE 1	..... (more points) .....
NODATA_VALUE -999	
19 19.5 19 20 19 18.5 18 -999	
-999 -999 -999 -999 20 19.5 19 20	
..... (5 more rows).....	

## 2.2 Analysis

Spatial data are usually autocorrelated as follows from the First Law of Geography (Tobler, 1970), whereas non-spatial data are frequently independent. This implies a difference in methods of analysis. Thus, in this section, some frequently used exploratory and inferential methods typical for GIS-based research will be introduced.

**Spatial Autocorrelation** Similar to “normal” autocorrelation, which gauges associations between measurements taken over successive time intervals, spatial autocorrelation refers to the association between the measurements taken over increasing spatial intervals. As Cliff and Ord (1973) define, “If the presence of some quantity in a county (sampling unit) makes its presence in neighboring counties (sampling units) more or less likely, we say that the phenomenon exhibits spatial autocorrelation” (p. 1). For interval and ordinal data, the spatial autocorrelation is usually measured with Moran’s I, which is a weighted correlation with weights representing spatial distances. Moran’s I varies from  $-1$  to  $+1$  with zero representing the absence of autocorrelation. Note that Moran’s I as well as other measures of spatial autocorrelation should be used as an inferential statistic; that means, prior to concluding that data tend to be spatially clustered, it should be tested against the null hypothesis (that there is no spatial clustering).

Moran’s I measures autocorrelation across an entire area of interest. In practice, the spatial distribution of data exhibits patches or clusters of data that tend to be similar. Numerically, this concept is represented by the Local Indicators of Spatial Association (LISA) (see Anselin, 1995). For example, Sarrión-Gavilán et al. (2015) researched the changes in the number of beds in Andalusian hotels over time. The authors found a statistically significant positive Moran’s I, which increased over time, indicating an increasing tendency of the hospitality industry to cluster together.

Furthermore, they used *local* Moran's I LISA statistics to map the specific locations with high autocorrelation pointing to developing clusters. Another useful LISA statistics is Gettis-Ord  $G_i^*$ , also known as the "hot spot analysis." While local Moran's I indicates where similar values tend to spatially co-occur, Gettis-Ord  $G_i^*$  reveals where high or low values are concentrated. For example, Van der Zee et al. (2020) used Gettis-Ord  $G_i^*$  to find clusters of frequently ("hot spot") and infrequently ("cold spot") reviewed restaurants in a city.

**Spatial Interpolation** While measurements are frequently associated with fixed locations (e.g., temperature is measured at meteorological stations), one might assume that a measured phenomenon changes gradually, which exhibits positive autocorrelation. Spatial interpolation aims to predict such missing values (e.g., of the temperature) across the entire area of interest or to downscale the data. Two interpolation methods, namely, the Inverse Distance Weighted (IDW) and Kriging, are the most frequently used. The idea of the IDW is to compute the distances from the point of interest to  $N$  nearest neighbors and to use the inverse distances to compute the weighted mean of the measurements. Kriging, in contrast, uses a statistical approach. The Ordinary Kriging works as follows: first, the spatial trend in measurements is removed, and then the squared differences between the measured data are observed. If the phenomenon of interest is indeed spatially autocorrelated, these differences generally increase with increased distance between the locations of the sample sites. The final step is to describe this observation (called experimental semi-variogram) mathematically by finding the best-fitting model. A modification known as CoKriging is a multivariate extension of the Ordinary Kriging. With this, for example, property prices can be predicted based on a sample of recent sales together with auxiliary information such as the floor area, age of the property, socioeconomic and sociodemographic characteristics of the neighborhood, etc. (Kuntz & Helbich, 2014).

**Spatial Regression** The goal of spatial regression analysis is to explain or predict the values of the variable of interest based on the values of independent variables *and* their spatial distribution. As an example, Su et al. (2020) used a spatial regression model to explain the country-wide differences in public reactions relating to a highly publicized hotel service incident in Beijing, China. The authors found that factors explaining the public's interest vary throughout the country; for instance, in the geographical location closest to the incident northwest of China, the most critical factor was hotel chain presence. Meanwhile, in the coastal area in the southeast of China, the main factor was the administrative status of the city.

Let us consider a "regular" linear regression model:  $y_i = \sum_j b_j x_{ij} + \varepsilon_i$  where  $y = (y_i)$  is a vector of dependent variables,  $X = (x_{ij})$  is a matrix of independent variables,  $b = (b_j)$  is a vector of regression coefficients, and  $\varepsilon_i = (\varepsilon_i)$  is a random vector. The Ordinary Least Squares (OLS) estimator then gives the value of the unknown vector  $b = (X^T X)^{-1} X^T y$ . Note that the coefficients  $b$  do not vary in space, which may appear unrealistic as the relationship between the dependent and independent variables may change in different locations. One way of dealing with this

problem is by introducing a dummy variable that represents different locations; this may improve model fitting but still leaves the relationships fixed in space. A more productive approach is to allow the coefficients  $b$  to vary in space:  $B_k = b_j(p_k)$ , with  $p_k$  designating a geographical location. A popular technique used to estimate the coefficients  $B_k$  for each location is the Geographically Weighted Regression (GWR) method (Brunsdon et al., 1998). The main idea of GWR is to run a “regular” OLS model for each spatial location  $k$ , extending the matrix  $X$  by including the data of neighboring locations. The data from nearby locations should thus have a higher influence on the model than the data from remote locations; in regards to the latter, GWR uses a weight matrix  $W$ . The obtained coefficients present an estimate of the matrix of coefficients and can be simplified as  $B_k = (X^T W_k X)^{-1} X^T W_k Y$ . The use of GWR is finally justified by testing it against the “regular” OLS estimator with  $H_0: B_1 = \dots = B_K$ .

### 2.3 Creating Maps

**Map Elements** The two most common map elements include the data frame and legend (Peterson, 2020). The data frame is a geographic window in which the map layers are displayed using the same scale and projection. A map may have multiple data frames; for example, a small data frame may be used to help the reader understand the general location of the area of interest on a global map. Regarding legends, they depict the meaning of the symbols and colors used on a map. Four other map elements, namely, map title, scale, north arrow, and copyrights, are less frequent in academic publications as the title is usually viewed as redundant due to the presence of figure captions, and other elements may be implied.

**Projection** Inevitably, maps present a distorted view of the world through its projection of the spherical globe onto a flat surface. As such, the following distortions may be present on a map: distance, area, direction, and shape (Snyder, 1997). The most widely used Mercator projection, for example, greatly inflates the areas towards the poles so that the area of Greenland looks equivalent to that of Africa while, in fact, it is actually 14 times smaller. There are recommended projections for any specific area of the world, allowing for smaller overall distortions in its representations. For instance, the US State Plane Coordinate System (Stem, 1989) divides the conterminous US into 120 zones, for the majority of which either a transverse Mercator or a Lambert conformal conic projection is used. Additionally, the ArcGIS manual<sup>1</sup> provides a good overview of various projections. A potential problem may arise when map features use multiple projections, but a GIS software may be able to automatically reorganize the data into a single uniform projection.

---

<sup>1</sup><https://desktop.arcgis.com/en/arcmap/10.3/guide-books/map-projections>

**Feature Representation** The vector and raster data models are useful for making decisions on data collection and storage. From a researcher's perspective, however, the object and field representations of the world are more beneficial. The object paradigm treats the world as a collection of objects (e.g., points of interest, roads, residential places, and so on), whereas the field paradigm imagines the world as a continuous field of scalar values. For example, land use/land cover types can be represented as a field of numbers that describe land utilization at each point within the study area. Internally, objects and fields are better stored as vectors and rasters, respectively.

There are many ways to show data on a map, and the main objective is to make the map understandable for the target audience. The investigator can use color variations, which, in turn, can be perceived via three dimensions: hue ("color name"), value ("lightness"), and vividness (saturation). Other possibilities include texture, orientation, shape, arrangement, focus, and size (MacEachren et al., 1994). However, the seemingly infinite number of ways to present data on a map may mislead (sometimes intentionally) potential readers. A classic on this topic is "How to Lie with Maps" by Monmonier (2018), first published in 1991, and is recommended for developing an awareness of the topic.

**Pitfalls to Avoid** Presumably, the most frequent fallacy when it comes to data representation on maps is the *Modifiable Aerial Unit Problem (MAUP)*. The MAUP arises from data aggregation, for example, aggregation of point-based crime location data onto a polygon-based map of crime distribution within a city. Different ways of splitting a city area into polygons (using census blocks, census tracts, zip codes, etc.) will result in a radically different representation of crime distribution, misleading the users of the map. The problem arising from MAUP is not limited to visualization, and the aggregated data may change the statistical relationship between variables. For instance, in the 2000, 2004, and 2008 US Presidential Elections, the wealthiest Americans had a tendency to vote Republican. Yet, when aggregated on the state level, the wealthiest state tended to vote Democrat (Gelman, 2009), resulting in a completely different association between wealth and political preferences. This type of error is known as *ecological fallacy*.

Another problem arises when visualized data represents rates. The visualization of a phenomenon of interest as a value per area or a value per person can result in entirely different map representations. Moreover, when the base value is small, the rate may become unstable. For example, when mapping cancer cases per person on a county basis, the unstable rate problem arises in counties with a small population. Due to chance alone, some of those counties will have no cases during the observed period, resulting in zero cases per person. Meanwhile, other sparsely populated counties may have several cases, resulting in a very high case rate.

**Best Practices** What seems to be the most common source of producing poorly interpretable maps is using the default settings of GIS software, which does not target the intended readers. Typically, this results in overcrowded maps with non-essential elements such as the north arrow and scale. Furthermore, text should be used sparingly as essential information can be shifted to the map legend, to which

the map legend should have meaningful category breaks. Color selection should also be intuitive for a reader. Thus, using more than 12 colors is discouraged as it makes it difficult to identify and distinguish between the colors on a map. In addition, one should refrain from using colors that are indistinguishable to individuals with color blindness or reproduced by identical shades of gray when printed in black and white.<sup>2</sup> The guides provided by Peterson (2020) and Brewer (2015) give an excellent introduction to effective map designing.

The following section will present a case study demonstrating how these guidelines can be realized in practice.

### 3 Practical Demonstration

In this demonstration, we refer to the ESRI ArcGIS software, which has become a de facto standard GIS instructional software in the geography department. The following research example is based on the project “Economic and Social Value of Tourism Industry for Florida Communities: Objective Measures and Local Perceptions” funded by Eric Friedheim Tourism Institute at the University of Florida. The goal of the project was to learn how Florida residents perceive the benefits that the tourism industry brings to their local communities. The study comprehensively described the objective economic and social benefits tourism provides to Florida (supported by statistical measures), including such indicators as unemployment rate, crime level, and promotion of inclusivity and diversity within the tourism industry, and contrasted them with the subjective perceptions of local communities (supported by survey data). The study researched four distinct major areas of tourism in Florida: Orlando, Miami, the Florida Panhandle, and St. Augustine. In this section, we will illustrate how GIS can be used to describe the study area and collected data.

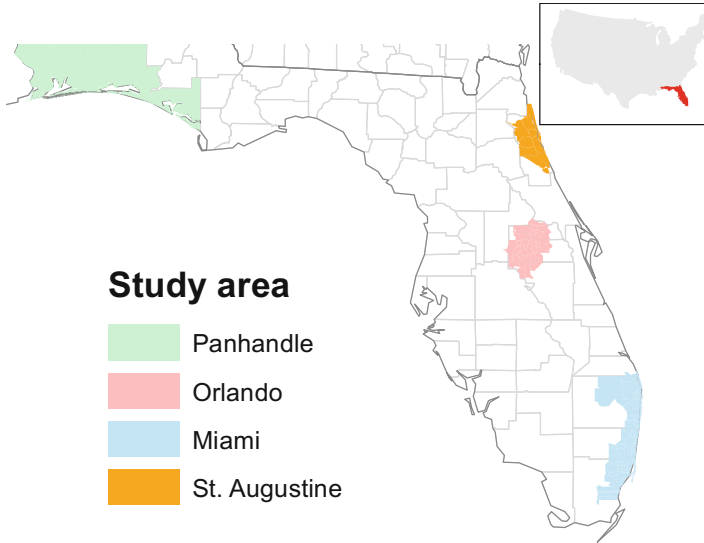
We start with a general description of the study area (Fig. 1). Notice that only two elements are present on the map: data frames and legend. A small inset map helps to find the location of the study area on the country map. Usually, the inset should be placed in the lower left-hand part of the map, however, in this particular case, that would increase the amount of white space. Note that two data frames are using different projections to reduce overall distortion.

All data, including the survey responses, should be combined in a geodatabase to be able to fully utilize GIS benefits. GIS software allows data import in many different formats. Regarding this project, the data was initially stored in Microsoft Excel format and then directly imported to ESRI ArcMap v. 10.4.

We added a spatial component to the survey data by implementing a survey question in which respondents were asked for their zip codes. Hence, we added respondents’ location using the database’s “join” operation to merge the survey table

---

<sup>2</sup>This resource <https://colorbrewer2.org/> can help in the selection of colorblind-safe and print-friendly colors.



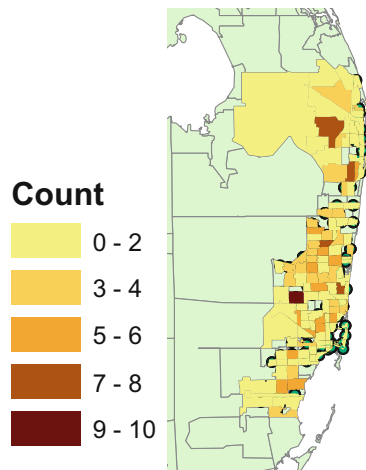
**Fig. 1** Study areas in the state of Florida

to the zip code layer. Notice that privacy considerations require a generalization of the survey's data prior to showing respondents' locations on a map.

For detailed data investigation, we can zoom into the Miami urban area. The geographical distribution of survey respondents is shown in Fig. 2. Note that this data has vector format (polygons), and also notice the change in map projection when comparing Fig. 1 to the one recommended for South Florida.

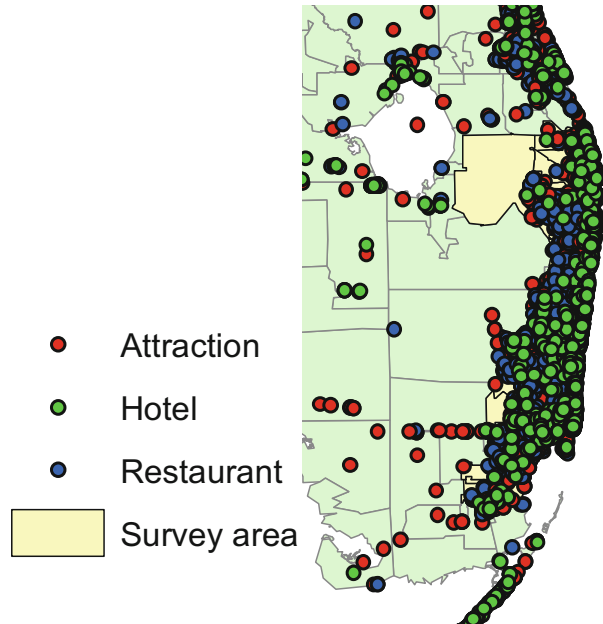
To understand the tourism, hospitality, and hotel industries' presence in reference to the study area, we used scraped TripAdvisor review data. As such, the review

**Fig. 2** Distribution of survey respondents in the Miami area





**Fig. 3** Tourism, hospitality, and hotel industries' presence in the study area

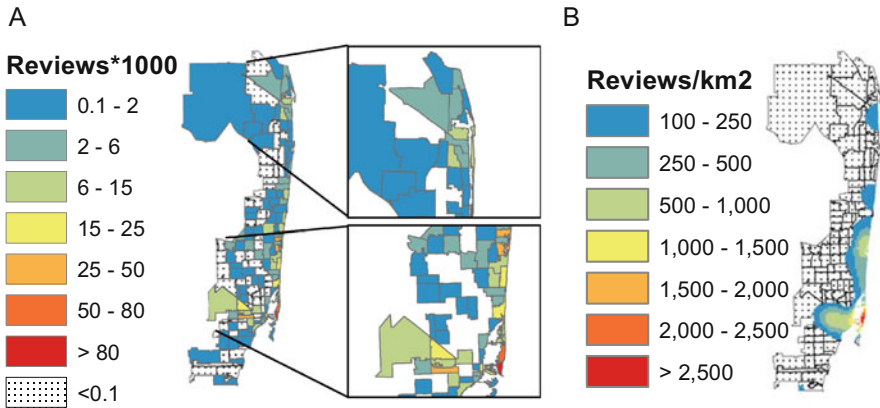


feature (Fig. 3) shows a considerable presence of tourism-related businesses in the area. Note that this data also has vector format (points).

In this example, we will concentrate on the hospitality industry. Therefore, to better understand hotel distribution, we can aggregate review data on a zip code basis. An easy way of achieving this is through the spatial join function; that is, we link generalized hotel review data to each zip code in the area. Spatial join matches the rows of two features based on their location. When there are multiple matching rows of the target feature (i.e., when a zip code contains multiple hotels), the target features can be aggregated using merging rules such as sum, mean, count, standard deviation, and others. Since we are interested in the overall tourism presence in the area, we used the “sum” merging rule (Fig. 4a). There are high numbers of hotel reviews in two areas, which are not well visible on the overall map. To show these areas using a higher resolution, we provide insets (right part of Fig. 4b).

Instead of aggregating tourist reviews based on zip codes (Fig. 4a), we can also use interpolation (Fig. 4b). Here, we used the kernel density tool, which is appropriate for individual point location (Bailey & Gatrell, 1995; Bowman & Azzalini, 1997). Note the differences in the hotel review density units; while Fig. 4a shows the number of reviews per zip code, Fig. 4b shows the number of reviews per  $\text{km}^2$ . The data format also differs between figures as the reviews per zip code are stored in vector format but the reviews per  $\text{km}^2$  is a raster. The differences in the type of data format explain the differences in spatial analysis tools that can be used for data processing.

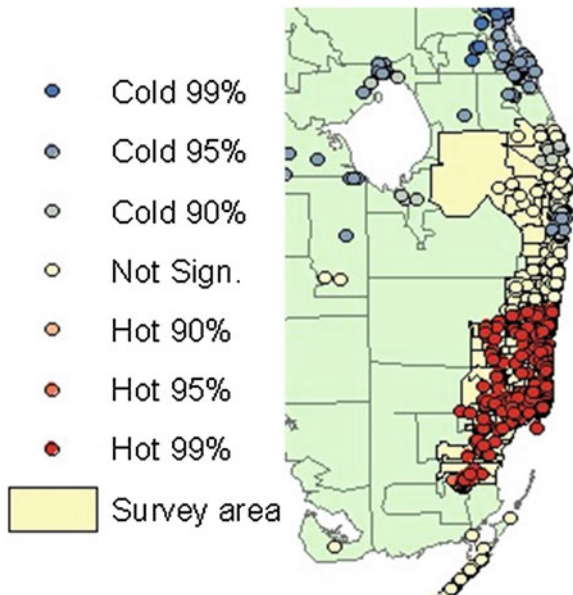
While the visualization of hotel density helps to understand the tourism presence distribution, the same density could result from the presence of few large hotels,



**Fig. 4** Hotel industry’s presence in the study area. (a) The number of TripAdvisor hotel reviews per zip code. (b) The number of TripAdvisor hotel reviews per km<sup>2</sup>

multiple small hotels, or a mix of small and large hotels. To understand which area of the city contains hotels that consistently generate large numbers of tourist reviews, we used the hot spot analysis (Getis-Ord  $G_i^*$ ). In this way, the number of hotel reviews is used as a proxy for the hotel’s popularity amongst tourists (Fig. 5). Notice a large hot spot south of the city, which includes Downtown Miami, Miami Beach, and the Hollywood coastal areas as well as locations close to Miami International Airport. The hotels in this area consistently generate many reviews, indicating a

**Fig. 5** Hot and cold spots of the hospitality industry in the Miami area. The hot spots designate co-locations of large hotels and the cold spots designate co-locations of small hotels, while not significant points relate to a mix of small and large hotels. Moreover, the percentages designate the confidence level for the hot and cold spots



large presence of tourists. On the contrary, a cold spot in the northern part of the study area depicts a consistent presence of small hotels. Thus, the opinions of local residents in these two areas may contrast.

One of the survey questions asked respondents about their opinion regarding the number of tourists coming to their area: Would they want to have a lower, same, or a higher number of tourists? Does the level of the hospitality industry's presence affect their answer? The following paragraphs demonstrate how to approach this question using GWR analysis.

The first step of GWR is accomplished with a “regular” OLS regression. The model fits the locals' satisfaction with the level of tourists' presence in the area to socio-demographic variables and the presence of the tourism industry (Fig. 4a). The OLS diagnostics then reveal that the model explains approximately 11% of variation in two dependent variables, the age of respondents and tourism industry presence ( $R^2 = 0.104$ ; adj.  $R^2 = 0.118$ ), with hospitality industry presence associated with locals wanting a decrease in the number of tourists. The older respondents are less likely to appeal to a smaller number of tourists. In addition, the Variance Inflation Factor ( $VIF = 1$  for both independent variables indicates an absence of model redundancy ( $VIF < 7.5$ ). For a brief walkthrough on other essential indicators, refer to the ESRI publication by Rosenshein et al. (n.d.).

In the second step of the analysis, the GWR model is compared with the OLS model. For GWR, the unbiased estimator of the model fit is 0.17 ( $R^2 = 0.234$ ; adj.  $R^2 = 0.170$ ), improving the OLS model fit. The Akaike Information Criterion (AIC) is another key diagnostic used to compare two models; it has generally been agreed upon that a difference of at least three units is required to accept a more complex model. In this case, the OLS'  $AIC = 350.7$  while the GWR's  $AIC = 344.8$ , supporting the selection of GWR over OLS. Hence, 17% of variation in the respondent's opinion on the number of tourists in their area is explained by the age of respondents and tourism industry presence.

### Service Section

**Main Application Fields:** One should consider using GIS whenever the spatial distribution of data is vital for understanding the investigated phenomenon. First, GIS provides a specialized way of storing and querying spatial data. Second, the geoprocessing tools provide a set of methods for data analysis, for example, spatial statistics. Third, data visualization provides the readers with effective communication of the study area, the collected data, and the outcomes thereof.

**Limitations and Pitfalls:** It is very easy to mislead the reader with unprofessionally selected data analysis and visualization. The most common is the Modifiable Aerial Unit Problem (MAUP), which arises from incorrect spatial aggregation of the data.

**Similar Methods and Methods to Combine with:** GIS methods expand on “regular” statistical methods and should be administered after a traditional statistical analysis has been applied to the data.

## Further Readings and Other Sources

### *Books*

- Bearman, N. (2020). *GIS: Research methods*. Bloomsbury Publishing A simple non-technical introduction to GIS for social scientists.
- Bowman, A. W., & Azzalini, A. (1997). *Applied smoothing techniques for data analysis: The kernel approach with S-plus illustrations* (Vol. 18). Oxford University Press.
- Cliff, A. D., & Ord, J. K. (1973). *Spatial autocorrelation*. Pion.
- Lovelace, R., Nowosad, J., & Muenchow, J. (2019). *Geocomputation with R*. CRC Press. <https://geocompr.robinlovelace.net> – free e-book suitable for those who already know R and want to expand their expertise with GIS methods.
- Margai, F., & Oyana, T. J. (2015). *Spatial analysis: Statistics, visualization, and computation methods: An overview of spatial analysis methods*. CRC Press.
- Peterson, G. N. (2020). *GIS cartography: A guide to effective map design*. CRC Press.

### *Websites*

- ColorBrewer. Online resource helping in selection of effective map color schemes. <https://colorbrewer2.org/>
- GIS lounge. A collection of GIS texts, data, and case studies. <https://www.gislounge.com/>
- GIS resources. A collection of GIS texts, data, and case studies. <https://www.gisresources.com/>
- ESRI product tutorials: <https://desktop.arcgis.com/en/arcmap/latest/get-started/main/get-started-with-arcmap.htm>

### *Tourism Applications*

- Jovanović, V., & Njeguš, A. (2008). The application of GIS and its components in tourism. *Yugoslav Journal of Operations Research*, 18(2), 261–272 An introduction to GIS application in tourism.
- Wei, W. (2012). Research on the application of geographic information system in tourism management. *Procedia Environmental Sciences*, 12, 1104–1109 An essay on GIS applications in tourism.

## References

- Anselin, L. (1995). The local indicators of spatial association LISA. *Geographical Analysis*, 27, 93–115.
- Bailey, T. C., & Gatrell, A. C. (1995). *Interactive spatial data analysis* (Vol. 413). Longman Scientific & Technical.
- Bowman, A. W., & Azzalini, A. (1997). *Applied smoothing techniques for data analysis: The kernel approach with S-Plus illustrations* (Vol. 18). OUP Oxford.
- Brewer, C. A. (2015). *Designing better maps: A guide for GIS users*. ESRI Press.
- Brunsdon, C., Fotheringham, S., & Charlton, M. (1998). Geographically weighted regression. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(3), 431–443.
- Cliff, A., & Ord, J. K. (1973). *Spatial autocorrection*. London: Pion.
- Gelman, A. (2009). *Red state, blue state, rich state, poor state: Why Americans vote the way they do*. Princeton University Press.
- Goodchild, M. F. (2018). Reimagining the history of GIS. *Annals of GIS*, 24(1), 1–8.
- Koch, T. (2004). The map as intent: Variations on the theme of John Snow. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(4), 1–14.
- Kuntz, M., & Helbich, M. (2014). Geostatistical mapping of real estate prices: An empirical comparison of kriging and cokriging. *International Journal of Geographical Information Science*, 28(9), 1904–1921.
- MacEachren, A., Bishop, I., Dykes, J., Dorling, D., & Gatrell, A. (1994). Introduction to advances in visualizing spatial data. In H. M. Hearnshaw & D. J. Unwin (Eds.), *Visualization in geographical information systems* (pp. 51–59). Wiley.
- Maguire, D. J. (1991). An overview and definition of GIS. *Geographical information systems: Principles and applications*, 1, 9–20.
- Monmonier, M. (2018). *How to lie with maps*. University of Chicago Press.
- Peterson, G. N. (2020). *GIS cartography: A guide to effective map design*. CRC Press.
- Rosenshein L., Scott, L., Pratt, M. (n.d.). *Finding a meaningful model*. ESRI. <https://www.esri.com/news/arcuser/0111/files/findmodel.pdf>
- Sarrión-Gavilán, M. D., Benítez-Márquez, M. D., & Mora-Rangel, E. O. (2015). Spatial distribution of tourism supply in Andalusia. *Tourism Management Perspectives*, 15, 29–45.
- Snyder, J. P. (1997). *Flattening the earth: Two thousand years of map projections*. University of Chicago Press.
- Stem, J. E. (1989). *State plane coordinate system of 1983* (Vol. 5). US Department of Commerce, National Oceanic and Atmospheric Administration, National Ocean Service, Charting and Geodetic Services. [https://www.ngs.noaa.gov/PUBS\\_LIB/ManualNOSNGS5.pdf](https://www.ngs.noaa.gov/PUBS_LIB/ManualNOSNGS5.pdf)
- Su, L., Kirilenko, A. P., & Stepchenkova, S. (2020). The effect of geographical and personal proximity on online discussions of service failure incidents. *Current Issues in Tourism*, 23(18), 2230–2234.
- Tobler, W. (1970). A computer movie simulating urban growth in the Detroit region. *Economic Geography*, 46, 234–240.
- Van der Zee, E., Bertocchi, D., & Vanneste, D. (2020). Distribution of tourists within urban heritage destinations: A hot spot/cold spot analysis of TripAdvisor data as support for destination management. *Current Issues in Tourism*, 23(2), 175–196.

# Visual Data Analysis



## Why, When, and How to Apply Data Visualization Techniques in the Data Analysis Process

Johanna Schmidt

### Learning Objectives

- Explain the basics of data visualization
- Illustrate the five stages of the data analysis process
- Demonstrate how data visualization can be used in every stage
- Provide examples for literature, studies, and libraries

## 1 Introduction and Theoretical Foundations

Data visualization is an interdisciplinary field that deals with the computer-supported graphical representation of data (Ware, 2019). Data visualization is highly relevant as an efficient and effective means of communication and continues to be increasingly applied in numerous domains (e.g., media, science, and marketing) for very different purposes (e.g., communication, reporting, analysis, and exploration). The reason why data visualization has become so successful is due to the fact that visual impressions strongly affect humans—in fact, vision is our dominant sense (Artal, 2016). As such, our visual system enables us to detect and interpret visual patterns rapidly and efficiently. In a physical sense, human vision can be seen as a fast and high-bandwidth information-processing channel (Cohen et al., 2016), and even ancient cultures, dating back more than 40,000 years, already used visual representations to facilitate communication. Today, visual signs and code virtually pervade and structure all realms of work and everyday life.

---

J. Schmidt (✉)

VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH, Vienna, Austria

e-mail: [johanna.schmidt@vrvis.at](mailto:johanna.schmidt@vrvis.at)

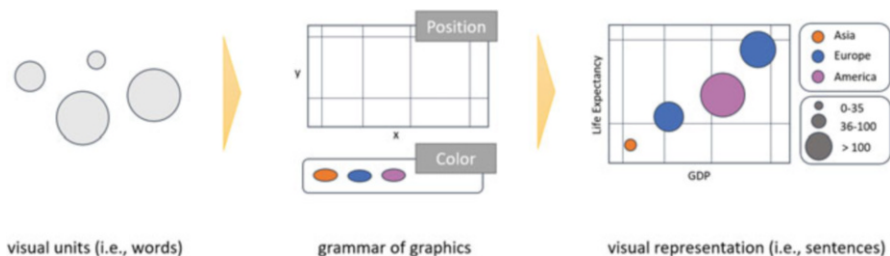
## 1.1 Data Visualization Techniques

The main idea of data visualization is to encode data (numeric and/or categorical information) as visual elements. In this sense, data visualization follows the structure of the so-called *grammar of graphics* (Wilkinson, 2005), which can be seen as a foundation for producing almost every quantitative graph or plot. The term *grammar* was chosen on purpose as the grammar of graphics is very similar to linguistic grammar. It describes how basic visual units (e.g., lines, circles, and squares) can be set in relation to each other (e.g., by position or alignment) and can be adapted with additional attributes (e.g., colors and sizes) to form a valid and comprehensible visual representation. Concerning linguistics, the basic visual units can be seen as words, and the resulting visual representations as a sentence, a text, or even a discourse (Fig. 1). The rules of graphical grammar ensure the validity and comprehensibility of the visual representation; hence, the grammar of graphics represents a framework to describe and construct data visualizations in a structured manner.

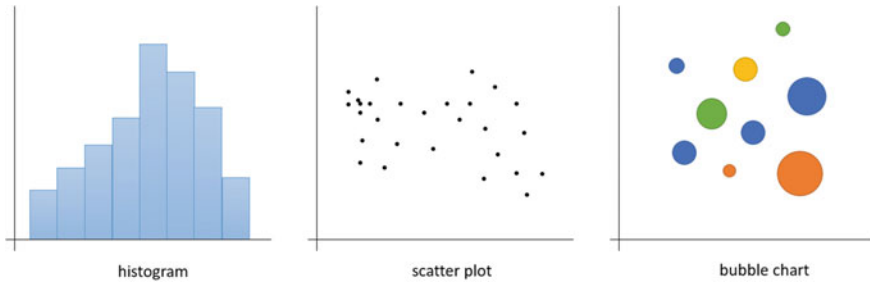
Based on the number of encodings used, different attributes of the data can be converted. For example, color can be mapped to one attribute in the data, and the same can be applied to size. The position of visual elements in a 2-dimensional space also encodes two attributes ( $x$ - and  $y$ -axis). Thus, depending on the encodings used in the data visualization, a different number of attributes can be encoded and the data can be classified into one of the following classes:

- **Univariate data:** Contains exactly one attribute
- **Bivariate data:** Contains two attributes
- **Multivariate data:** Contains more than two attributes

An example for a univariate dataset would be, e.g., a list of weights of cars that are currently on the market. An example for a bivariate dataset would be, e.g., a list of the weight and engine power of the cars currently on the market. In the case of a multivariate dataset, one could think of, e.g., a list of all possible technical attributes (weight, engine power, fuel consumption, maximum speed, engine displacement, age, etc.) belonging to the cars currently on the market. Unsurprisingly, nowadays,



**Fig. 1** Grammar of graphics. The basic elements of a data visualization are visual units (corresponding to words). The grammar of graphics then describes how these elements can be set in relation to each other and enriched with color to construct a valid visual representation (corresponding to a sentence, a text, or even a discourse)



**Fig. 2** Different types of encodings. Univariate data is often represented as a histogram, while bivariate data elements can be positioned within a scatterplot according to the values of their two attributes. For multivariate data, additional attributes can be included in the visualization by using size and color

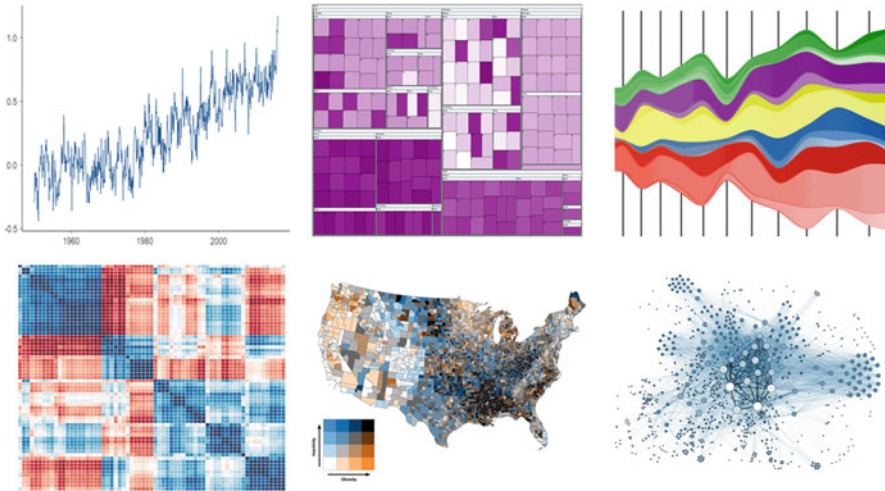
primarily due to datasets becoming more extensive and more complex, multivariate datasets are to be found more often than the other two.

Different types of data also require different ways of encoding the data attributes as visual elements (Fig. 2). In the case of univariate data, histograms are commonly used since these representations allow the viewer to analyze the statistical distribution of the data alongside outliers. In histograms, the data elements are mapped to rectangles. For bivariate data, scatterplots in which the data elements are positioned according to the values of their two attributes are commonly used. Lastly, for multivariate data, additional visual encodings need to be introduced. As such, in a bubble chart, circles represent the data elements, and by adjusting the size of the circles and their colors, it is possible to expand upon the scatterplot representation with two additional attributes.

Data visualization is an ongoing and, currently, very significant research field. Here, researchers are continuously searching for new ways to encode data as visual elements, and, therefore, the current catalog of available data visualization techniques does not end with scatterplots or bubble charts. Within the last years, research in data visualization has led to an even richer and more diverse landscape of different techniques, systems, frameworks, and applications (Post et al., 2003). Current surveys show a large variety of visualization techniques, where over 80 survey papers describing relevant state-of-the-art techniques could be identified (McNabb & Laramee, 2017). Similarly, a recent survey of books in information visualization revealed a large quantity and variety of available information (Rees & Laramee, 2019). A selection of such possible visualization techniques can be seen in Fig. 3. Moreover, different techniques have been developed depending on the semantical meaning of data, e.g., network and graph data. If data contains spatial information, this will also be represented in the visualization (e.g., by using maps). In case temporal information is essential, data can be represented along a timeline.

Nowadays, since multivariate datasets appear more often in data analysis, more data and data attributes cause a cluttered representation of the data visualization, as





**Fig. 3** Different visualization techniques. Data visualization research has led to a large catalogue of visualization techniques that can be used for numeric data, hierarchical data, temporal data, correlations, spatial data, and networks and graphs. Sources—Top left to bottom right: Esprabens et al. (2020), Soares et al. (2020), Cuenca et al. (2018), Pripdeevech et al. (2018), Strode et al. (2020), and Grandjean (2014)

can be seen in the examples in Fig. 3 (especially the network example). It is, in many cases, not possible to represent all these necessary data attributes in one single visualization. Thus, data visualization research proposes several different approaches (Munzner, 2014) to deal with this issue:

- **Filtering:** Remove data items from the visualization to reduce complexity.
- **Aggregation:** A group of elements can be represented by a newly derived element (e.g., computing averages).
- **Multiple Views:** In a multiple view approach, different visualizations represent different parts of the data (e.g., in a dashboard).
- **Interaction:** Users are able to manipulate the data visualization in order to explore different aspects.

When it comes to interaction, this is considered an important concept to keep in the loop during data analysis. Two essential principles when applying interaction to data visualization involve *Overview-First-Details-on-Demand* and *Focus + Context*. Both concepts address how users can potentially get lost while operating a large and complex dataset. To comply with the *Overview-First-Details-on-Demand* principle, it is crucial to first provide an overview of the complete dataset to the users before they can start interacting with the data visualization to get more details on demand. While interacting, it is then essential to comply with the *Focus+Context* principle. Here, visualization designers need to ensure that users do not lose the full context while interacting with the data visualization.

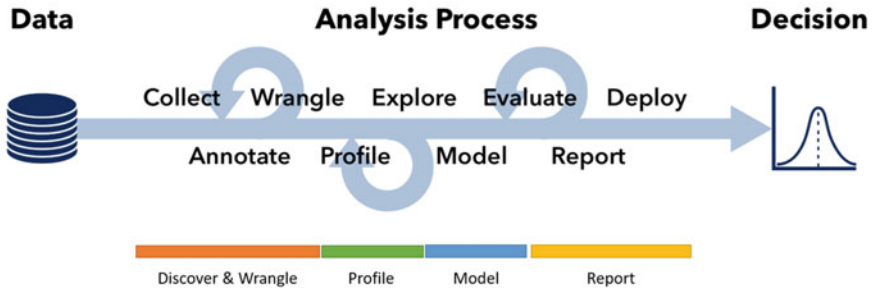
## 1.2 Data Analysis Workflow

Data visualization is progressively being applied to data analysis, and the exchange between data analysts, data scientists, and data visualization researchers is essential for visualization research (Alspaugh et al., 2019). Not only do data analysts give valuable feedback on techniques and applications to improve the proposed research results further, but data visualization researchers also depend on these new tasks, new directions for further research, and new interesting datasets and application ideas.

Within the last years, data science has emerged as its own research field since data has transformed from a purely confirmatory analysis to a more exploratory approach. As data is often explored without a pre-defined hypothesis and, in many cases, without prior knowledge (Egger & Yu, 2022a), data science seems to comprise more than pure statistical data analytics. In this sense, data science needs to act as an interdisciplinary concept that combines input from other domains like mathematics, statistics, computer science, and/or graphics (Egger & Yu, 2022b; Parsons et al., 2011). Data science also involves taking domain knowledge into consideration during the analysis as well as when dealing with the interpretation of the data and the results (Blei & Smyth, 2017).

Several studies have been conducted over the past years to better understand the tasks and requirements of data scientists (Harris et al., 2013)—for example, when working closely together with software development teams (Kim et al., 2018). In addition, the usage of tools and applications and the decision toward specific workflows are also of great interest (Liu et al., 2019). As an essential step forward, the workflow of data analysts has been categorized into systematic steps (Kandel et al., 2012). The categorization is based on semi-structured interviews with data analysts from different organizations, including companies from healthcare, retail, marketing, and finance, and, therefore, covers a broad range of disciplines. Thus, the workflow of all data analysts can be summarized into the following five high-level categories:

- **Discover:** As a first step, data scientists usually search for suitable datasets, either by locating them in databases, browsing online, or asking colleagues. Especially within large organizations, the aspects of finding and understanding relevant data as well as access restrictions are often considered a significant bottleneck in the work process.
- **Wrangle:** When available, the datasets need to be changed into the desired format. As such, data wrangling involves parsing files, manipulating data layouts, and also integrating multiple heterogeneous data sources. This process usually consumes the majority of one's time during data analysis, rendering it a very tedious and highly manual task.
- **Profile:** In the next stage, the quality of the data has to be verified. Datasets often contain severe flaws, including missing data, outliers, erroneous values, and other issues. Understanding the structure of the data is therefore considered a significant task in data science.



**Fig. 4** Data analysis workflow. The five stages of the workflow (Discover, Wrangle, Profile, Model, and Report) contain unstructured and iterative steps where data analysts often have to rethink their actions. Source: Heer (2019)

- **Model:** Finally, an essential and exciting part of the data science workflow is to use the datasets as training sets to train prediction models. In this stage, the models have to be created and evaluated against existing real-world data so as to test their performance.
- **Report:** All analysis results eventually need to be reported to stakeholders or an external person in charge. Thus, it is crucial to cover the essential findings discovered during the data science process in a presentation. In many cases, dashboards or reports are used to present such findings.

All the above-mentioned steps involve a circular process, meaning that data scientists typically have to reevaluate the actions they made and restart the analysis process from scratch. The repetitive nature of the workflow is illustrated in Fig. 4, showing the different steps of the data analysis workflow and indicating where data analysts most likely need to revise and repeat specific steps. This includes the *Discover* and *Wrangle* phases, where new datasets need to be included if the present ones do not fulfill the requirements. In the *Profile* phase, iterative analysis of the data ensures that all aspects of the data are covered in an exploratory analysis, while during the *Model* phase, by applying techniques from the *Report* phase, the model results are continuously evaluated and the parameters are adjusted accordingly. It is also possible that after the *Model* phase, analysts go back to the *Discover* and *Wrangle* phases in case new datasets need to be included in the analysis.

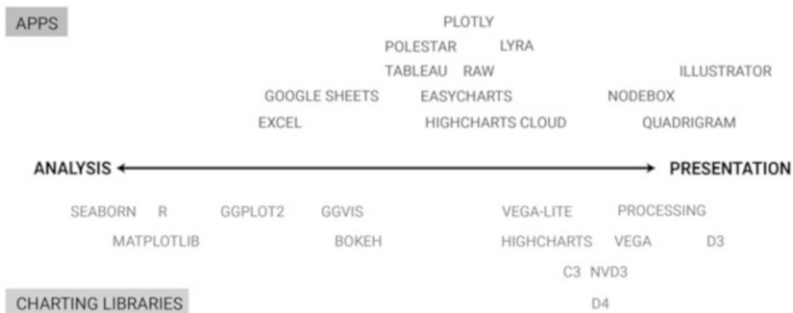
More and more data analysis software applications, many of which are open source, have evolved in the last few years (Barlas et al., 2020). Tools are often focused on specific tasks, such as efficient data storage and access (e.g., for Big Data applications), data wrangling (i.e., mapping data to another format), or automated analysis (e.g., machine learning). They are based on different programming languages (e.g., Python, R, and JavaScript) or are built as fully featured, standalone applications. However, due to the highly interactive and undirected workflow, no tool or application can cover the entire data science workflow process. Data scientists are therefore required to use a combination of different sets of tools to achieve their goals, and, depending on their skillset, they usually prefer to use either programming interfaces or fully featured applications (Liu et al., 2019). For instance, a data scientist who

identifies as an archetype hacker would not be happy using a standalone application because he/she would not be able to access the latest library in a scripting environment and, in turn, could not customize his/her workflow.

### 1.3 Data Visualization in the Data Analysis Workflow

More specifically, alongside the data analysis workflow itself, researchers also had a closer look at the data visualization tools being used within the workflow. These tools can be divided into charting libraries and apps (Rost, 2016). *Charting libraries* refer to the types of visualization libraries that require a programming environment to work. In many cases, this involves a scripting environment; therefore, many libraries nowadays are based on Python or R. The popularity of data visualization libraries, however, fluctuates and changes from year to year since many of these libraries are open source and undergo continuous adaptations and improvements. Some examples for charting libraries include ggplot2 (R), Matplotlib (Python), Seaborn (Python), Bokeh (Python), D3 (JavaScript), and Chart.js (JavaScript). The differences between these libraries are set by the different programming environments they are located in and by the different features and assets they offer for data visualization. *Apps* are considered to be fully featured, standalone applications that do not require any programming environment to be installed on a system in order for them to run. As apps are more targeted toward users without programming skills and who are unfamiliar with manual data processing, analytics, and visualization, data visualizations can be created by using the user interface tools provided by the application. In almost all cases, applications are commercial products since much maintenance and continuous developments are needed to keep the apps up-to-date. Tableau, Microsoft Power BI, and Qlik are considered to be the most prominent examples of apps.

When comparing charting libraries and apps, which have been classified as more suitable for exploration (as needed in the Profile stage) or presentation (as needed in the Report stage), exciting differences can be seen. The analysis in Fig. 5 illustrates



**Fig. 5** Charting libraries and apps. While charting libraries can also cover the analysis part (as needed in the Profile stage), apps are more suitable for the presentation aspect (as needed in the Report stage). Source: Rost (2016)

very nicely that charting libraries are, in general, suited for both analysis and presentation. Apps, however, find themselves more situated on the presentation side and are, therefore, less beneficial for the Profile stage. Interestingly, the charting libraries rather tailored for presentation are based on JavaScript (Vega-Lite, Processing, Highcharts, Vega, D3, C3, NVD3, D4), also confirming that web-based visualization methods are currently placed in the Report stage of the data science workflow. This also makes sense when keeping the client-server environment of web-based visualizations in mind as well as the fact that visualization designers have to carefully think about which type of data should be shown in this setting. Since large datasets are usually not transferrable over a network and could potentially lead to processing or rendering issues on the client's side (e.g., smartphones), such a careful design can typically only be achieved after the analysis (Profile and Model) has already been completed.

Alongside the broad selection of tools, data analysts are often confronted with the question of which visualization technique to use for which type of data and for which task. An example of such a decision is shown in Fig. 6 in which a temporal dataset containing a value for each month of the year is used (values are shown on top) to design data visualizations. In this example, eight different data visualizations have been created, but one could certainly come up with more and could think of a use case where all eight of these types of visualizations could be beneficial. For example, showing the values in a scatterplot (top left) is very helpful in finding outliers,



**Fig. 6** Representing data in different visual ways. Due to the multitude of visualization techniques being available, it is possible to make use of different techniques when visualizing a dataset. Source: Maguire (2017)

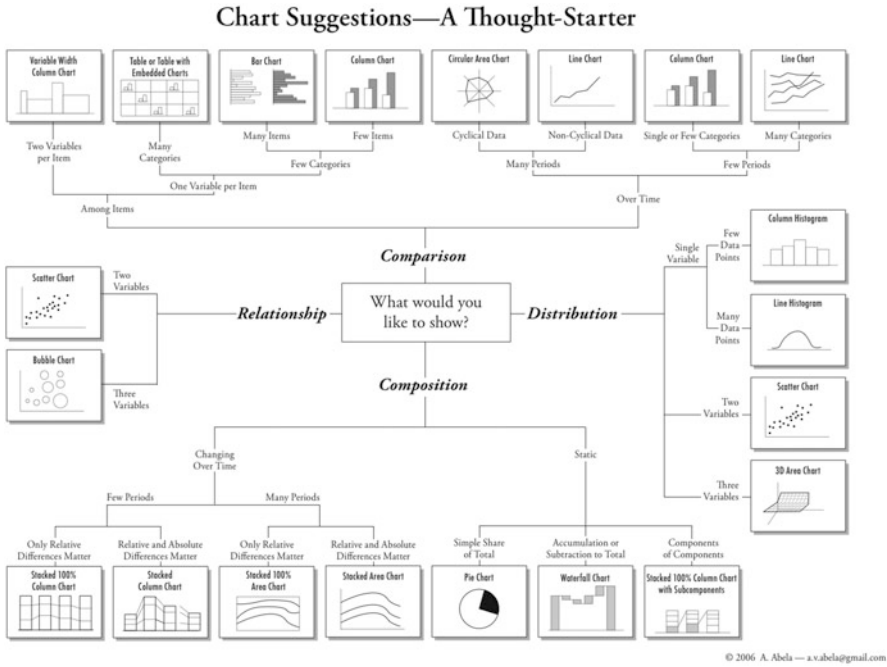
whereas using a line, as shown in the top right, can reveal trends. On the other hand, point-based visualizations, like the four at the bottom, can help to depict months with high or low values.

Overall, this example illustrates the necessity of analyzing the tasks, use cases, and end-users very carefully before determining which visualization would be most appropriate. Task- and data-dependent selection of visualization is, therefore, an essential topic in data visualization research. To this end, certain rules (Munzner, 2014) that should be taken into consideration when preparing and designing a data visualization have already been established:

- **No Unjustified 3D:** Due to many problems arising from occlusion, perspective distortion, and complex interactions, 3D representations should only be used if justifiable. This might be, for example, for the visualization of inherent 3D information (e.g., 3D geometry) or in cases where shape perception is an important asset.
- **Eyes Beat Memory:** It is easier for us to switch between different views (in a multi-view environment) than compare a current state to a visualization that we have seen before and is merely ingrained in our memory.
- **Interaction:** When using interaction, consider the principles of Overview-First-Details-on-Demand and Focus+Context. Furthermore, the latency and response time of the visualization has to be fast enough so that users do not lose attention/become uninterested.
- **Function First, Form Next:** Always be aware of the task that needs solving in order to precisely define what should be seen in the visualization. Also, careful consideration of the users is essential, i.e., whether they are familiar with the data and have enough experience interpreting visualizations.

Another problem in data visualization research that has yet to be solved is how to best define guidelines for creating visualizations (Diehl et al., 2018). Stemming from this need, automatic recommender systems for visualization have been created. These seek to automate the design of specific types of visualizations or seek more available selections on how to visualize a particular type of data. The Draco system (Moritz et al., 2019), for example, selects a suitable visualization technique for a given dataset. This is accomplished via pre-defined rules that are primarily based on the generally known best practices for the design of visualizations (e.g., the preferred usage of bar charts over pie charts). Notably, however, there is no consensus on evaluating data visualizations based on the meanings that humans may derive from it.

The field of *data visualization literacy*, which describes the ability of humans to create and read data visualizations (Börner et al., 2019), represents one approach where the human is placed at the focal point. Knowledge about data literacy is mainly collected via user studies (Börner et al., 2016), and such studies have shown that users still have problems interpreting specific charts, for example, network representations (Zoss, 2018). Previous studies have also revealed significant differences between novices and experts when using visualizations (Maltese et al., 2015) as well as a difference between static and interactive representations (Schwan & Riempp, 2004). Nevertheless, more studies with users from different backgrounds are needed in order to gain a better understanding of data visualizations.



**Fig. 7** Chart suggestion flowchart. Based on practical experience, researchers in data science came up with a concept to suggest suitable data visualizations for certain types of data and for specific tasks. Source: Abela (2009)

Another idea that was developed to try and create best-suited data visualizations was by means of implementing automatic evaluation metrics for existing data visualizations (Behrisch et al., 2018). Furthermore, an additional possibility of evaluating data visualizations is to interpret the statistical significance of human insights gained from visualizations (Wickham et al., 2010). However, these evaluation techniques have yet to be primarily linked to perceptual research and user studies. Other researchers, therefore, started to develop their own guidelines and suggestions for selecting data visualization techniques, one of the most prominent ones being the “Chart Suggestions” flowchart (Abela, 2009). In this flowchart, which is shown in Fig. 7, charts are suggested based on specific tasks (Comparison, Distribution, Composition, and Relationship) and the available data type.

## 2 Demonstration

Tourism data can contain various pieces of helpful information, ranging from information on hotel and sightseeing ratings to geospatial data on tourists moving in cities to numeric information on the travel modes of tourists. In this example, we will analyze

the global number of tourists per country and put them in relation to other country attributes. In many cases, the datasets for tourism data can be considered multivariate datasets since they contain a considerably large amount of attributes. Therefore, a multivariate dataset was also chosen for the purposes of this demonstration.

## 2.1 Datasets

For this demonstration, we combined two datasets. The first one (Tourism-Data) contains information about countries across the globe and the number of tourists that visited them. Each distinct year, from 1995 to 2017, was organized as a separate column and every country as a row, leading to 24 attributes available in the data (see Table 1). The second dataset (Country-Data) contains statistical information on every country in the world. In total, the dataset contains 50 attributes. For the analysis here, 13 attributes were chosen (see Table 2).

**Table 1** Tourism-Data. Dataset containing tourism attributes

Tourism-Data	
Country	String (name of the country)
1995	Numeric
1996	Numeric
1997	Numeric
...	...
2015	Numeric
2016	Numeric
2017	Numeric

**Table 2** Country-Data. Dataset containing country attributes

Country-Data	
Country	String (name of the country)
Region	String
Surface area	Numeric
Population	Numeric
Population density	Numeric
GDP per capita	Numeric
Economy: Agriculture (% of GVA)	Numeric
Economy: Industry (% of GVA)	Numeric
Economy: Services and others (% of GVA)	Numeric
Exports	Numeric
Imports	Numeric
Population growth rate	Numeric
Urban population (% of total population)	Numeric



## 2.2 Data Analysis Workflow

As mentioned before, the data analysis workflow can be divided into five stages: Discover, Wrangle, Profile, Model, and Report. As such, we will demonstrate the usage of data visualization in these five stages using the datasets described above.

### 2.2.1 Discover

In the Discover stage, data analysts try to find suitable datasets to answer the questions they have about the data and solve the task at hand. In our case, we started with the Tourism-Data as described in Table 1. In addition, we wanted to include more information about the countries mentioned in the dataset and place the country in relation to the tourism data information. Therefore, we searched for a dataset containing statistical information on countries and found what we were looking for in the Kaggle database (Kaggle, 2021). Online communities like Kaggle that are targeted toward data scientists often offer a broad range of freely available datasets. The same applies to open data portals offered by governments, e.g., the Open Data Portal of Austria (<https://www.data.gv.at/>).

Data visualization tends to be of little use when finding suitable datasets as searching for datasets requires other skills like domain knowledge and knowledge about data sources, which can hardly be supported by data visualization. Nonetheless, in the case of the Kaggle database, simple data visualizations like histograms and textual explanations give users a brief overview of the datasets they are currently interested in (Fig. 8). Quick insights as such can help assess whether the dataset would be suitable for the current task (e.g., if the data contains too many missing values or follows a specific distribution).



**Fig. 8** Data overview. Simple visualizations (text and histograms) used by Kaggle to give a quick overview of the data contained in different data columns

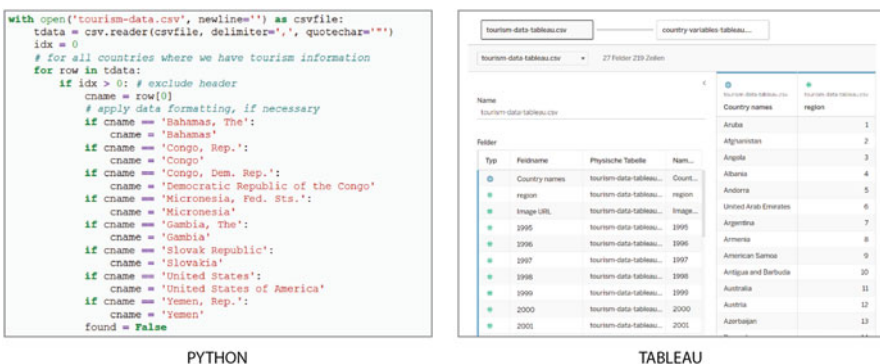
### 2.2.2 Wrangle

The data wrangling stage describes all necessary steps needed in order to transform the available datasets into its desired formats, clean the data, and most likely filter or pre-process it. Generally speaking, it can be estimated that wrangling data takes up roughly 50–80% of an analyst’s time (Rattenbury et al., 2017). Yet, despite data wrangling being a very tedious and highly manual task, it is also a crucial step, which needs to be done before any further analysis can take place. The ongoing process requires the data to be structured (e.g., in a data matrix).

In our demonstration case, we also have to perform data wrangling. Both datasets are available as CSV files, meaning they were already nicely structured as data matrices. There are, however, data quality issues that need to be solved. For example, instead of numeric information, some rows contain fuzzy information indicated by symbols like “~” (e.g., export = “~0.0”). Therefore, we decided to parse these rows and set the values to 0. Then, we joined the two datasets based on the country names in order to obtain one large data matrix composed of attributes from both datasets for most countries. Some country names could not be matched automatically, for example, “Bahamas” and “Bahamas, The,” “Vietnam” and “Viet Nam,” or “Czech Republic” and “Czechia,” and some countries like North Macedonia or Kosovo were not found in the list of countries and were therefore excluded from the analysis.

These simple examples show that data wrangling still requires many manual adjustments and can hardly be automatized. Tools like Trifacta (<https://www.trifacta.com>), however, aim at applying machine learning and natural language processing tools to ease the data wrangling process. These tools try to predict from the available data and the previous actions of the users which tasks the user needs to perform and suggest practical next steps (e.g., merging lines and columns). Data visualization is, once again, of little help during the data wrangling phase.

When using scripting languages like Python, one has to manually perform all tasks and define which data should be merged. Using data visualization apps like Tableau can provide some more advanced features for merging datasets (e.g., similar to joins in a database); however, this also does *not* solve any problem related to inconsistent data points. A comparison of Python and Tableau can be seen in Fig. 9.



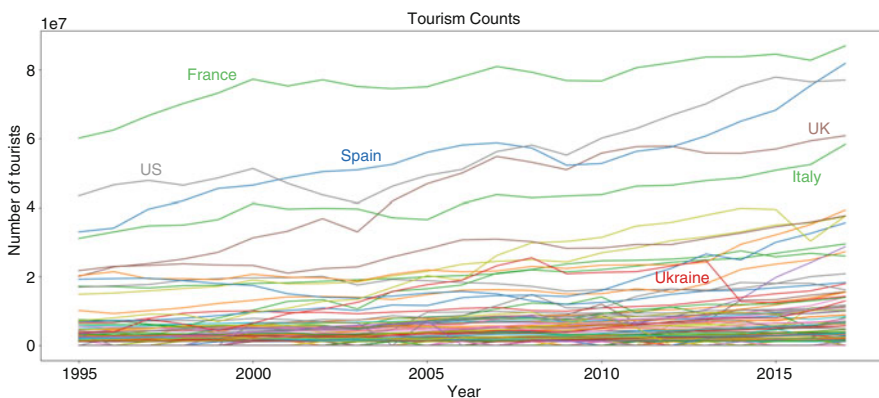
**Fig. 9** Joining two datasets. In Python, this has to be done manually by the data analyst. Apps like Tableau provide features for automatic dataset joins

### 2.2.3 Profile

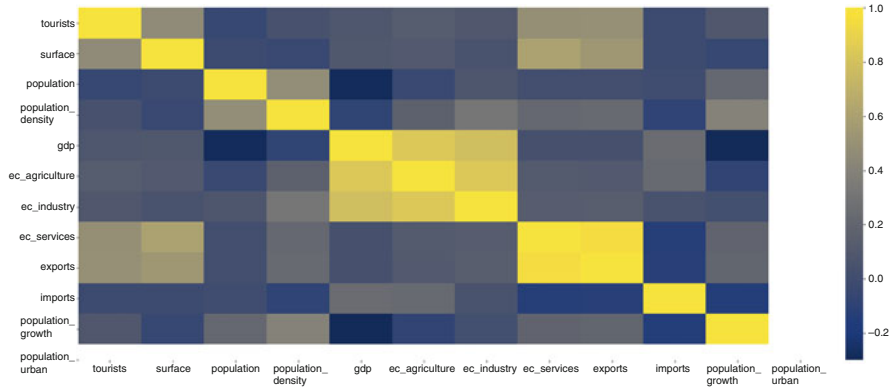
In the Profile stage, data analysts try to understand the data better and explore it in more detail. This is necessary to assess whether the data at hand contains all necessary information, if there are any data quality problems (e.g., missing data), and/or whether enough information is available for the cases that should be studied.

After merging our datasets, our data matrix contained 219 rows (lines) and 37 columns (attributes). We then wanted to explore whether we can detect any exciting patterns in the data. For this, we plotted all countries and their tourism counts over the years (Fig. 10). For many countries, the numbers increased, although there are some exceptions (e.g., Ukraine). We also tested to see if there was any correlation between country values and tourism counts over the years. For this, we created a correlation matrix for all attributes in the data in which the number of tourists was summed up to one value. In a correlation matrix, the diagonal is always 1 and the two sides to the left and right are similar. Visualizing such a correlation matrix using a heatmap is a convenient way to provide a quick overview of the possible correlations in the data. In Fig. 11, one can observe that there is no strong correlation between the number of tourists and other country parameters, but rather only a weak correlation in regard to the surface area of the country, the percentage of people working in the service sector, and the number of exports.

Exploratory data analysis (EDA) is very fast and efficient when using scripting languages like Python. It is not only easy to transfer the data into a new format but also allows one to quickly try out different data visualizations. In apps like Tableau, creating different plots results in a data analyst having to operate many different user interface elements, which might be a slower procedure than when using programming skills. Despite this, one advantage of apps is that simple user interactions like



**Fig. 10** Number of tourists for all 219 countries. There are some countries with very high tourist rates (France, United States, Spain, and Italy), but, overall, most of the countries saw a rise in the number of tourists over the years. While, in particular, the numbers in the UK increased drastically, some other countries such as Ukraine lost tourists after 2014

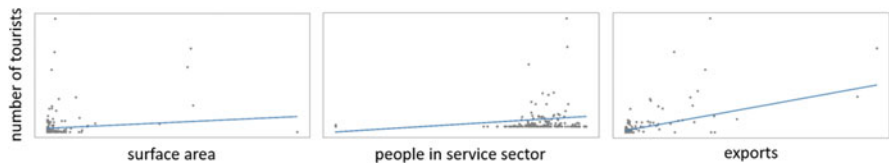


**Fig. 11** Correlation heatmap. No clear correlation between the total number of tourists and other country parameters can be seen. Only weak correlations with the surface area, the number of people working in services or other sectors, and to the amount of exports can be detected. Nevertheless, strong correlations can be found between other country parameters (e.g., GDP and people working in the industry sector)

mouse hovers, selections, and clicks are naturally supported and do not need to be manually included in the data visualization. When it comes to using scripts, it is also possible to reuse the code in new projects, and scripting environments like Jupyter Notebooks (<https://jupyter.org/>) enable data analysts to organize scripts in a narrative way, making it is easy to remember the steps that have already been taken (Schmidt & Ortner, 2020).

### 2.2.4 Model

During the Model stage, data analysts create a model out of the data to explain a specific phenomenon. In our example, by taking the parameters of surface area, number of people working in the service sector, and exports, we tried to predict the number of tourists using a linear regression model. The results are visualized in Fig. 12, and, as expected, the linear correlation between the parameters is not very strong. A better regression model can be created when combining these three



**Fig. 12** Visualization of regression models. As expected, the linear correlation between the number of tourists and the three country parameters shown here is not very strong

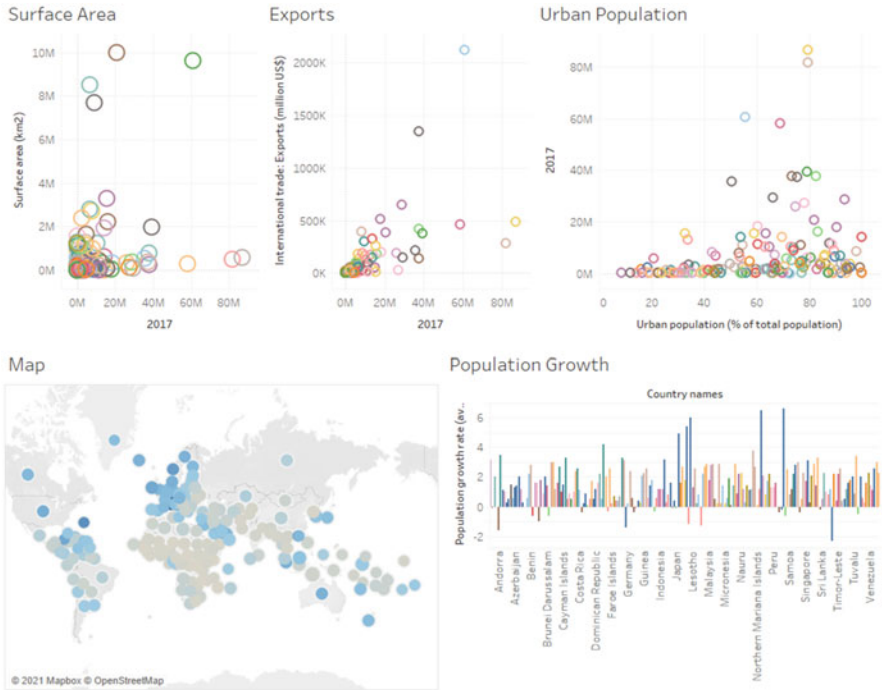
parameters to form a multivariate regression model. In this case, the problem is that the results are much more difficult to visualize than when only one parameter is used. Based on the outcomes of this stage, we can conclude that searching for new data might be of interest—for example, climate and weather data or information about the political situation. This would then be an iterative step back to the Discover and Wrangle stage.

In general, it is elementary to detect linear dependencies in data visualizations. Therefore, data visualization can confirm a hypothesis about the linear correlation between parameters in the data. To create more sophisticated models, one can turn to the research field of Explainable AI (Samek et al., 2019), where the aim is to provide visual means to better understand AI models (see Chap. 14). However, solutions in this direction are usually very targeted toward a particular domain and have not found their way into commonly used libraries.

### 2.2.5 Report

All results that have been derived from the data must be reported to others, such as stakeholders, additional data analysts, domain experts, and/or the public. As such, in the Report stage, data analysts must find the best way to communicate their findings to others. In many cases, this happens by creating textual reports or by using slides in a presentation. Yet, within the last years, data analysts have increasingly turned toward creating interactive dashboards, where users can explore the data themselves. The dashboards are often web-based, meaning that users merely require a web browser in order to access the data visualizations. The creation of dashboards is the key discipline of modern business intelligence (BI) tools like Tableau or Power BI. To provide an example, we also created a dashboard in Tableau for the tourism data, as shown in Fig. 13. Apps like Tableau offer many features for interaction as well as features for data filtering, which are already built-in and do not need to be manually defined by the data analyst. Creating dashboards is, therefore, much easier when using apps. Open Source solutions like Python Dash (<https://plotly.com/dash/>) also offer solutions for creating web-based dashboards, but here more programming skills and more manual effort are demanded to achieve the same result.

For a long time, dashboards were not recognized as their own data visualization technique; instead, they were seen as a much-maligned visualization vehicle of choice for decision-making in commercial and governmental situations. However, dashboards are now becoming many peoples' direct connection to “big data” sources, enabling data democratization and wider access to data. Researchers, therefore, have started to explore the genre of dashboards more, identifying different consumption by different parties, all of which have varying levels of visualization literacy, data literacy, and decision agency. Dashboards can, in this way, be classified according to their analysis goals, audiences, and decision support (Sarikaya



**Fig. 13** Dashboard for a report. BI tools especially are very well suited for creating interactive and, in many cases, web-based dashboards

et al., 2019). Very similar to other visualization techniques, a clear focus on the end-users, the tasks, and the questions one would like to have answered must be kept in mind when designing a dashboard.

### Service Section

**Main Application Fields:** Data visualization is a beneficial tool in data analysis as it enables us to use our human visual system to interpret large and complex data. When analyzing data, different tasks have to be performed, ranging from finding and preparing the data to analyzing and exploring it to finally reporting the findings. The influence of data visualization is different when it comes to each individual stage of the pipeline. While there is little support for data preparation, iterative exploration and data analysis itself can be greatly supported by different data visualization tools. To this end, many different techniques have been developed within the last years that allow making sense of multivariate data. Especially when using scripting languages, many different visualizations can be quickly created to view data from

(continued)

different perspectives and to study different aspects of the data. Data visualization also plays an important role when reporting findings to stakeholders and others, with primarily commercial applications being strongly targeted toward creating interactive dashboards where user engagement is involved. To present these foundations, the usage of data visualization throughout the data analysis workflow has been demonstrated with a dataset containing tourists counts for different countries in the world.

**Code:** The Python code is available at: <https://github.com/DataScience-in-Tourism/Chapter-25-Data-Visualization>

## References

- Abela, A. (2009). *Chart suggestions: A thought starter*. Accessed June 05, 2021, from <https://extremepresentation.typepad.com/files/choosing-a-good-chart-09.pdf>
- Alspaugh, S., Zokaie, N., Liu, A., Jin, C., & Hearst, M. A. (2019). Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE Transactions on Visualization and Computer Graphics*, 25(1), 22–31.
- Artal, P. (2016). *The eye as an optical instrument. Optics in our time*. Springer.
- Barlas, P., Lanning, I., & Heavey, C. (2020). A survey of open source data science tools. *International Journal of Intelligent Computing and Cybernetics*, 8(3), 232–261.
- Behrisch, M., Blumenschein, M., Kim, N. W., Shao, L., El-Assady, M., Fuchs, J., Seebacher, D., Diehl, A., Brandes, U., Pfister, H. P., Schreck, T., Weiskopf, D., & Keim, D. A. (2018). Quality metrics for information visualization. *Computer Graphics Forum*, 37, 625–662.
- Blei, D. M., & Smyth, P. (2017). Science and data science. *Proceedings of the National Academy of Sciences*, 114(33), 8689–8692.
- Börner, K., Bueckle, A., & Ginda, M. (2019). Data visualization literacy: Definitions, conceptual frameworks, exercises, and assessments. *PNAS*, 116(6), 1857–1864.
- Börner, K., Maltese, A., Balliet, R. N., & Heimlich, J. (2016). Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors. *Information Visualization*, 15, 198–213.
- Cohen, M. A., Dennett, D. C., & Kanwisher, N. (2016). What is the bandwidth of perceptual experience? *Trends in Cognitive Sciences*, 20(5), 324–335.
- Cuenca, E., Sallaberry, A., Wang, F. Y., & Poncelet, P. (2018). MultiStream: A multiresolution streamgraph approach to explore hierarchical time series. *IEEE Transactions on Visualization and Computer Graphics*, 24(12), 3160–3173.
- Diehl, A., Abdul-Rahman, A., El-Assady, M., Bach, B., Keim, D. A., & Chen, M. (2018). *VisGuides: A forum for discussing visualization guidelines* (EuroVis 2018 Short Papers).
- Egger, R., & Yu, C. (2022a). Epistemological challenges. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 17–34). Springer.
- Egger, R., & Yu, C.-E. (2022b). Data science and interdisciplinarity. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism* (pp. 35–49). Springer.
- Esprabens, J., Arango, A., & Kim, J. (2020). *Time series for beginners*. Accessed June 30, 2021, from <https://bookdown.org/JakeEsprabens/431-Time-Series/>
- Grandjean, M. (2014). La connaissance est un réseau. *Les Cahiers du Numérique*, 10(3), 37–54.
- Harris, H. D., Murphy, S. P., & Vaisman, M. (2013). *Analyzing the analyzers: An introspective survey of data scientists and their work*. O'Reilly Media.

- Heer, J. (2019). *EuroVis Capstone Talk* <http://eurovis2019.tecnico.ulisboa.pt/wp-content/uploads/2019/06/EuroVis2019-Capstone.pdf>
- Kaggle. (2021). *Datasets*. Accessed July 01, 2021, from <https://www.kaggle.com/datasets>
- Kandel, S., Paepcke, A., Hellerstein, J. M., & Heer, J. (2012). Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12), 2917–2926.
- Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2018). Data scientists in software teams: State of the art and challenges. *IEEE Transactions on Software Engineering*, 44(11), 1024–1038.
- Liu, J., Boukhelifa, N., & Eagan, J. R. (2019). Understanding the role of alternatives in data analysis practices. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 66–76.
- Maguire, E. (2017). *Principles of data visualization*. CERN School of Computing Talk. Accessed May 07, 2021, from <https://www.slideshare.net/eamonnmag/principles-of-data-visualization-71834041>
- Maltese, A. V., Harsh, J. A., & Svetina, D. (2015). Data visualization literacy: Investigating data interpretation along the novice-expert continuum. *Journal of College Science Teaching*, 45, 84–90.
- McNabb, L., & Laramée, R. S. (2017). Survey of surveys (SoS) – Mapping the landscape of survey papers in information visualization. *Computer Graphics Forum*, 36, 589–617.
- Moritz, D., Wang, C., Nelson, G., Lin, H., Smith, A. M., Howe, B., & Heer, J. (2019). Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1), 438–448.
- Munzner, T. (2014) *Visualization analysis & design*. AK Peters Visualization Series.
- Parsons, M. A., Godøy, Ø., LeDrew, E., de Bruin, T. F., Danis, B., Tomlinson, S., & Carlson, D. (2011). A conceptual framework for managing very diverse data for complex, interdisciplinary science. *Journal of Information Science*, 37(6), 555–569.
- Post, F. H., Nielson, G., & Bonneau, G.-P. (2003). *Data visualization – The state of the art*. Springer Science+Business Media. 9781402072598.
- Pripdeevech, P., Rothwell, J., D'Souza, P., & Panuwet, P. (2018). Differentiation of volatile profiles of Thai oolong tea no. 12 provenances by SPME-GC-MS combined with principal component analysis. *International Journal of Food Properties*, 20, 1–13.
- Rattenbury, T., Hellerstein, J. M., Heer, J., Kandel, S., & Carreras, C. (2017). *Principles of data wrangling: Practical techniques for data preparation* (1st. ed.). O'Reilly Media.
- Rees, D., & Laramée, R. S. (2019). A survey of information visualization books. *Computer Graphics Forum*, 38(1), 610–646.
- Rost, L. C. (2016). *What I learned recreating one chart using 24 tools*. Accessed March 05, 2021, from <https://source.opennews.org/articles/what-i-learned-recreating-one-chart-using-24-tools/>
- Samek, W., Montavon, G., Vedaldi, A., Hansen, L. K., & Müller, K.-R. (2019). *Explainable AI: Interpreting, explaining and visualizing deep learning*. Springer International Publishing.
- Sarikaya, A., Correll, M., Bartram, L., Tory, M., & Fisher, D. (2019). What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization and Computer Graphics*, 29(1), 682–692.
- Schmidt, J. & Ortner, T. (2020) *Visualization in Notebook-Style Interfaces*. In Proceedings of VisGap – The Gap between Visualization Research and Visualization Software. EuroVis 2020 Workshops.
- Schwan, S., & Riempp, R. (2004). The cognitive benefits of interactive videos: Learning to tie nautical knots. *Learning and Instruction*, 14(3), 293–305.
- Soares, A. G. M., Miranda, E. T. C., Lima, R. S., Resque dos Santos, C. G., & Meiguins, B. S. (2020). Depicting more information in enriched Squarified Treemaps with layered glyphs. *Information*, 11(2), 123.
- Strode, G., Morgan, J. D., Thornton, B., Mesev, V., Rau, E., Shortes, S., & Johnson, N. (2020). Operationalizing Trumbo's principles of bivariate choropleth map design. *Cartographic Perspectives*, 94, 5–24.



- Ware, C. (2019). *Information visualization: Perception for design* (4th ed.). Morgan Kaufmann.
- Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2010). Graphical inference for Infovis. *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 973–979.
- Wilkinson, L. (2005). *The grammar of graphics (statistics and computing)*. Springer. 0387245448.
- Zoss, A. (2018). *Network visualization literacy: Task, context, and layout*. PhD thesis, Indiana University, Bloomington, IN.

# Software and Tools



Roman Egger

While the field of Data Science (DS) continues to grow in strength and popularity, its repertoire of methods, including a wide plethora of various frameworks, software, and tools, is also increasingly advancing in parallel. As such, data scientists, researchers, and other users are not only faced with the challenge of keeping track of all the available solutions and alternatives on the market but also of choosing which tool is most suitable for each individual project. Furthermore, in addition to the requirements of having a solid theoretical background and understanding the broad range of methodological skills, it is also necessary to be able to use and apply these solutions in a correct and appropriate manner. For example, if one wishes to apply deep learning methods, available frameworks could include Caffe, Torch, Keras, Theano, or TensorFlow, to mention just a few. The question now remains: how, when, and where can these be sufficiently applied? Such frameworks are primarily based in Python, although modules are also available for MATLAB or R. Nonetheless, this presupposes that one is well-versed in at least one of these scripting or programming languages. Python is undoubtedly the most important programming language in the field of DS, but also R or even new programming languages like Julia are suitable. In the practical demonstrations provided in the individual chapters of this book, Python was used almost exclusively, with the codes provided in the form of Jupyter Notebooks on the book's Github profile (<https://github.com/DataScience-in-Tourism/>).

As mentioned in the introductory chapter on Machine Learning (Chap. 6), numerous cloud solutions (e.g., from Google, Amazon, IBM, Microsoft, etc.) exist for outsourcing ML tasks (partly as AutoML solutions). This is particularly relevant when Big Data is involved in an analysis since, when using one's own hardware, it can quickly happen that the available computing power is too low; for example, if

---

R. Egger (✉)

Innovation and Management in Tourism, Salzburg University of Applied Sciences, Salzburg, Austria

e-mail: [roman.egger@fh-salzburg.ac.at](mailto:roman.egger@fh-salzburg.ac.at)

GPU use is not rendered possible due to the hardware equipment. In such a case, cloud solutions seem to be the only practical alternative. If one wishes to avoid the costly use and configuration of such solutions (for instance, training a BERT language model from scratch can easily take up to several days with high-performance hardware), Google Colab (or Colab Pro) is a reliable alternative (depending on the size of the project). Colab allows browser-based Python coding and the execution thereof in Google Cloud servers while also providing free access to GPUs and TPUs. Although these resources have limitations in terms of memory, runtime (max. 12 h), faster GPUs, etc., they should be sufficient for most research projects.

This book is not intended to guide and support data scientists but is written for practitioners and researchers working in the tourism domain. Therefore, it is likely that the majority of readers will have little to no programming knowledge. Python is considered a relatively easy programming language, with numerous excellent online courses to learn the basics of Python programming. Nevertheless, there are also numerous solutions that do not require any programming knowledge and can be used in the form of visual programming. These include Rapidminer, Knime, and Orange, which will be discussed in more detail later on. What these solutions have in common is that one can put together a workflow by dragging and dropping the available modules and widgets, which are constantly being improved and extended. Thus, most of the methods described in this book (both supervised and unsupervised algorithms for the analysis of numerical data as well as for text and image analysis) can be performed with these non-programming alternatives. Furthermore, one is naturally dependent on the existing choices, which can lead to the fact that, for example, specific setting options may no longer be available for an algorithm. Although it is not necessary to have programming skills when using one of the tools, a thorough understanding of the processes and algorithms in use is required nonetheless. Finally, independent software solutions and platforms have additionally been developed for a huge variety of DS tasks at enterprise scale.

In order to give the reader a rough overview of the available tools and their performance spectrum, a selection of software, platforms, and solutions for data science is briefly presented below. With regard to the tabular list provided at the end, the authors of the respective chapters have compiled a list of relevant software packages in relation to their content. On that note, I would like to take this opportunity to thank the authors as well as the developers and providers of the solutions for the texts and short software descriptions they provided.

# 1 RapidMiner (by Wolfram Höpken)

RapidMiner Studio<sup>1</sup> is a visual workflow designer for data science that especially supports all relevant phases of the Cross Industry Standard Process for Data Mining (CRISP-DM). In addition to a visual programming interface, RapidMiner Studio offers fully interactive modules for data preparation, modeling, and deployment. Figure 1 shows the visual workflow designer with available operators on the left, a data science process consisting of a flow of operators in the middle, and parameters for each operator on the right.

In the *data understanding* phase, RapidMiner offers access to source data of any kind and format, including structured data (e.g., CSV files, Excel files, databases, etc.) as well as semi-structured data (e.g., html files) or free text. Specific extensions additionally support the extraction of data from web resources via web scraping or API access. Moreover, different kinds of visualizations (e.g., scatter plots, histograms, line charts, parallel coordinates, box plots, and interactive visualizations) allow for an interactive and ad-hoc explorative data analysis (EDA).

The phase of *data preparation* is supported by 100+ operators for cleansing and blending data, including typical approaches for normalization, outlier detection, or dimensionality reduction, which can be combined with complex and repeatable data preparation and ETL processes.

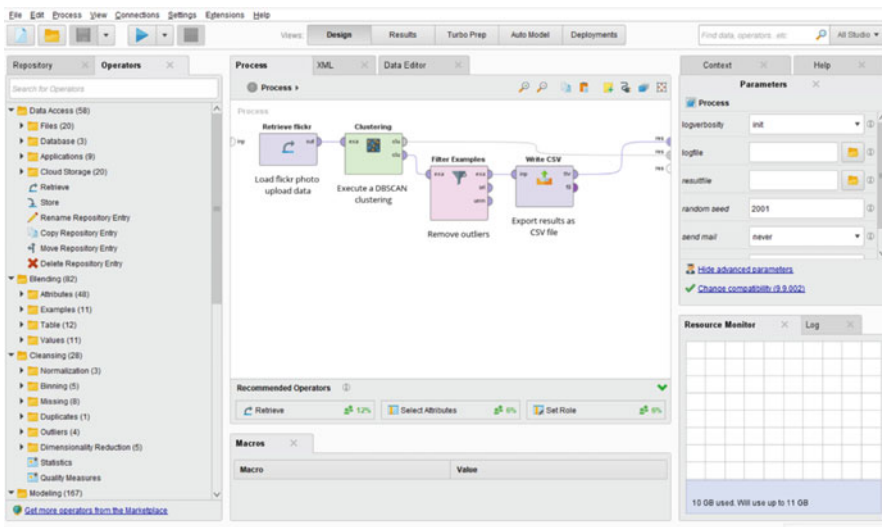


Fig. 1 RapidMiner Studio visual workflow designer

<sup>1</sup><https://rapidminer.com/products/studio/>

In the *modeling* phase, RapidMiner offers typical supervised and unsupervised machine learning techniques, including regression, clustering, time-series, text analytics, and deep learning, and supports manual as well as automated feature engineering. Model *validation* is then supported by typical cross-validation approaches, model visualizations like a ROC plot, or an interactive model simulator, enabling a kind of sensitivity analysis.

Finally, in the phase of model *deployment*, RapidMiner offers specific reinforcement whilst putting models into production and managing them properly. Beyond the client application RapidMiner Studio, the server application RapidMiner AI Hub authorizes the scheduling of data science processes and deploys models in the form of web services.

RapidMiner Studio's visual programming approach offers a high degree of flexibility and extendibility. Besides over 400 built-in operators, other specific extensions are available, for example, the full WEKA library, or extensions for information extraction, web mining, time series analysis, etc. One's own extensions can also be developed as Java programs and added to the publicly available marketplace. Additionally, these built-in operators can be seamlessly integrated into Python codes or R scripts.

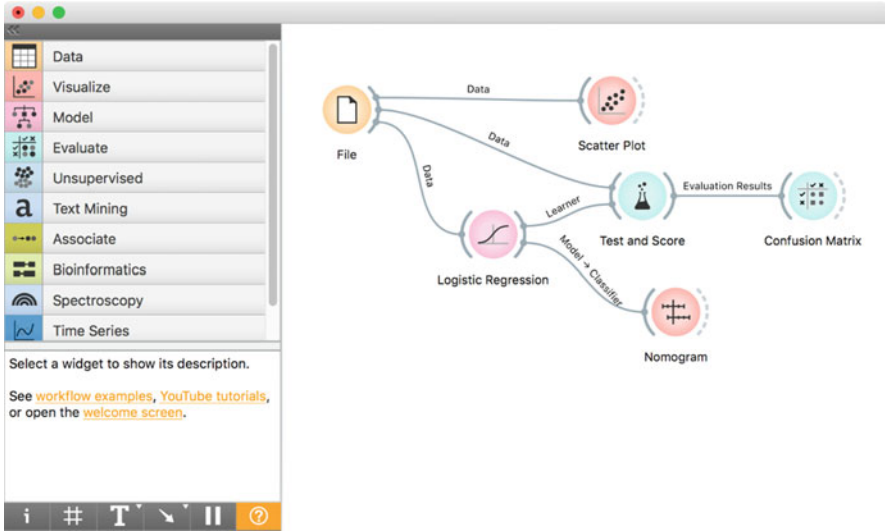
## 2 Orange (by Ajda Pretnar)

Orange is an open-source visual programming tool for data mining and machine learning. It was developed by the Laboratory for Bioinformatics at the Faculty of Computer and Information Science, University of Ljubljana, and is freely available at <https://orangedatamining.com/>. The software is intended for beginners and experts in data science, who prefer building visual workflows instead of coding.

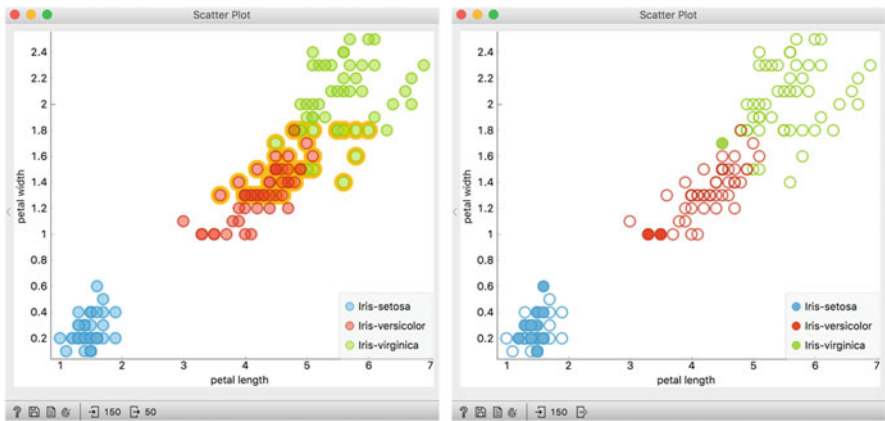
The main component of Orange is the canvas where users build analytical workflows, and these workflows are composed of individual components called widgets. Orange's core includes widgets for data loading and pre-processing, visualization, predictive modeling, model evaluation, and unsupervised learning. Additional widgets are available as add-ons for model explanation, association rule mining, text mining, time series, geolocated data, image analytics, network analysis, bioinformatics, single-cell analysis, spectroscopy, and data science education (Fig. 2).

One of the main features of Orange include the interactive visualizations; the user can select a subset of the data directly from the visualization and inspect it. Alternatively, he/she can highlight a subset of interest in the visualization itself (Fig. 3).

Orange has also been designed with educators in mind. Aside from having a graphical user interface for machine learning, it also offers visualizations of key algorithms and data generation widgets, making it a valuable tool for teaching (Fig. 4).



**Fig. 2** Canvas with a predictive modeling workflow. A list of widgets organized by categories can be seen on the left, while a workflow, which loads the data, shows it in a scatter plot, learns a logistic regression model, and inspects it in a nomogram and confusion matrix is located on the right



**Fig. 3** A scatter plot with selected data points (left), and a scatter plot with highlighted input/data from a subset (right)

The description and purpose of the widgets are available as a document,<sup>2</sup> and new users can watch YouTube tutorials<sup>3</sup> to get started or to find out how the software can be used for different use cases. Moreover, the blog demonstrates interesting

<sup>2</sup><https://orangedatamining.com/docs/>

<sup>3</sup><https://www.youtube.com/c/OrangeDataMining>

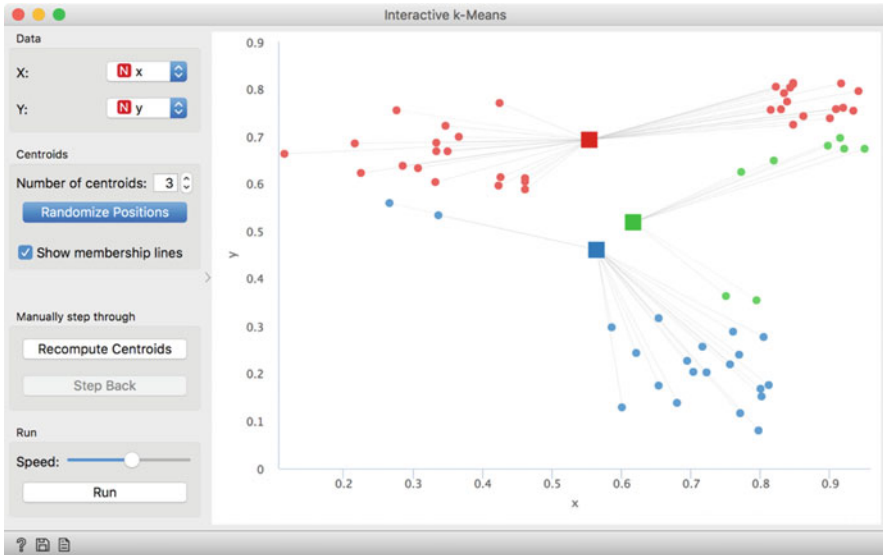


Fig. 4 Interactive k-means showing the algorithm's steps

workflows and explains new features of the software, while, lastly, educators can download sample scripts for teaching with Orange.

### 3 KNIME Analytics Platform (by Stefan Helfrich)

KNIME Analytics Platform (often referred to as just KNIME<sup>4</sup>) is an open source, modular platform for performing data science tasks. It covers the entire data science life cycle: from accessing and blending data to data preparation and transformation, from data visualization to training machine learning models, and, finally, from testing to deployment. KNIME's Visual Programming approach makes both simple and complex analyses (using state-of-the-art analytics methods) accessible to anyone, while removing unnecessary technical complexities. Users can define visual workflows for their analyses themselves, borrow blueprints from the community, or re-use workflows built by colleagues (Fig. 5).

Workflows are built by dragging and dropping so-called "nodes" into the workflow editor. Each node implements a dedicated task, and their "input ports" and "output ports" are connected to define the flow of data between nodes. The resulting workflow defines the entire analysis, from raw data to processed results, models, or reports (see Fig. 6 for an example). Intermediate results, data, and models

<sup>4</sup><https://www.knime.com/>

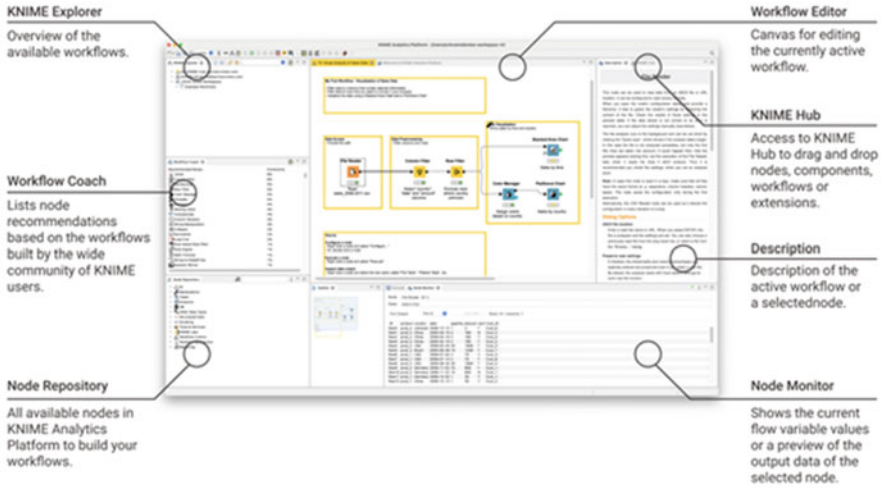


Fig. 5 Overview of the KNIME Analytics Platform’s workbench

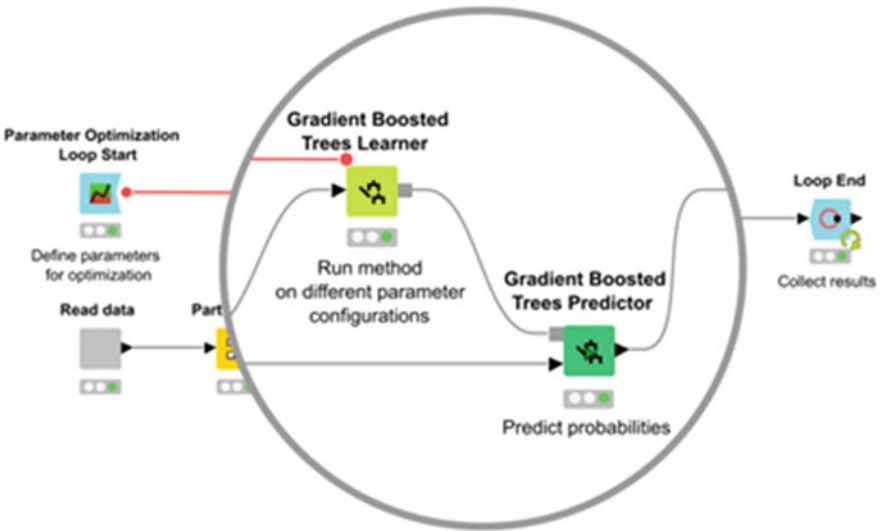


Fig. 6 Workflow for the automated optimization of a Gradient Boosted Tree model for classification. The zoomed in part of the figure shows two nodes that both retrieve data (black triangles) but also pass an ML model (gray rectangles) between them

can easily be inspected at output ports while the workflow is built, which is particularly handy for spotting any errors.

As an open (source) platform, KNIME also embraces a wide variety of other technologies and tools via extensions and integrations, such as KNIME Database Extension, KNIME Deep Learning—Keras Integration, and KNIME Tableau



Integration. This allows for the continuous integration of new methods and the support of additional data types so that nodes can process not only structured data but also text documents, images, and molecular structures, to name only a few. On top of that, the visual environment provides the right amount of abstraction to share work, both inside and outside of a team. For example, segments of workflows can be packaged into so-called “components” for reuse and sharing (via KNIME Hub).

A more considerable amount of information regarding (the use of) KNIME Analytics Platform is available online. It is complemented by the commercial KNIME Server for collaboration, automation, cloud execution, and deployment of data science workflows as analytical applications and services to end users. KNIME Software thus enables users to create and productionize data science in one, uniform environment.

## 4 WEKA (by Tony C Smith)

A great way to go about data mining via using machine learning algorithms is with WEKA, one of the most widely used, open-source, free data mining tools available on the market. It was the first to bring together just about all available machine learning algorithms, complete with visualization tools, data filters, and utilities for experimental design, into one single user-friendly application. It is continuously supported and kept up-to-date by a large user community, and a vast array of free literature, tutorials, and online courses are easily found on the web. Written entirely in Java, it is transferable to basically every platform and includes a well-documented API for experienced programmers who want to include machine learning in their own software or processing pipeline. Moreover, WEKA is available as a package for inclusion within other programming environments such as Python or R (Fig. 7).

WEKA allows users to load data from a CSV or relational file (amongst many other file types), import it directly from a database, or get data off the web via a URL. The downloaded software even includes a range of small, well-known public datasets to practice and learn with. Once loaded, a number of visualization tools and statistical measures allow the user to inspect the characteristics of their data and then choose, from hundreds of filters, actions such as discretizing, normalizing, standardizing, converting values from one type to another, or even generating new derivative features. Thereafter, when the data is ready, separate tabs shown on the interface enable the selection between prediction tasks (i.e., classification and regression), clustering, and association mining. Within each tab, the user can select the specific algorithm they would like to trial as well as the train-and-test procedure they would like to employ. Each algorithm lets more experienced users change default parameter values, while, for new users, pop-up help is available to guide and inform them on how to use that algorithm.

WEKA also includes visualization tools to let users evaluate their results, and the experiments can be saved for re-examination or re-running at a later point in time. Of course, this also allows users to save models for future deployment or inclusion in

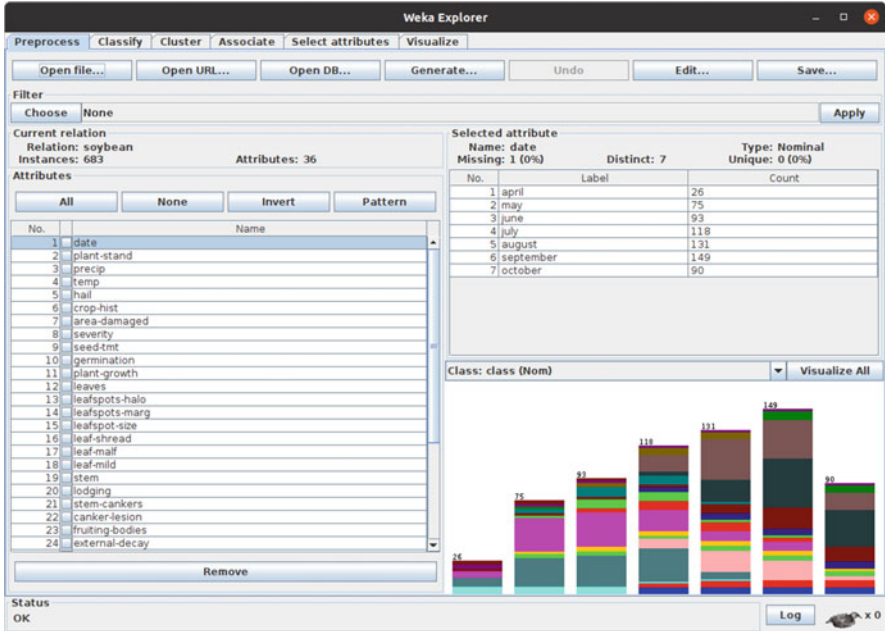


Fig. 7 WEKA opening screen

other software systems. In short, whether you are new to data mining or well-versed in it, or if you just want a free and simple, yet comprehensive, point-and-click application to experiment with machine learning, then WEKA is indeed a good choice.

## 5 SAS Viya (by Piere Paolo Ippolito)

SAS Viya is a cloud-based analytics framework designed to help organizations speed up their data analytics workflow. As such, the key objective of SAS Viya is to support organizations performing any big data analytics task, from data collection to deployment monitoring. The key advantages of SAS Viya include their characteristics of being decision-focused, developer-friendly, automated (automated data preparation and ML), and governed.

In order to achieve the best performances when creating a machine learning model, SAS Viya provides different hyperparameter optimization techniques such as Random and Grid Search, Bayesian Optimization, Genetic Algorithms, and Latin Hypercube Sampling. Additionally, SAS provides various general-purpose optimization techniques such as Linear, Quadratic, Constraint, and Non-linear

Programming, which can be used in applications in tourism (e.g., Hotel Analytics<sup>5</sup>) and forecasting.

Free SAS Viya licenses are currently available for both academics and students through the SAS Viya for Learners program.<sup>6</sup>

## 6 BigML (by BigML)

Simply stated, BigML removes the complexities of Machine Learning so that businesses can focus on what matters most: enhancing and automating decision-making.

In addition to data wrangling features, BigML provides a selection of robustly engineered Machine Learning algorithms proven to solve real-world problems by applying a single, standardized framework. This helps users to avoid dependencies on many disparate libraries that increase complexity, maintenance costs, and technical debt in their projects. BigML covers not only classification (Fig. 8), regression, and time series forecasting but also unsupervised learning tasks such as cluster analysis, anomaly detection, topic modeling, and association discovery in order to facilitate predictive data applications for a variety of verticals like automotive, transportation, pharmaceuticals, and more.

BigML also has a built-in AutoML capability named OptiML. OptiML utilizes an optimization process for model selection and parametrization, which automatically finds the best supervised model to help you solve classification and regression problems. Using Bayesian Parameter Optimization, it creates and evaluates hundreds of supervised models (decision trees, ensembles, logistic regressions, and deepnets) and returns a list of the best models for your data. This not only eliminates the need for manual, trial-and-error-based exploration of algorithms and parameters but also saves significant time and provides improved performance for Machine Learning practitioners of all levels.

Anyone can create a free account on BigML and gain immediate access to the wealth of functionalities mentioned above. For businesses, BigML offers the ability to manage user groups with different levels of access privileges as well as choices to deploy BigML in the cloud or on-premises. BigML is very developer friendly with a full-featured REST API underpinning the platform. Bindings and libraries are available for all popular languages, including Python, Node.js, Ruby, Java, Swift, and more. Further advantages of BigML include:

- **Reproducible:** Often completely overlooked in other Machine Learning tools, BigML's detailed record-keeping and transparency are crucial to meet regulatory and audit compliance requirements.

---

<sup>5</sup>[https://www.sas.com/en\\_us/industry/hotels.html#all-solutions](https://www.sas.com/en_us/industry/hotels.html#all-solutions)

<sup>6</sup><https://tinyurl.com/SAS-learners>



Fig. 8 Interpretable and Exportable Models

- **Traceable:** All resources on BigML are immutable and stored with a unique ID and creation parameters, which enable you to track any Machine Learning workflow at any time.

To find out more, BigML offers a number of educational videos (<https://bigml.com/education/videos>) and detailed documentation (<https://bigml.com/documentation/>).

## 7 Dataiku (by Laura Wiest)

Dataiku is another AI and machine learning platform, providing agility in organizations' data efforts via collaborative, elastic, and responsible AI, all at enterprise scale. At its core, Dataiku believes that in order to stay relevant in today's changing world, companies need to harness Enterprise AI as a widespread organizational asset instead of siloing it into a specific team or role. To make this vision of Enterprise AI a reality, Dataiku provides one simple UI for the entire data pipeline, from data

preparation and exploration to machine learning model building, deployment, monitoring, and everything in between.

Dataiku was built from the ground up to support usability in every step of the data pipeline and across all profiles, from data scientist to cloud architect to analyst. Point-and-click features allow those on the business side and other non-coders to explore data and apply AutoML in a visual interface. At the same time, robust coding features (including interactive Python, R, and SQL notebooks, the ability to create reusable components and environments, and much more) enable data scientists and other coders to work hands-on as well.

Since each company's path to Enterprise AI looks different, Dataiku supports the creation of a spectrum of applications, whether that means building out a self-serve analytics platform or a fully operationalized AI integrated with business processes. No matter what underlying changes in architecture or advancements in technology take place, Dataiku remains at the center as the cornerstone of data governance and responsible AI. Dataiku continuously incorporates the most recent technologies available in order to lower the barrier to integration for the companies themselves. These include computation, storage, programming languages, machine learning technologies, and more.

Dataiku's centralized, controlled, and flexible environment fuels exponential growth in the amount of data, the number of AI projects, and the number of people contributing to such projects. The platform was built to scale as businesses strive to go from a handful of models in production to hundreds (or thousands). Bottom line is that Dataiku is built for every industry, every use case, and also for everyone.

## 8 DataRobot (by DataRobot)

DataRobot is a self-service enterprise AI platform that, by automating end-to-end lifecycles, makes data science more accessible to everyone. With DataRobot, users can quickly and easily prepare their data and, thereafter, create and test a wide variety of traditional and newer machine learning models to find the best model for the use case at hand. These models can then be deployed to a variety of production environments and monitored and managed over time. In this way, the DataRobot platform allows users to create and deploy trusted AI applications at scale, giving organizations and individuals the power to gain predictive insights and find better solutions. Moreover, by using DataRobot, all key stakeholders can also extract business value from data.

Referring to Fig. 9 above, one can see that DataRobot is a platform catered to various types of users. AI Creators (data scientists, software developers, and business/data analysts), for instance, can gain access to hundreds of the latest machine learning algorithms in order to build, tune, and deploy models, with full transparency and control over these processes. Meanwhile, AI Operators (IT, DevOps) can put machine learning models into production with just a few clicks as well as govern, monitor, and manage those models over their entire lifecycles. Finally, AI

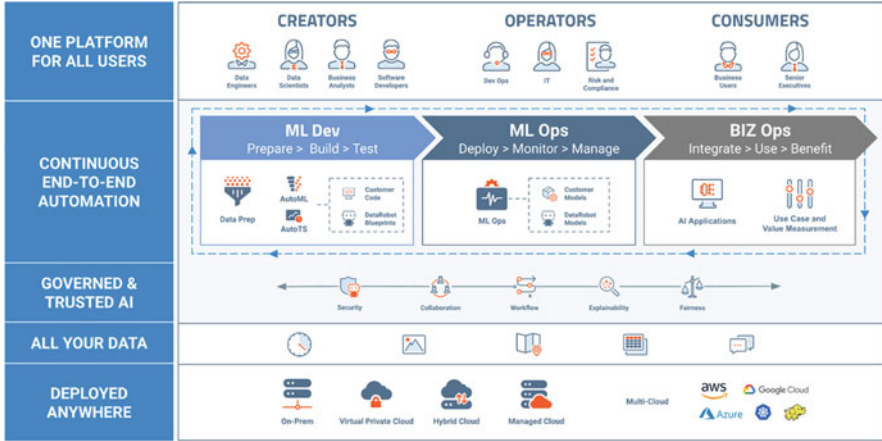


Fig. 9 DataRobot AI Platform

Consumers (business executives, analytics, and department leaders) can leverage pre-built AI applications and find even more opportunities to apply AI and data-driven decision-making to help derive a faster ROI.

DataRobot also makes it easy for students and teachers at eligible universities to gain access for teaching, learning, and research purposes. For example, through the Academic Support program, advanced machine learning can be made accessible without programming (<https://www.datarobot.com/education/academic-support-program/>). Furthermore, there are also free practical data science classes available online through DataRobot University <https://university.datarobot.com/>.

This table gives an overview of the software, tools, and solutions from all the non-theoretical chapters, which was compiled by the authors of each respective chapter. The links are also available online in each chapter section at: <http://www.datascience-in-tourism.com/>.

<b>Chapter 5</b>		<b>Web Scraping</b>		
Name	Website	Description	Costs	Distribution
Beautiful Soup	<a href="https://www.crummy.com/software/BeautifulSoup/bs4/doc/">https://www.crummy.com/software/BeautifulSoup/bs4/doc/</a>	Beautiful Soup is a Python library that can be used to read data from HTML and XML files. It offers possibilities to navigate, search, and change parse trees.	Free	Python library
Scrapy	<a href="https://scrapy.org/">https://scrapy.org/</a>	A framework for extracting data from websites, where one can build and run web spiders.	Free	Python library
Selenium	<a href="https://www.selenium.dev/">https://www.selenium.dev/</a>	Technically, a software for automating web applications for testing purposes, but it can also be used for web scraping.	Free	Python library
Octoparse	<a href="https://www.octoparse.com/">https://www.octoparse.com/</a>	Web scraping without coding. Turns web pages into structured spreadsheets by building workflows.	Free trial; Prices starting from \$75/Month	Download—Stand-alone
Scrapesform	<a href="https://www.scrapesform.com/">https://www.scrapesform.com/</a>	AI-Powered Visual Web Scraping Tool. No programming needed.	Free; Starting prices from \$50/Month	Download—Stand-alone

Machine Learning			
Name	Website	Description	Costs
Prodigy 101	<a href="https://prodigy.ai/">https://prodigy.ai/</a>	Prodigy is an annotation tool to create training and evaluation data for ML models. It also helps to inspect and clean data and to perform error analysis.	Lifetime license: Personal \$390 Company \$490
Orange	<a href="https://orangedatamining.com/">https://orangedatamining.com/</a>	An open-source visual programming data mining and machine learning software. The focus is on exploratory data analysis, interactive visualizations, and rapid prototyping. The program offers additional specialized components for text mining, network analysis, associative rule mining, analysis of geolocated and time-series data, image analytics, and much more.	Free—GNU v.3.0
RapidMiner	<a href="https://rapidminer.com/">https://rapidminer.com/</a>	RapidMiner is a data science software platform supporting the extraction and preprocessing of data, data visualization and data analysis, and predictive analytics, i.e., statistical, mathematical, and machine learning methods. The RapidMiner platform offers <i>RapidMiner Studio</i> , a client-based software for visual programming and execution of analysis processes as well as an <i>AI Hub</i> as a server solution to execute analysis processes in a shared environment.	RapidMiner is available as a free version (limited to 10,000 rows in a dataset) or a free academic version without limitations.
KNIME	<a href="https://www.knime.com/">https://www.knime.com/</a>	Open-source analytics platform for data science projects based on visual computing. Build workflows with drag and drop, no coding needed. See also the KNIME Academic Alliance.	Free—GNU v.3 KNIME Server for business from €12,5k

(continued)



(continued)

<b>Machine Learning</b>				
Name	Website	Description	Costs	Distribution
WEKA	<a href="https://www.cs.waikato.ac.nz/ml/weka/">https://www.cs.waikato.ac.nz/ml/weka/</a>	A collection of machine learning algorithms for data mining tasks, containing tools for data preparation, classification, regression, clustering, association rules mining, and visualization.	Free—GNU v.3	Download—Stand-alone
Shogun	<a href="https://www.shogun-toolbox.org/">https://www.shogun-toolbox.org/</a>	An open-source machine learning library, offering a wide range of machine learning methods; supports Python, R, Scala, Ruby, etc.	Free GPL v.3	Python library and other installation packages
Sci-kit Learn	<a href="https://scikit-learn.org/stable/">https://scikit-learn.org/stable/</a>	Scikit-learn is a free machine learning software library for the Python programming language. It offers various classification, regression, and clustering algorithms.	Free—3-Clause BSD license	R > 3.4.3, Uvot 0.1.10, Rtsne 0.15, scikit-learn 0.24.0
Auto-Sklearn	<a href="https://automl.github.io/auto-sklearn/master/">https://automl.github.io/auto-sklearn/master/</a>	Auto-Sklearn is an automated machine learning toolkit that relieves the user of algorithm selection and hyperparameter tuning.	Free—3-Clause BSD license	Python library
TensorFlow	<a href="https://www.tensorflow.org/">https://www.tensorflow.org/</a>	TensorFlow is a fast, flexible, and scalable open-source machine learning library offered by Google for research and production.	Free—Apache License 2.0	Python library
Keras	<a href="https://keras.io/">https://keras.io/</a>	Very popular open-source neural network library.	Free—MIT License	Python library
PyTorch	<a href="https://pytorch.org/">https://pytorch.org/</a>	Machine learning libraries developed by Facebook, often used for computer vision and NLP.	BSD license	Python library



Chapter 8			
Unsupervised Machine Learning—Clustering			
Name	Website	Description	Costs
RapidMiner Orange Knime WEKA Shogun Sci-kit Learn		See Chap. 6	
IBM SPSS	<a href="https://www.ibm.com/uk-en/analytics/spss-statistics-software">https://www.ibm.com/uk-en/analytics/spss-statistics-software</a>	IBM SPSS Statistics is a software package used for interactive, or batched, statistical analysis. The software name originally stood for Statistical Package for the Social Sciences but later changed to Statistical Product and Service Solutions.	IBM SPSS Statistics Standard; Authorized User; License + SW Subscription & Support 12 Months: 6.437,00€; Yearly renewal license: 1.293,00€
			Distribution
			Download—Stand-alone IBM SPSS as an authorized user, one needs to sign into the IBM Passport Advantage Online (PAO) website.

<b>Chapter 9</b>			
<b>Unsupervised Machine Learning—Dimensionality Reduction</b>			
Name	Website	Description	Distribution
RapidMiner			
Orange			
KNIME			
WEKA			
Shogun			
Sci-kit Learn		See Chap. 6	Costs

<b>Chapter 10</b>			
<b>Supervised Machine Learning—Classification</b>			
Name	Website	Description	Distribution
caret	<a href="https://github.com/topepo/caret">https://github.com/topepo/caret</a>	caret (Classification And Regression Training) is an R package that contains miscellaneous functions for training and plotting classification and regression models.	Free—GPL3 License
RapidMiner Orange KNIME WEKA Shogun		See <a href="#">Chap. 6</a>	

Chapter 11		Supervised Machine Learning—Regression		
Name	Website	Description	Costs	Distribution
RapidMiner Orange KNIME WEKA Shogun Sci-kit Learn caret		See Chap. 6		
		See Chap. 10		

<b>Chapter 12</b>				
<b>Hyperparameter Tuning</b>				
Name	Website	Description	Costs	Distribution
Hyperopt	<a href="https://github.com/hyperopt/">https://github.com/hyperopt/</a>	Hyperopt is a Python library designed for either serial or parallel optimization. It supports different types of search spaces, such as real-valued, discrete, and conditional dimensions.	Free—MIT License	Python library
Optuna	<a href="https://github.com/optuna">https://github.com/optuna</a>	Optuna is an automatic hyperparameter optimization Python framework defined by a high-modularity style user API.	Free—MIT License	Python library
TPOT (Tree-based Pipeline Optimization Tool)	<a href="https://github.com/EpistasisLab/tpot">https://github.com/EpistasisLab/tpot</a>	TPOT is a Python automated machine learning library designed for optimizing machine learning pipelines by taking advantage of genetic programming.	Free—GNU v. 3.0	Python library
SAS Viya	<a href="https://www.sas.com/en_us/software/viya.html">https://www.sas.com/en_us/software/viya.html</a>	SAS Viya is a cloud-based data analytics framework designed to help organizations speed up their data analytics capabilities (from data collection to data deployment).	Academic and student free licenses available	Cloud-based service

<b>Chapter 13</b>		<b>Model Evaluation &amp; Overfitting</b>	
Name	Website	Description	Costs
RapidMiner		See Chap. 6	
Orange			
KNIME			
WEKA			
Shogun			
Sci-kit Learn			Distribution



Chapter 14		Data Interpretability of ML-Models		
Name	Website	Description	Costs	Distribution
SHAP	<a href="https://github.com/slundberg/shap">https://github.com/slundberg/shap</a>	Most widely used framework for model interpretation. Can be used for any type of model. Basic coding skills required.	MIT license	Python library or R package
LIME	<a href="https://github.com/marcotcr/lime">https://github.com/marcotcr/lime</a>	A widely used framework for model interpretation. Can be used for any type of model. Local interpretations only. Basic coding skills required.	Free modification and use keeping the same copyright	Python library or R package
ELI5	<a href="https://github.com/TeamHG-Memex/eli5">https://github.com/TeamHG-Memex/eli5</a>	An easy-to-use framework for model interpretation through linear model approximation. Can be used for any type of model. Local & global interpretations. Basic coding skills required.	Free modification and use keeping the same copyright	Python library
Alibi	<a href="https://github.com/SeldonIO/alibi">https://github.com/SeldonIO/alibi</a>	A library grouping algorithm for model interpretation. Basic coding skills required.	Free, custom Apache license	Python library
DATAIKU	<a href="https://www.dataiku.com/product/">https://www.dataiku.com/product/</a>	All-in-one data science platform. Contains a module for model interpretability. No coding required.	Billing on demand	Cloud platform
AZURE ML	<a href="https://azure.microsoft.com/en-us/services/machine-learning/">https://azure.microsoft.com/en-us/services/machine-learning/</a>	Enterprise-grade machine learning service to build and deploy models faster. Interpretation module integrated. Basic coding skills required.	Pay-as-you-go	Cloud platform
Captum	<a href="https://captum.ai/">https://captum.ai/</a>	Interpretation framework of deep learning models with PyTorch. Advanced coding skills required.	Free, BSD 3-Clause License	Python library

tf explain	<a href="https://github.com/sicara/tf-explain">https://github.com/sicara/tf-explain</a>	Interpretation framework of deep learning models with TensorFlow. Advanced coding skills required. See <a href="#">Chap. 6</a>	MIT license	Python library
RapidMiner Orange				
Interpret ML	<a href="https://github.com/interpretml/interpret">https://github.com/interpretml/interpret</a>	A library grouping algorithm for model interpretation. Basic coding skills required.	MIT license	Python library
H2O	<a href="https://www.h2o.ai/">https://www.h2o.ai/</a>	An open-source data science and machine learning platform with a model interpretation module. No coding skills are required.	Open Software and Enterprise; Billing on demand	A Python library or R package; cloud service.
Data Robot	<a href="https://www.datarobot.com/platform/">https://www.datarobot.com/platform/</a>	DataRobot's Enterprise all-in-one AI platform. No coding skills are required.	Billing on demand	Cloud service
Craft ai	<a href="https://www.craft.ai/">https://www.craft.ai/</a>	API-enabling product & operational teams to quickly deploy and run explainable AIs. Basic coding skills.	Billing on demand	API

<b>Chapter 15</b>		<b>Introduction: Natural Language Processing</b>	
Name	Website	Description	Costs
RapidMiner Orange KNIME		See Chap. 6	
Voyant Tools	<a href="https://voyant-tools.org/">https://voyant-tools.org/</a>	Voyant Tools is an easy-to-use, web-based text reading and analysis environment.	Free—GPL3 License
NLTK	<a href="https://www.nltk.org/">https://www.nltk.org/</a>	One of the leading Python platforms to perform NLP tasks such as tokenizing, POS, sentiment analysis, topic segmentation, NER, etc.	Free
spaCy	<a href="https://spacy.io/">https://spacy.io/</a>	Industrial-Strength NLP Platform with 64+ languages, many trained pipelines, pre-trained word vectors, NER, POS, text classification, dependency parsing, and much more.	Free
NLU	<a href="https://github.com/JohnSnowLabs/nlu">https://github.com/JohnSnowLabs/nlu</a>	Python library for state-of-the-art text mining on any data frame. Easy to implement with a single line of code. Comes with 1000+ pre-trained models and many utilities for NLU applications.	Free—Apache License 2.0
CoreNLP	<a href="https://stanfordnlp.github.io/CoreNLP/">https://stanfordnlp.github.io/CoreNLP/</a>	One-stop shop for NLP, using Java to perform POS, NER, sentiment, etc. Supports Arabic, Chinese, English, French, German, and Spanish.	Free—GNU 3 license
TextRazor	<a href="https://www.textrazor.com/">https://www.textrazor.com/</a>	Complete cloud or self-hosted text analysis infrastructure for entity and key phrase extraction, topic tagging and classification, disambiguation and linking, etc. in 12 languages.	Commercial; Offering increased free limits and special pricing for academic users.
MITAO	<a href="https://github.com/catars/mitao">https://github.com/catars/mitao</a>	MITAO is a user-friendly, modular, and flexible software written in Python and Javascript for performing text analysis, and can be run locally on a machine by using any modern Web browser	Free—ISC License
			Distribution
			Web-based
			Python library
			Python library
			Python library
			Download—Stand-alone
			SDKs for Python, PHP, Java
			Download—Stand-alone (based on Python)

<b>Chapter 16</b>			
<b>Text Representation and Word Embeddings</b>			
Name	Website	Description	Costs
Gensim	<a href="https://radimrehurek.com/gensim">https://radimrehurek.com/gensim</a>	Gensim is a Python library used to represent documents as semantic vectors. It processes unstructured text data using unsupervised ML. Gensim provides algorithms such as word2vec, doc2vec, Fasttext, LDA, LSI, etc.	Free—GNU LGPLv2.1 license
NLU	<a href="https://github.com/JohnSnowLabs/nlu">https://github.com/JohnSnowLabs/nlu</a>	Python library with 100 of the latest word embeddings (BERT, ELMo, Glove, Electra, XLNET, etc.).	Free—Apache License 2.0
Glove	<a href="https://nlp.stanford.edu/projects/glove/">https://nlp.stanford.edu/projects/glove/</a>	An unsupervised learning algorithm for obtaining vector representations for words.	Free—Apache License 2.0
Elmo	<a href="https://allenmp.org/elmo">https://allenmp.org/elmo</a>	Deep contextualized word representations.	Free—Apache License 2.0
Bert	<a href="https://github.com/google-research/bert">https://github.com/google-research/bert</a>	State-of-the-art, transformer-based machine learning algorithms for NLP. Pre-trained by Google.	Free—Apache License 2.0
Huggingface	<a href="https://huggingface.co/">https://huggingface.co/</a>	Huggingface provides a huge number of pre-trained models (transformers, including BERT, GPT-2, RoBERTa, DistilBERT, XLNet, XLM, etc.) as open-source libraries, which can be implemented with one line of code. These can be used for tasks such as sentiment analysis, NER, Question Answering, text summarization, feature extraction, etc.	Free—Apache License 2.0

Chapter 17			Sentiment Analysis	
Name	Website	Description	Costs	Distribution
MeaningCloud	<a href="https://www.meaningcloud.com/">https://www.meaningcloud.com/</a>	MeaningCloud provides a public API for multiple categories of text processing including sentiment analysis. The sentiment analysis tool is able to detect global, local, and aspect sentiments within texts in multiple European languages. Notably, irony and polarity disagreement are detected. The API can be used with a variety of products including Excel, Rapidminer, Google Sheets, etc.	Free (up to 20,000 requests monthly); for a fee.	Download—Stand-alone
LJWC (Linguistic Inquiry and Word Count)	<a href="https://liwc.wpengine.com/">https://liwc.wpengine.com/</a>	LJWC is a popular lexicon-based package in the social sciences used for comprehensive text analysis, including sentiment analysis. One advantage of the package is the easy-to-use graphical interface. The analysis is possible in multiple languages including English, Chinese, Arabic, Spanish, Dutch, French, German, Italian, Russian, and Turkish.	Subscription fee starts at \$9.95 monthly for academics.	Download—Stand-alone
SentiStrength	<a href="http://sentistrength.wlv.ac.uk/">http://sentistrength.wlv.ac.uk/</a>	Lexicon-based sentiment analysis originally optimized for short social network texts. SentiStrength provides a wide variety of sentiment classification types on a document, sentence, and aspect level, even though the latter option returns only fair results in our experience. The base dictionary is for the English language and customized for multiple domains. In addition, user-supplied dictionaries in many other languages are available for download (stand-alone), yet sentiment recognition accuracy varies dramatically between those additional lexicons. The base software does not require programming competencies; however, a Java version with extended capabilities is available and can be used in conjunction with different popular packages and general-purpose programming languages including Weka and Python.	Free for academic use.	Download—Stand-alone package or analyze online.

<p>RapidMiner Orange KNIME</p>	<p>See <a href="#">Chap. 6</a></p>	<p>Free; for a fee.</p>	<p>Included in many libraries for Python, Java, R, RapidMiner, and other software environments. Also a part of the popular NLTK (Natural Language Toolkit) Python library.</p>
<p>VADER (Valence Aware Dictionary for Sentiment Reasoning)</p>	<p><a href="https://github.com/cjhutto/vaderSentiment">https://github.com/cjhutto/vaderSentiment</a> and many others</p>	<p>Lexicon-based sentiment analysis combining existing sentiment word-banks, such as LIWC with slang, emoticons, acronyms, and initialisms (e.g., WTF), typically found in social media. The sentiment word list is carefully curated using Amazon Mturks; in addition, a set of rules developed from analyzing representative tweets is used to estimate sentiment intensity. Accordingly, the algorithm works best with Twitter messages, but users report good performance with a variety of social media platforms.</p>	<p>Free</p>
<p>NLTK SentimentAnalyzer</p>	<p><a href="https://www.nltk.org/api/nltk.sentiment.html">https://www.nltk.org/api/nltk.sentiment.html</a></p>	<p>NLTK (The Natural Language Toolkit) is a large library for natural language processing targeted at building new computational linguistic software. It includes multiple sentiment analysis methods. While more specialized packages such as Spacy exist, NLTK is arguably the most flexible one. The NLTK project is well documented and has a large and active community, making the platform ideal for teaching text analysis with Python programming. Programming skills required.</p>	<p>Free</p>
<p>SentiWordNet</p>	<p><a href="https://github.com/aesuli/SentiWordNet">https://github.com/aesuli/SentiWordNet</a></p>	<p>SentiWordNet lexicon is developed from Princeton’s WordNet lexical database of the English language. While WordNet includes over 100,000 sets of words denoting similar concepts (synsets), SentiWordNet extends the synsets by assigning the positivity, negativity, and objectivity scores. Several benchmark comparisons have shown that SentiWordNet’s performance is moderate, but its lexicon has been adapted and extended further in other software programs such as VADER.</p>	<p>A part of the popular NLTK (Natural Language Toolkit) Python library.</p>

(continued)

(continued)

Sentiment Analysis			
Name	Website	Description	Costs
AFINN	<a href="https://github.com/fnielsen/afinn">https://github.com/fnielsen/afinn</a>	AFINN is a lexicon-based sentiment analysis software with a relatively small (3300 words) sentiment lexicon, manually trained by Arup Nielsen Finn. The sentiment is measured using a scale from -5 (very negative) to 5 (very positive). The lexicon originates from the ANEW (Affective Norms for English Words) sentiment lexicon and is optimized for Twitter posts and microblogs. Notably, the lexicon includes slang (e.g., “haha” scores +3) and multiword phrases. It is also updated on a regular basis to reflect the ongoing progress in online discourse; e.g., a recent new entry includes “f*ing cute,” while the negativity score of “damn” has recently been downgraded from -4 to -2. Benchmarking indicates a good performance of FINN on tweets and a moderately good performance when it comes to BBC and New York Times comments. Implementation requires programming experience.	Free
			Distribution Python library

<b>Chapter 18</b>				
<b>Topic Modeling</b>				
Name	Website	Description	Costs	Distribution
RapidMiner Orange KNIME		See Chap. 6		
InPhO Topic Explorer	<a href="https://www.hypershelf.org/">https://www.hypershelf.org/</a>	Interactively explore and interpret document collections	Free—MIT License	Source code on GitHub
Mallet	<a href="http://mallet.cs.umass.edu/topics.php">http://mallet.cs.umass.edu/topics.php</a>	Machine Learning for language toolkit based on Java. Python wrapper available.	Free—CPL-1.0	Source code for download
Gensim		See Chap. 16		
BERTopic	<a href="https://github.com/MaartenGr/BERTopic">https://github.com/MaartenGr/BERTopic</a>	BERTopic is a topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics while keeping important words in the topic descriptions.	Free—MIT License	Python library
Top2Vec	<a href="https://github.com/ddangelov/Top2Vec">https://github.com/ddangelov/Top2Vec</a>	Top2Vec is an algorithm for topic modeling and semantic search. It automatically detects topics present in texts and generates jointly embedded topic, document, and word vectors.	Free—BSD-3-Clause	Python library
CorEx	<a href="https://github.com/gregversteeg/corex_topic">https://github.com/gregversteeg/corex_topic</a>	Hierarchical unsupervised and semi-supervised topic models for sparse count data.	Apache 2.0	Python library
BigARTM	<a href="https://github.com/bigartm/bigartm">https://github.com/bigartm/bigartm</a>	BigARTM is a powerful tool for topic modeling based on a novel technique called Additive Regularization of Topic Models.	See Github	Python library
MITAO		See Chap. 15		
Blei's Topic Modeling List	<a href="http://www.cs.columbia.edu/~blei/topicmodeling_software.html">http://www.cs.columbia.edu/~blei/topicmodeling_software.html</a>	A collection of topic modeling software from Dabid Blei.		



<b>Chapter 19</b>		<b>Entity Matching</b>		
Name	Website	Description	Costs	Distribution
DeepMatcher	<a href="https://github.com/anhaidgroup/deepmatcher">https://github.com/anhaidgroup/deepmatcher</a>	DeepMatcher allows building end-to-end entity matching pipelines. It utilizes various types of neural networks and recent advances in the field of NLP to build state-of-the-art entity matching solutions.	Open-source	Python package
Py_entitymatching	<a href="https://github.com/anhaidgroup/py_entitymatching">https://github.com/anhaidgroup/py_entitymatching</a>	Py_entitymatching is a framework to build entity matching pipelines that utilize supervised learning. It has a very powerful candidate generation component that can be used separately.	Open-source	Python package
Dedupe	<a href="https://github.com/dedupeio/dedupe">https://github.com/dedupeio/dedupe</a>	Dedupe is a framework for entity matching with an emphasis on record deduplication.	Open-source	Python package

<b>Chapter 20</b>		<b>Knowledge-Graphs</b>		
Name	Website	Description	Costs	Distribution
SpaCy		See Chap. 15		
PyPi Edit Distance	<a href="https://pypi.org/project/editdistance/">https://pypi.org/project/editdistance/</a>	A fast Python implementation of the edit distance (also known as the Levenshtein distance).	Free	Python library
OpenKE	<a href="http://openke.thunlp.org/">http://openke.thunlp.org/</a>	An open-source framework for knowledge graph embeddings/representation learning implemented with PyTorch.	Free to use; no license information provided	Source available on GitHub
DIG	<a href="https://github.com/usc-isi-i2/dig-etl-engine">https://github.com/usc-isi-i2/dig-etl-engine</a>	DIG is an end-to-end configurable system for constructing knowledge graphs from Web data.	Free, MIT License	Available to pull as a docker container, with source provided on website.
Computing Precision-Recall with Scikit-Learn	<a href="https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html">https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html</a>	The package allows users to compute precision, recall, and other accuracy metrics, including a precision-recall tradeoff curve, to assess the quality of predictions.	Free	Download—Stand-alone
Schema Tourism Working Group	<a href="https://schema-tourism.sti2.org/">https://schema-tourism.sti2.org/</a>	The mission of the Schema Tourism Working Group is to provide reference vocabularies for the touristic domain.	Resources on the website seem to be available for free.	Copyright belongs to the Schema Tourism Working Group, but otherwise, resources can be downloaded directly from the website.
TouristDestination Concept from Schema.org Vocabulary	<a href="https://schema.org/TouristDestination">https://schema.org/TouristDestination</a>	A concept used to represent a tourist destination and that can be used as markup on any website using schema.org.	No cost associated	As this is a Schema.org “Type,” it is not distributed; instead, website developers can directly incorporate this term and its predicates (described on the website) on their website to describe tourist destinations.

(continued)



<b>Chapter 21</b>			
<b>Social Network Analysis</b>			
Name	Website	Description	Costs
Social network analysis software	<a href="https://en.wikipedia.org/wiki/Social_network_analysis_software">https://en.wikipedia.org/wiki/Social_network_analysis_software</a>	Long list of network analysis software applications and libraries.	
Gephi	<a href="https://gephi.org/">https://gephi.org/</a>	Visualization and calculation of basic network measures.	Free
Ucinet	<a href="https://sites.google.com/site/ucinetsoftware/home">https://sites.google.com/site/ucinetsoftware/home</a>	Calculation of basic network measures.	Commercial
Pajek	<a href="http://mrvar.fdv.uni-lj.si/pajek/">http://mrvar.fdv.uni-lj.si/pajek/</a>	Visualization and calculation of basic network measures.	Free
Python language and libraries (Networkx, python-igraph)	<a href="https://www.python.org/">https://www.python.org/</a>	Complete libraries for visualization, calculations, and dynamic modeling.	Free
R libraries (igraph, sna, tnet, etc.)	<a href="https://www.r-project.org/">https://www.r-project.org/</a>	Complete libraries for visualization, calculations, and dynamic modeling.	Free
Matlab toolboxes	<a href="https://www.mathworks.com/">https://www.mathworks.com/</a>	Complete libraries for visualization, calculations, and dynamic modeling.	Commercial (expensive); many toolboxes, however, are free.
			Distribution
			Download from website. Multipatform.
			Download from website. Windows only (can run on Mac or Linux via Wine).
			Install a distribution such as Anaconda <a href="https://www.anaconda.com/">https://www.anaconda.com/</a> , a freely available compilation of Python language and libraries that includes Networkx and all the dependencies needed. Python-igraph, if needed, must be added separately (can be installed using: "conda install -c conda-forge python-igraph").
			All libraries are available on one of the "The Comprehensive R Archive Network (CRAN)" mirrors: <a href="https://cran.r-project.org/mirrors.html">https://cran.r-project.org/mirrors.html</a> .
			Academic licenses or other facilities may exist depending on the singular institutions. Many toolboxes for network analysis exist (generally free). They can be located with a Google search (e.g., "matlab network analysis toolbox").

<b>Chapter 22</b>		<b>Time Series Analysis</b>		
Name	Website	Description	Costs	Distribution
EViews	<a href="https://eviews.com/home.html">https://eviews.com/home.html</a>	EViews is a statistical software designed for time-series forecasting and modeling. It is a highly interactive program and provides detailed data analysis in terms of building and assessing models, conducting residuals analysis, and examining hypotheses with either univariate or multivariate variables (Agung, 2011). No programming knowledge is needed since it has a user-friendly GUI. The website also provides various examples and offers a help section for questions.	Academic licensing as well as a student edition is offered. A free EViews 10 Student Version Lite is available for students.	Software to download
Python	<a href="https://www.python.org/">https://www.python.org/</a>	Python features productive libraries and packages at no cost. For example, Python has “fbprophet,” “pmdarima,” and “tsfresh,” all of which can be used for forecasting.	Free	Software to download
R	<a href="https://www.r-project.org/">https://www.r-project.org/</a>	R features productive libraries and packages at no cost. For example, R has the “ts,” “forecast,” and “smooth” packages, all of which can be used for forecasting.	Free	Software to download
Prophet	<a href="https://facebook.github.io/prophet/">https://facebook.github.io/prophet/</a>	Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It is well suited for tourism-related data, as it works best with time series that have strong seasonal effects and several seasons of historical data.	Free—MIT License	Python library R library

SAS	<a href="https://www.sas.com/en_us/home.html">https://www.sas.com/en_us/home.html</a>	SAS is an advanced statistical analytics software designed to accomplish time series analysis effectively and provide excellent visualization of time series analysis results (Brocklebank & Dickey, 2003).	Free trial: 30 days; Entry costs to license the most basic package (SAS Analytics Pro): \$8700 (first year fee) through the SAS online store.	Both cloud service and software to download available.
MATLAB	<a href="https://www.mathworks.com/products/matlab.html">https://www.mathworks.com/products/matlab.html</a>	MATLAB provides time series functions that enable users to explore the dynamics of an arbitrarily large set of time series with a variety of measurement and numerical computations (Kugiumtzis & Tsimpiris, 2010). See <a href="#">Chap. 6</a>	Free trial: 30 days; Student license: \$49	Software to download
RapidMiner Orange KNIME				

Chapter 23		Agent-based Modeling		
Name	Website	Description	Costs	Distribution
<i>Cormas (Common-pool Resources and Multi-Agent Simulations)</i>	<a href="http://cormas.cirad.fr/">http://cormas.cirad.fr/</a>	Natural resource simulations in combination with companion modeling. Moderate difficulty. Medium scale. See Abar et al. (2017) for details.	Open source and free following the Cormas user charter.	Online through website
<i>Envision</i>	<a href="http://envision.bioe.orst.edu/">http://envision.bioe.orst.edu/</a>	Spatially explicit platform for human-environment interactions. Microsoft Visual C++ Moderate programming difficulty. Medium scale. See Abar et al. (2017) for details.	Open source and free	Online through website
<i>NetLogo 6.0.4.</i>	<a href="http://ccl.northwestern.edu/netlogo/">http://ccl.northwestern.edu/netlogo/</a>	Scala code and is fully interoperable with Java and other JVM languages. Easy programming tool for medium-scale simulations. See Abar et al. (2017) for details.	Open source and free	Online through website
<i>Repast</i>	<a href="https://repast.github.io/index.html">https://repast.github.io/index.html</a>	Java programming language. High programming difficulty. High scale. See Abar et al. (2017) for details.	Open source and free	Online through website
CoMSES OpenABM	<a href="https://www.comses.net/">https://www.comses.net/</a>	An extensive ABM repository and forum.	Free	
<i>Exploratory Modeling and Analysis (EMA) Workbench</i>	<a href="https://emaworkbench.readthedocs.io/en/latest/">https://emaworkbench.readthedocs.io/en/latest/</a>	Supports tools for designing and performing experiments of complex and uncertain systems and provides connectors to <i>NetLogo</i> , <i>Vensim</i> , and Excel.	Free	Varies depending on software choice
<i>PyNetLogo</i>	<a href="https://pynetlogo.readthedocs.io/en/latest/">https://pynetlogo.readthedocs.io/en/latest/</a>	Interface to access <i>NetLogo</i> from <i>Python</i> . The library can load models, execute commands, and get values from reporters.	Free	Python library
<i>SALib—Sensitivity Analysis Library in Python</i>	<a href="https://salib.readthedocs.io/en/latest/#">https://salib.readthedocs.io/en/latest/#</a>	Sensitivity analysis tools executed in <i>Python</i> .	Free	Python modeling library
Stackoverflow NetLogo ABM-related questions	<a href="https://stackoverflow.com/questions/tagged/netlogo">https://stackoverflow.com/questions/tagged/netlogo</a>	Forum for sharing information on <i>NetLogo</i> .	Free	

<b>Chapter 24</b>		<b>Data Visualization</b>		
Name	Website	Description	Costs	Distribution
Bokeh	<a href="https://bokeh.org/">https://bokeh.org/</a>	Bokeh is a data visualization library based on Python.	BSD 3-Clause	Python library
Plotly	<a href="https://plotly.com/">https://plotly.com/</a>	Used for building Dash apps.	Free; Enterprise Version	Python, R, Julia
Matplotlib	<a href="https://matplotlib.org/">https://matplotlib.org/</a>	Matplotlib is a data visualization library based on Python.	Free	Open Source
Seaborn	<a href="https://seaborn.pydata.org/">https://seaborn.pydata.org/</a>	Seaborn is a data visualization library based on Python.	Free	Open Source
Tableau	<a href="https://www.tableau.com/">https://www.tableau.com/</a>	Tableau is a data visualization and business intelligence application for creating interactive plots and dashboards.	Commercial; free licenses for students available	Cloud-based service



<b>Chapter 25</b>		<b>GIS Analysis</b>		
Name	Website	Description	Costs	Distribution
ESRI ArcGIS	<a href="https://www.esri.com/index.html">https://www.esri.com/index.html</a> <a href="https://www.esri.com/en-us/home">https://www.esri.com/en-us/home</a>	ArcGIS (which comprises a multitude of products, extensions, and licenses) is the most well-known and arguably the most widely-used commercial GIS software on the market. Within this range of products, the ArcGIS Pro desktop software provides excellent map-making features, including 3D visualization, time animation, and online publishing. Yet, there are also multiple and diverse spatial analysis tools. Taking advantage of ArcGIS' flexibility, one can easily adapt its functionality into Python using ArcGIS API or expand an ArcGIS toolbox using Python scripts (note, however, that ArcGIS Pro uses Python v. 3, while other products use Python v. 2, making the scripts non-interchangeable). For field data collection, including street surveys, an app can be installed on a tablet or mobile phone.  The software comes with access to a wide range of online data, including the data contributed by the ESRI community.	Not provided, but generally high (thousands of dollars) for a meaningful set of tools. A basic license provides severely limited functionality. Student license: \$100/yr.	Stand-alone and online.
QGIS	<a href="https://qgis.org/">https://qgis.org/</a>	QGIS is an extremely popular open-source GIS. QGIS' capabilities, at least for academic research, are more or less similar to ArcGIS' described above. Similar to ArcGIS, there are also QGIS-compatible data collection apps for iOS and Android, allowing for easy field data collection. There are also many third-party add-ons. For process automation or to add missing tools, QGIS can be integrated with R or Python. When comparing the usability of both packages, reviewers typically note a gentler learning curve for QGIS. As a downfall, however, QGIS plugins may be hard to implement, and it is also more prone to crash, especially during 3D processing.	Free	Download stand-alone package.
GeoDa	<a href="https://spatial.uchicago.edu/software">https://spatial.uchicago.edu/software</a>	GeoDa is not considered a full-fledged GIS as its toolset is quite limited. Some, however, consider this to be its strong point, making it popular for teaching. GeoDa has an intuitive graphic interface and is easy to learn; meanwhile, it still includes spatial statistics tools for exploratory data analysis and spatial regression.	Free	Download stand-alone package.

<p>R</p>	<p><a href="https://www.r-project.org/">https://www.r-project.org/</a></p>	<p>R is a leading open-source software environment for statistical computing and graphics, which also includes the GIS functionality. The main advantage of carrying out geoprocessing with R is probably its superior capability of working with very large datasets as well as its versatile and flexible set of tools for complex quantitative analysis. The learning curve is, however, steep, and there is no intuitive graphic interface such as with QGIS and ArcGIS. Instead, operations are completed through a command-line interface. One way around this problem is the integration of R code into QGIS, ArcGIS, or another GIS that supports R scripts. For more detail, read the free online book by Lovelace et al. (2021).</p>	<p>Free</p>	<p>Download stand-alone package.</p>
<p>Geopandas and other Python geospatial analysis packages</p>	<p><a href="https://geopandas.org/">https://geopandas.org/</a></p>	<p>While Python scripts can be used with leading GIS programs, there are multiple ways to work with geospatial data using only Python geoprocessing packages. The most notable Python packages for geoprocessing include GeoPandas (for data processing and visualization), Rasterio (for working with rasters), and PySAL (for exploratory and confirmatory statistical analysis of spatio-temporal data, spatial econometrics, cluster analysis, and data visualization), among many more useful tools.</p>	<p>Free and/or for a fee (for some packages).</p>	<p>Download stand-alone package or analyze online.</p>
<p>SPSS</p>	<p><a href="https://www.ibm.com/products/spss-statistics">https://www.ibm.com/products/spss-statistics</a></p>	<p>The base version of the popular, especially among the social scientists, IBM SPSS statistical software now includes GIS capabilities. The geoprocessing tools are accessible through the Analyze &gt; Spatial and Temporal Modeling &gt; Spatial Modeling menu.</p>	<p>Contact your ESRI representative for licensing information.</p>	<p>Download stand-alone package or analyze online.</p>

## References

- Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. (2017). Agent based modelling and simulation tools: A review of the state-of-art software. *Computer Science Review*, 24, 13–33.
- Agung, I. G. N. (2011). *Time series data analysis using EViews*. Wiley.
- Brocklebank, J. C., & Dickey, D. A. (2003). *SAS for forecasting time series*. Wiley.
- Kugiumtzis, D., & Tsimpiris, A. (2010). Measures of analysis of time series (MATS): A MATLAB toolkit for computation of multiple measures on time series data bases. *arXiv preprint arXiv:1002.1940*.
- Lovelace, R., Nowosad, J., & Muenchow, J. (2021). *Geocomputing with R*. CRC Press. Available online: <https://geocompr.robinlovelace.net/>

# Glossary

**Accuracy** A metric for evaluating how often an algorithm classifies a data point correctly. It is the fraction of predictions a model got right.

**Adversarial attack** Adversarial machine learning is a machine learning technique that attempts to fool models by supplying deceptive input. The most common reason is to cause a malfunction in a machine learning model.

**Agency** The ability of an individual or being to autonomously act or make decisions.

**Agent** In its most basic form, an autonomous individual or entity can make decisions. More elaborate agents may be able to move, learn, and change strategies based on learnings.

**Agent-based model** A form of modeling that incorporates individual agents, a spatial setting, and agent-to-agent and agent-environment interactions.

**Algorithm** A method, function, or set of rules used to solve specific problems or to perform a computation. Examples include (linear, logistic, etc.) regression, random forest, support vector machines, and neural networks.

**Arousal** Active to passive state.

**Artificial Neural Networks (ANNs)** Artificial neural networks, or neural networks for short, are a set of algorithms that attempt to mimic the human brain in order to recognize the underlying relationships in a dataset.

**Area under curve (AUC)** The AUC quantifies the two-dimensional area under the ROC curve, thus providing a measure of performance for classification models.

**Attribute** In global information system, an attribute refers to non-spatial information about a geographic feature.

**Balanced accuracy** Classification accuracy with all classes weighted equally.

**Bias** Bias is an error from erroneous assumptions in the learning algorithm.

**Bidirectional encoder representations from a transformer.** A transformer-based machine learning model that was pre-trained by Google and has been used in a variety of natural language processing tasks, including question answering and

information extraction. It is more advanced than “classic” word embedding models like word2vec and GloVE.

**Black-box model** Black-box models are the modes that have observable input-output relationships but lack clarity around the inner workings.

**Candidate generation** The process of pre-filtering candidate record pairs to be compared in the entity matching pipeline.

**Categorical variables** These are variables with a discrete set of possible values. They can be ordinal (order matters) or nominal (order does not matter).

**Classification** Task of assigning objects to categories/classes.

**Classification accuracy** A score for estimating classification model quality that measures the proportion of correctly predicted/classified data instances.

**Classification error** Proportion of wrongly classified samples.

**Classifier** A classifier is a type of machine learning algorithm used to assign a class label to a data input.

**Clustering** Unsupervised task to divide data points into a number of groups, where data points in the same group (cluster) are more similar (in their characteristics) to each other than to those in other groups.

**Complex adaptive systems** Socio-ecological systems are characterized by ongoing interactions among individuals and their surrounding environment that are nonlinear and exhibit individual and/or spatial heterogeneity. These system characteristics pose challenges for modeling and developing policies as actions and solutions can have unexpected and undesirable consequences for other parts of the system.

**Confusion matrix** A matrix summarizing correctly and incorrectly predicted data instances for classification models.

**Corpus (plural corpora)** Collection of texts. Corpus generally refers to a structured set of textual data.

**Continuous variables** A variable whose value is determined by measurement and can take on a non-countable set of values. Values are defined by a number scale such as sales and income.

**Convergence** A state in which the loss between iterations changes only slightly during model training.

**Curse of dimensionality** The curse of dimensionality refers to a typical mathematical pitfall in relation to high-dimensional data analysis in which the normality assumptions of traditional frequentist statistics are violated and no meaningful model fitting can be established. It defines different problems that arise when using high-dimensional data. The dimensionality of a dataset is determined by the number of features available. Some of the problems related to working with high-dimensional data can appear when analyzing the data or creating a machine learning model. One of the main problematic issues is that, with an increasing number of features, an exponentially increasing amount of data is needed in order to avoid overfitting.

**Data leakage** Data leakage is when information from outside the training dataset is used to create the model.

- DBpedia** DBpedia is a dataset derived from Wikipedia infoboxes. As a central hub in the linked open data ecosystem developed within the Semantic Web, DBpedia's resources are linked to many other knowledge sources in linked open data, including GeoNames and the New York Times ontology.
- Deep Learning** Deep learning is a set of machine learning methods based on artificial neural networks with multiple layers to extract higher-level features from raw input.
- Dimension** The number of features (input variables) for a dataset (e.g., the number of feature vectors in an image).
- Document** A group of logically related content. In database terminology, the term "record" is often used. Depending upon the focus and goals, documents for topic modeling can significantly vary in length: a tweet, an email, a paragraph, a page, and an article.
- Dominance** Dominant to submissive state.
- Embedding** Low-dimensional representation of high-dimensional data.
- Emergence** The development of a phenomenon as the result of different (ongoing) interactions among agents or between agents and the environment.
- Emoticon (emoji)** Emoticon is an emotion icon. While an emoji is a picture, J, emoticon is its approximation using generic symbols :-).
- Emotion analysis** An aspect of natural language processing dealing with the detection of different types of feelings expressed in the texts, such as anger, sadness, or joy.
- Entity linking** Entity linking is the problem of automatically mapping entity extractions (typically obtained using a natural language processing system) to entities in a canonical knowledge base like Wikipedia. The problem is hard for various reasons, one of which is that distinct entities can sometimes be referred to using the same string label (e.g., "Wimbledon" the location versus "Wimbledon" the tennis tournament).
- Entity matching** The process of finding records of the same real-world entity throughout one or more data sources.
- Entity resolution** Entity Resolution (ER) is the algorithmic problem of determining when two or more entities refer to the same underlying entity.
- Epoch** A hyperparameter that defines the number of times a learning algorithm works through the entire dataset.
- Error analysis** Study of the results of a trained model by an analyst with the purpose of improving the model and understanding the type of errors made.
- Explainability** The possibility of explaining the prediction of an algorithm from a technical point of view.
- Exploratory data analysis** Study of the raw input data to determine potential features and models that suit better. This is a typical first step in the data analysis process in which an analyst becomes familiar with unknown data via visualization of variables, plotting histogram distributions, and other exploratory methods without attempting to address a concrete research question or confirm a hypothesis.

- F1-score** The F1-score of two variables is the harmonic mean of those variables. It is typically applied to precision and recall in natural language processing and other artificial intelligence sub-communities where accuracy is an important performance metric. Since there is usually a trade-off between precision and recall, the F1-score helps to quantify such a trade-off.
- False Positive Rate (FPR)** A negative sample that is incorrectly classified as positive.
- False Negative Rate (FNR)** A positive sample that is incorrectly classified as negative.
- Feature** A single column of data, being a component of observation. Also called an attribute.
- Feature engineering** Piece-wise improvement of input data to improve machine learning results.
- Feature selection** Process of reducing features by filtering the most important features.
- First law of geography** “Everything is related to everything else, but near things are more related than distant things” by Waldo Tobler (1970).
- Generalization error** Expected error on future samples.
- Generative pre-trained transformer 3** Generative pre-trained transformer 3 (GPT-3) is an autoregressive language model that produces human-like text by using deep learning. It was created by OpenAI, an artificial intelligence research laboratory based in the United States.
- Geodatabase/spatial database** A database designed for spatial data.
- Geospatial data/spatial data** Data combining the attributes with locational information (usually coordinates).
- Global information system vector data** Representation of geographical features in a form of discrete points, lines, and polygons.
- Global explanations** Global explanations explain the general decision-making of the model.
- Google Colab** Colab allows one to write and execute Python in his/her browser. It requires no configuration, provides graphics processing units (GPU) and random access memory (RAM), and offers easy sharing opportunities.
- Geographically weighted regression** A linear regression modified to model relationships that change spatially.
- Heterogeneity** Differences in characteristics or states of individuals (agents in agent-based modeling) and the environmental setting.
- High-dimensional data** High-dimensional data refers to a particular (difficult) type of data in which the number of samples/observations ( $n$  parameter) is lower than the number of the data’s descriptive variables/attributes ( $p$  parameter), i.e.,  $p > n$  or  $p \gg n$ .
- Hot spot analysis** Identification of spatial clusters with high values (hot spots) and low values (cold spots).
- Hyperparameters** All the different parameter values that can be arbitrarily defined by a user before training a model.

- Imputation** Replacement of missing values in raw data with generic, and correct, values.
- Information extraction** The problem of extracting relevant pieces of information, such as named entities, relations, and events, from a span of text.
- Instance** A single row of data, being an observation from the domain.
- Interaction effect** Interaction effects occur when the effect of one variable depends on the value of another variable.
- Interpretability** Interpretability can be defined as the ability to explain or provide meaning in terms that are understandable to a human being.
- Inverse logit** → see Sigmoid
- K-nearest neighbors** K-nearest neighbors (KNN) is an algorithm from supervised machine learning classification that uses Euclidean distance to project query data points on a reference “atlas” by computing similarities between query and reference data points.
- K-means** Popular algorithm from cluster analysis that is best capable of working with data processing spherical symmetry. K-means uses Euclidean distance for establishing similarities between data points.
- Knowledge graph** A graph-theoretic way of representing entities, relations, events, facts, and other kinds of “knowledge” for machines. Typically, this is accomplished either through a directed, labeled, multi-relational graph, or as a set of triples. A knowledge graph may also be referred to as a knowledge base in some communities.
- Laplacian eigenmap** An interesting hybrid of linear and nonlinear dimensionality reduction techniques that implements both neighbor graph construction and matrix factorization.
- Leave-one-out** Model evaluation technique in which the model is trained on all but one of the instances and the remaining instance is used for testing.
- Lemmatization** Replacement of the inflected forms of a word with its dictionary form (lemma).
- Lexicon** A component of a natural language processing system that contains information about individual words or word strings, vocabulary of a language, or a subject.
- Local explanations** Local explanations refer to explanations concerning a particular prediction. For instance, local explanations answer the question: why has the model predicted this particular value for this customer or entry?
- Loss** Loss is the penalty for a bad prediction. This number indicates how poor a model’s prediction was on a single example. A perfect prediction has a loss value of zero, otherwise, the loss is greater.
- Machine Learning (ML)** ML is a branch of artificial intelligence (AI) and is based on the idea that machines can learn from data with minimal human intervention.
- Mean absolute error** A score for estimating regression model quality that measures the average of the sum of deviations from true values.
- Mean absolute percentage error** Mean of deviations between predictions and true target values relative to true target values.



- Mean squared error** A score for estimating regression model quality that measures the average square deviation from true values.
- Misclassification rate** Classification error.
- Missing values imputation** Method of guessing/predicting values that are missing in a data table.
- Mobility** The movement from one position to another both within a location or tourism destination as well as between destinations or locations.
- Model** A concise, actionable, and predictive representation of the system or phenomenon created to meet a specific goal. Models can be broadly divided into descriptive and analytical. Descriptive models rely on simulations and are often the basis for numerical simulations. Analytical models are made up of sets of equations describing the characteristics and behavior of a system.
- Multidimensional scaling** Linear dimensionality reduction technique that uses numeric optimization of quadratic distance between original data and its low-rank approximation. Multidimensional scaling (MDS) can be viewed as a machine learning interpretation of principal components analysis.
- Named entity recognition** A popular task in the natural language processing community and is the problem of developing an algorithm for extracting mentions of named entities (such as instances of people, locations, geopolitical entities, and so on) with high accuracy.
- Neo4j** Neo4j is a graph database containing its own query language that is prominently used in enterprise-scale and commercial applications.
- Network** A system composed of interconnected elements. It can be rendered graphically (graph) by using a series of points (vertices or nodes) linked by lines (edges or links). The nodes in a network can represent simple objects (a person in a friendship network) or complex entities (a firm or a website). A link indicates some type of relationship between two nodes. This can be an information exchange, a chemical reaction, a force, etc. Links can be symmetric or directed (a trip from one place to another) and can be assigned a weight for measuring strength, importance, or a value. Many measures of individual and global features can be calculated for characterizing the different configurations. These are mostly rooted in the mathematical discipline of graph theory.
- Natural language processing** Natural language processing is a subfield of mathematical linguistics, artificial intelligence, and computer sciences dealing with a computer's understanding of the human language.
- Normalized discounted cumulative gain** A popular metric in communities like information retrieval, measuring the quality of a "ranked" list of items retrieved in response to a query, assuming the relevance of the items is known. It ranges from 0.0 to 1.0 and, unlike competing metrics such as mean reciprocal rank, can work with nonbinary relevance scores of items as well as multiple relevant items (for a query).
- Noise** The signal is the information that allows the model to generalize to new situations and what we are interested in. Noise refers to all non-informative signals that are perceived as a disturbance.

- Nuisance variations** Variations in the data that are unrelated to the problem addressed by the machine learning model.
- Numerical simulation** A computation, typically run as a computerized algorithm, that implements a model for a system. Numerical simulations are used to study the behavior of systems in situations where analytical models are too complex to provide analytical solutions, as in most nonlinear systems, or when real-life experiments are not feasible for theoretical or practical reasons, as in the case of social and economic systems.
- Observation** The active acquisition of information from a primary source. See also → instances
- Outlier** A data point that differs significantly from other observations.
- Overfitting** A problem that occurs when models fit too precisely to the training data at the expense of generalization.
- Parameters** Model parameters are the different variables learnt while training a model. These types of parameters can include, for example, the weight values in Artificial Neural Networks (ANNs) and linear regression.
- Part-of-speech tagging** Assigning parts of speech to each word, e.g., “woman—common noun (Part-of-speech tag NN).”
- Perplexity** An important hyperparameter in t-distributed stochastic neighbor embedding (tSNE) that represents, approximately, an average number of nearest neighbors for each data point.
- Polarity** Sentiment orientation (positive, neutral, or negative) of a text.
- Posterior (class) probability** Probability that a given sample belongs to a certain class.
- Precision** Precision is a score for estimating classification model quality that measures the proportion of true positive instances among all positively predicted instances. It assesses the correctness of a system’s results against a ground truth.
- Preprocessing** Preparing a text for analysis through processes such as data cleaning, stemming, and tokenization.
- Principal component analysis** Linear dimensionality reduction technique based on the matrix factorization approach. Principal component analysis (PCA) can also be described as an eigen value decomposition problem formulated from linear algebra.
- Projection** In global information system, projection is the transformation between spherical and planar coordinate systems.
- Raster data** Representation of geographical features in the form of a grid with each cell of the grid associated with a specific geographical location.
- Recall** A score for estimating classification model quality that measures the proportion of correctly identified positive instances.
- Receiver operating characteristic curve (ROC curve)** Refers to a graphical representation of the ranking performance of classification scores. The curve plots the true positive rate against the false-positive rate for all possible thresholds.
- Regression** Task of assigning numbers to data objects.

- Regressor** A regressor is a name given to any variable in a regression model used to predict a response variable.
- Regularization** In the regularization process, information is added to prevent overfitting or to solve an ill-posed problem. Techniques include Ridge Regression (L2), Lasso (L1), and Dropout.
- Reinforcement learning** Reinforcement learning is a type of machine learning technique that enables an agent to learn by trial and error in an interactive environment using feedback from its actions and experiences.
- Residual** Difference between prediction and true target value.
- Resource description framework** A model for representing and publishing structured data on the Web as sets of triples, where a triple includes the form of <subject, predicate, object>. The triples may be thought of as edges in a knowledge graph.
- Root mean square error** A score for estimating regression model quality that measures the square root of average square deviation from true values.
- Scenario discovery** A form of analysis similar to backcasting in which the outputs are defined in the form of a threshold. The analysis looks at which input variables are most influential in exceeding the threshold.
- Semantic Web** A field within artificial intelligence and computer science that attempts to make Internet data machine readable by extending the World Wide Web through standards set by the World Wide Web Consortium.
- Sensitivity** Sensitivity is a measure of how well a test can identify true positives. It can also be referred to as the recall, hit rate, or true positive rate.
- Sensitivity analysis** Analysis of how the variations in input parameters influence output responses. In agent-based modeling, the goals of sensitivity analysis are typically how emergent properties are generated, the robustness of emergent properties, and to quantify model outcome variations resulting from changes to input parameters.
- Sentiment analysis (opinion mining)** An aspect of natural language processing dealing with the extraction of opinions from a text.
- Sigmoid** Function used for transforming linear scores into probabilities.
- Signal** → see Noise
- Spatial autocorrelation** A measure of association between the measurements taken over increasing spatial intervals.
- Spatial interpolation** Prediction of the missing values over a larger area of interest.
- Specificity** A score for estimating classification model quality that measures the proportion of true negative instances among negative predicted instances.
- Stemming** Replacement of the inflected forms of a word with its word stem (base/root form).
- Stop word** An unwanted word, i.e., commonly used words that do not contain significant meaning for the study being conducted: is, and, the, that, but, can, etc. Custom stop word lists are often used for specific domains.
- Superpixel** A group of pixels that share common characteristics (like pixel intensity).

**Supervised Learning** In supervised tasks, labeled training data is used to learn a function based on these training examples.

**Surrogate model** An interpretable model that is trained to approximate the predictions of a black-box model.

**Switch** A switch feedforward neural network (FFN) layer that is more innovative than the traditional transformer architecture. Unlike many other models that came before it, switch trains a sparse model, leading to an increase in the number of model parameters without a proportionate increase in the amount of computation. It is measured in floating-point operations per second (FLOPS).

**System** A conceptual or real entity made of a number of elements interacting dynamically and generating some global behavior. Systems can be simple, complicated, or complex. Simple systems have few components with linear interactions and show predictable behaviors. Complicated systems contain a large number of components also with linear interactions. The global behavior can be (at least in principle) analyzed and derived as a superposition of the characteristics of some smaller parts. Complex systems are characterized by nonlinear interactions and feedback loops. They can become chaotic, and display high sensitivity to initial conditions as well as dynamic behavior that is adaptable to the environment. Emergent, self-organizing structures and behaviors typical of these systems cannot be derived as a composition of its elements' features and properties.

**Target value** The value that is or has been assigned to an object.

**Test Set** A dataset is used to provide an unbiased evaluation of a final model fit on the training dataset.

**Training Set** The training set is an initial set of examples used to fit the parameters of the model to the data.

**t-distributed stochastic neighbor embedding (tSNE)** tSNE is a nonlinear dimensionality reduction technique based on constructing a neighbor graph from the data with each data point laid out on the graph in a low-dimensional space.

**Token** An instance of a sequence of characters in some particular document that is grouped together as a useful semantic unit for processing. In natural language processing, each word is mostly referred to as a token.

**Tokenization** Breaks textual data down into atomic analysis units, for example, words and word combinations (n-grams).

**Topic** In natural language processing, topic refers to the theme of the words. When a collection of words occurs frequently, they create a theme, which, in turn, is called a topic. For example, breakfast, dinner, barbecue, and salad sauce can be considered the topic "meal." While finding hidden topics in a corpus, we usually initially examine (i.e., "name") a topic using the top 10 output words given by the unsupervised model. The researcher later labels these output topics by incorporating domain knowledge.

**Topology** The study of the intrinsic properties of an object or system due to its structural configuration. They are not modified by certain types of deformations or transformations that may radically change the geometric characteristics.

- Transfer Learning** The knowledge gained from solving one ML problem is transferred to another, similar problem.
- Transparency** Transparency is a property of a model that is understandable on its own.
- True Negative Rate (TNR)** A negative sample that is correctly classified as negative.
- True Positive Rate (TPR)** A positive sample that is correctly classified as positive, also called  $\rightarrow$  recall or  $\rightarrow$  sensitivity.
- Underfitting** A phenomenon that occurs when a statistical model cannot adequately capture the underlying structure of the data.
- Uniform manifold approximation and projection (UMAP)** Uniform manifold approximation and projection (UMAP) is a nonlinear dimensionality reduction technique that is similar to tSNE in many ways but implements a few modifications that lead to its improved performance.
- Unsupervised Learning** In contrast to  $\rightarrow$  supervised learning, this type of algorithm learns patterns from untagged data.
- Valence** State of positiveness to negativeness/pleasure to displeasure.
- Validation Set** A dataset that is held back from training a model to obtain an estimate of the model's skill while the model's hyperparameters are being tuned.
- Vocabulary** A list of all unique words in a corpus.
- Web ontology language** A model for defining and publishing ontologies on the Web that builds upon the resource description framework. It contains more pre-defined terms such as owl:sameAs, which are useful for adding a layer of agreed-upon semantics in ontologies and knowledge graphs.
- Wikidata** Maintained by the Wikimedia Foundation and similar to Wikipedia, Wikidata is a crowdsourced, encyclopedic knowledge graph with its own data model and access modalities.
- Word representation** Representing a word in a vector to capture different words with similar meanings in an efficient manner.
- Word-sense disambiguation** The ability to determine the meaning of a word in a specific computational context. For example, word-sense disambiguation (WSD) occurs when determining whether a reference to "apple" in a document refers to the company or the fruit.

# Index

## A

Ablation study, 115  
ABM, *see* Agent-based modelling (ABM)  
Accessibility of data, 54  
Accountability, 10  
Accuracy, 175, 176, 187, 203  
Activation function, 193–195, 217  
ADAM, 195  
ADF-GLS, 469  
Adjacency matrix, 454  
AFINN, 367  
Agent-based modelling (ABM), 481  
Agglomerative (bottom-up) clustering, 142  
Agglomerative clustering, 140, 141  
Agglomeration coefficient, 133  
Agglomeration schedule, 133, 135  
Agglomerative techniques, 132  
AI, *see* Artificial intelligence (AI)  
AIC, *see* Akaike's information criterion (AIC)  
Airbnb, 37, 71  
Akaike's information criterion (AIC), 473  
Albert, 347  
Alexa, 85  
Algorithm, 5, 6, 9, 10, 38–40, 44–46, 86, 88–98, 100, 101  
Algorithm fairness, 57, 58  
Algorithmic bias, 10  
ANN, *see* Artificial neural networks (ANN)  
APIs, *see* Application programming interfaces (APIs)  
Application programming interfaces (APIs), 41–43, 69, 72, 75, 77, 80, 549, 554, 556  
ArcGIS, 515, 516, 518, 520

Arcs, 455  
Area under the receiver operating characteristic (ROC) curve (AUC), 44, 178, 260, 265, 269, 271, 272  
Artificial intelligence (AI), 56, 60–62, 86–88, 98, 550, 557–559  
Artificial neural networks (ANNs), 98–100, 147, 192, 193, 195, 204, 216, 233, 234, 468  
Association rule analysis, 143, 145, 147  
Assortativity, 455  
Attention mechanism, 347  
Augmented and virtual reality, 10  
Autocorrelation, 516, 517  
AutoKeras, 43  
AutoML, 101, 102, 110, 111, 126, 127, 233, 547, 556, 558  
Autoregressive integrated moving average (ARIMA), 468, 478  
Auto-sklearn, 43  
Auto-WEKA, 102  
Average linkage, 132, 140  
Average path length, 455, 457, 461, 463  
AWS Sagemaker, 102

## B

Backpropagation, 100, 194  
Backward feature selection, 115  
Bagging, 188  
Bag-of-words (BoW), 337, 339, 351  
Balanced accuracy, 175, 177, 202, 203  
Batch learning, 195

- Bayes classifiers, 183
  - Bayesian bidirectional long short-term memory (BILSTM), 247
  - Bayesian information criterion (BIC), 473
  - Bayesian optimization, 235, 237–240, 245, 246, 248
  - BeautifulSoup, 75, 76, 78
  - Bernoulli, 370, 372
  - BERT, *see* Bidirectional encoder representation for transformers (BERT)
  - BERTopic, 375–377, 382, 390
  - Between-cluster variation, 130
  - Betweenness, 456, 458, 459, 463
  - BI, *see* Business intelligence (BI)
  - Bias, 53, 55, 57, 58, 60, 61
  - Bias weight, 193–195
  - BIC, *see* Bayesian information criterion (BIC)
  - Bidirectional encoder representation for transformers (BERT), 336, 339, 347–348, 350, 351, 354–357, 365, 377, 410, 429, 431, 434
  - Big data, 5–12, 53, 55, 58–61, 68, 86–88, 93, 532, 542
  - BigML, 556, 557
  - Bi-gram, 316
  - BILSTM, *see* Bayesian bidirectional long short-term memory (BILSTM)
  - Binary classification, 170, 174, 176, 177, 189, 192, 194
  - Binning, 113
  - Bipartite, 455, 458, 464
  - Bivariate, 528, 529
  - Bixby, 85
  - Black-box, 366
  - Black-box models, 195, 217
  - Blue, 125
  - Bokeh, 533
  - Booking system, 136
  - Boosting, 188, 189, 201, 203
  - Border points, 134
  - BoW, *see* Bag-of-words (BoW)
  - Box models, 99
  - Breusch–Godfrey, 469
  - Bubble chart, 529
  - Business ecosystem, 87
  - Business intelligence (BI), 542, 543
  - Business value, 6
- C**
- CA, *see* Classification accuracy (CA)
  - Caffe, 547
  - Calinski–Harabasz index, 133, 135
  - Canberra, 160
  - Candidate generation, 408, 412–415, 418
  - CatBoost, 284
  - Categorical feature, 171, 179, 180, 184, 187
  - Centering, 113
  - Centralities, 456, 463
  - Centroid distance, 132
  - C4.5, 185
  - Charting libraries, 533
  - Chart.js, 533
  - Chatbots, 85
  - Chebyshev distance, 138
  - Chi-square correlation, 114
  - Churn prediction, 170, 205
  - Circular economics, 11
  - Classification, 169–204, 253–256, 260, 261, 264, 272, 273
  - Classification accuracy (CA), 256, 265, 267
  - Classification boundaries, 189, 202, 204
  - Classification error, 175
  - Classification function, 170–174, 176, 178, 179, 182, 183, 191
  - Classifier, 170, 171, 175, 181–184, 187, 189–191, 202, 203
  - Closeness, 455, 456, 458, 463
  - Cluster centroid plot, 140
  - Clustering, 90, 92–94, 96, 129–147
  - Clustering coefficient, 456, 459, 461, 463
  - CNNs, *see* Convolutional neural networks (CNNs)
  - Code of ethics, 53, 59
  - Coefficient of determination ( $R^2$ ), 212, 262
  - Cohen’s kappa, 370
  - Community detection, 93, 147
  - Complex adaptive systems, 454, 483, 486, 487, 489
  - Complete linkage, 132, 140, 141
  - Complex system, 453, 454, 483, 484, 486
  - Computable features, 111, 116
  - Computational models, 5, 7
  - Computer vision, 8, 10
  - Confusion matrix, 255, 257, 265
  - Confusion table, 174–176, 202, 203
  - Connection weights, 194, 195
  - Content analysis, 12
  - Content generation, 12
  - Continuous bag-of-words (CBOW), 343–345
  - Convolutional neural networks (CNNs), 100, 193
  - Coordinates, 513, 515, 518
  - Cophenetic distance, 133
  - Core points, 134
  - CorEX, *see* Correlation explanation (CorEX)

- Correlation-based measure, 131
  - Correlation explanation (CorEx), 375, 376, 381, 390, 397–398
  - Correlation heatmap, 541
  - Cortana, 85
  - Cosine, 343, 352
  - Cosine similarity, 138
  - Coronavirus disease-19 (COVID-19), 57, 61, 322
  - CountVectorizer, 337
  - Creativity techniques, 11
  - Cross-entropy, 179, 183, 194
  - Cross-industry standard process (CRISP), 40, 549
  - Cross-/upselling, 170, 205
  - Cross-validation, 44, 89, 172, 173, 175, 181, 205, 235, 242, 263, 264, 268–270
  - CSS, 75, 78, 80
  - C-support vector classifier (C-SVC), 191
  - C-SVC, *see* C-support vector classifier (C-SVC)
  - Cultural and historical heritage, 12
  - The Curse of Dimensionality, 151, 152, 156, 165
  - Customer segmentation, 136, 146, 147
  - Cyber-physical systems, 6
- D**
- Dashboards, 530, 532, 542–544
  - Data access, 41, 42
  - Data accuracy and validity, 54
  - Data analysis, 10
  - Data cleaning, 7, 9
  - Data collection, 41, 42, 46
  - Data exploitation, 7, 10, 11
  - Datafication, 8
  - Data gathering, 7–9
  - Data mining, 549, 550, 554, 555
  - Data processing, 6, 7, 9, 10
  - DataRobot, 558, 559
  - Data science (DS), 5–7, 10
  - Datasets, 4–6, 8, 9, 11, 12
  - Data validity, 56–58
  - Date values, 117
  - Data visualization, 6, 9, 527
  - Data wrangling, 531, 532, 539
  - Davies-Bouldin, 133, 135, 138
  - DBpedia, 435–444
  - DBSCAN, *see* Density-based spatial clustering of applications with noise (DBSCAN)
  - Decision trees, 96, 184–187, 189, 201
  - DeepExplainer*, 284
  - Deep learning (DL), 90, 100, 110, 126, 192, 194, 310, 311
  - DeepMatcher, 410, 416, 418, 419
  - Degree, 455–459, 463
  - Degree distribution, 455, 457, 463
  - Dendrogram, 133, 135
  - Density, 455–457, 461, 463
  - Density-based clustering, 93
  - Density-based spatial clustering of applications with noise (DBSCAN), 93, 129–147
  - Deontology, 52, 53
  - Descriptive features, 112, 114
  - Destination manager, 11
  - Diameter, 455, 457, 461, 463
  - Dickey–Fuller, 469, 470
  - Dimensionality reduction, 348
  - Dimension reduction, 138, 147
  - Dirichlet distribution, 377
  - Discretization, 113, 114
  - Discriminating, 56
  - Discriminatory models, 58
  - Distance-based measures, 131
  - Distillbert, 347
  - DL, *see* Deep learning (DL)
  - Doc2vec, 157, 344, 345, 350, 381
  - Domain expertise, 6
  - Domain knowledge, 35, 36, 38–40, 42–44, 531, 538
  - DSM Directive, 73, 74
  - D3, 533, 534
  - Durbin–Watson, 469
- E**
- EA, *see* Error analysis (EA)
  - EDA, *see* Exploratory data analysis (EDA)
  - Edges, 455, 456
  - Eigenvector, 456, 463
  - Elastic net, 183, 213
  - ELI5, 282
  - ELMo, 339, 341, 346–347, 351
  - ELU, *see* Exponential linear units (ELU)
  - Embedding Projector, 348
  - Emergence, 486, 487, 494, 496
  - Emotion, 363, 366–368, 372
  - Ensembles, 187, 188, 203, 204
  - Ensembling, 214
  - Entity linking, 429, 430, 437, 438, 444
  - Entity matching, 405
  - Entity resolution (ER), 431, 442
  - Environmental economics, 11
  - Epsilon, 135, 142
  - ER, *see* Entity resolution (ER)



Error analysis (EA), 112, 115  
 Error trend seasonal (ETS), 471–473, 476  
 Ethics, 11, 51–63  
 ETS, *see* Error trend seasonal (ETS)  
*e*-tube, 215, 216  
 Euclidean, 156, 160, 165  
 Euclidean distance, 133, 138, 142  
 Evaluation score, 253  
 Explainability, 10, 275, 277–280, 284–286, 289, 297, 299  
 Explainable artificial intelligence (XAI), 279, 283, 285, 299  
 Explanation (functions), 210  
 Exploratory data analysis (EDA), 110, 112, 115, 119, 121, 122, 126, 152, 165, 540  
 Exponential linear units (ELU), 194

## F

Facebook, 37, 69, 72  
 Fact checking, 12  
 Fake news detection, 10  
 False discovery rate, 175  
 False negative rate, 175  
 False negatives, 255  
 False positive, 255, 259, 260, 265  
 False positive rate, 175, 177, 203  
 Farthest neighbor, 132  
 FastText, 339, 345–347, 351  
 FE, *see* Feature engineering (FE)  
 Feature drill down, 116  
 Feature engineering (FE), 41–44, 88, 89, 94, 109–126  
 Feature filtering, 115  
 Feature importance, 114, 115, 123, 124, 215  
 Feature interaction, 114–116  
 Features, 57, 86–88, 92–94, 96, 97, 101, 253, 254, 267, 272, 383  
 Feature selection, 111, 114–116, 124  
 Feature stores, 126  
 Feature types, 111  
 Feedbacks, 483, 484, 486, 506  
 First Law of Geography, 514, 516  
 Flickr, 129, 130, 142–145  
 FNNs, *see* Fully connected networks (FNNs)  
 F1, 44, 257–258, 265, 370–372  
 F1-Score, 175, 444  
 Forecasting, 97, 100, 209, 210, 226  
 Formation and evolution of the network, 457  
 Forward feature selection, 115  
 FP-Growth algorithm, 145  
 Fraud, 54, 58  
 Fully connected networks (FNNs), 192, 193

## G

Gaussian, 154  
 Gauss kernel, 191  
 General Data Protection Regulation (GDPR), 74, 278  
 Generalization error, 172, 174, 188  
 Generalized Sequential Pattern (GSP) algorithm, 145  
 Generative Adversary Networks (GAN), 284  
 Genetic Algorithms, 235, 239, 240, 245, 246  
 Gensim, 377, 389, 390  
 Geodatabase, 514, 520  
 Geographically weighted regression (GWR), 518, 524  
 Geographic information system (GIS), 489, 494, 496, 498, 506, 513  
 Gini Coefficient, 44  
 Gini impurity, 185  
 Global vector embeddings (GloVe), 339, 342, 346, 347, 350, 351  
 Google Cloud Vision, 157  
 Google Colab, 548  
 Google TransformerXL, 347  
 GPS, 43, 68, 69  
 GPT-3, 434  
 GPU, 547, 548  
 Gradient boosting, 97, 215  
 Gradient descent, 189, 194, 217  
*GradientExplainer*, 284  
 Gradient tree boosting, 189, 201, 203  
 Grammar of graphics, 528  
 Granger causality, 469  
 Greedy methods, 115  
 Grid Search, 235–237, 240, 243, 244, 246  
 Ground truth label, 89

## H

Hamming, 409, 415  
 Hamming distance, 415  
 HDBSCAN, 382  
 Heterogeneity, 481, 483–486, 494–496, 503, 506  
 Hidden layer, 99  
 Hierarchical cluster analysis, 132–134  
 Hierarchical clustering, 93, 130, 133, 139, 140, 142, 143  
 Highcharts, 534  
 Histograms, 114, 529, 538  
 Hold-out method, 172, 175, 181, 201  
 Hot spot analysis, 517, 523  
 H<sub>2</sub>O, 285  
 H2O AutoML, 102

Human-in-the-loop, 111, 112  
 Hume, D., 52  
 Hyperopt, 240, 245  
 Hyperparameters, 89, 101, 173, 182, 187–189, 191, 192, 195, 202, 205, 231–248, 344–346, 379, 383, 385, 386, 390, 393  
 Hyperparameter tuning, 231–248  
 Hyperparameter-tuning and supervised ML models, 288  
 Hypothesis, 531, 542

## I

ID.3, 185  
 IE, *see* Information extraction (IE)  
 ImageNet, 94  
 Importances of features, 188  
 Imputation, 113, 123  
 Incremental construction, 115  
 Industry 4.0, 8  
 Information extraction (IE), 429, 431  
 Information gain, 44, 185  
 Information systems, 6, 8, 10  
 Instagram, 37, 43, 69, 72, 157, 163, 164  
 Instances, 86, 93, 99, 102  
 Intelligence augmentation (IA), 4  
 Interactions, 482–495, 497–500, 502, 504–506  
 Interdisciplinarity, 35–46  
 Interdisciplinary, 60, 527, 531  
 Internet of things (IoT), 6, 8, 9, 46, 68  
 Interpretability, 275, 277–280, 285, 299  
 Inverse distance weighted (IDW), 517  
 Inverse logit, 182  
 IoT, *see* Internet of things (IoT)

## J

Jaro Winkler, 409  
 Java, 550, 554, 556  
 Jupyter, 547  
 Jupyter notebooks, 111, 121, 122, 125, 541

## K

Kaggle, 71, 538  
 Kantian approach, 52  
 Keras, 204, 225, 284, 547, 553  
 Kernel function, 191  
 Kernel trick, 191  
 k-Fold, 235  
 KG, *see* Knowledge graph (KG)  
 k-means, 129–147  
 k-means clustering, 93, 134, 136, 138–144

k-nearest neighbors (k-NN), 97, 161  
 KNIME, *see* Konstanz Information Miner (KNIME)  
 k-NN, *see* k-nearest neighbors (k-NN)  
 Knowledge graph (KG), 423  
 Knowledge representation, 4  
 Kohonen networks, 147  
 Konstanz Information Miner (KNIME), 102, 548, 552–554  
 KPSS, 469  
 Kriging, 517  
 Kullback-Leibler divergence, 154

## L

Labeling, 135  
 Language detection, 315, 322  
 LASSO, *see* Least absolute shrinkage and selection operator (LASSO)  
 Latent dirichlet allocation (LDA), 310, 313, 375–379, 381, 383–397  
 Latent semantic analysis (LSA), 376, 389  
 LDA, *see* Latent Dirichlet Allocation (LDA)  
 Lda2vec, 94  
 LDavis, 382, 385, 386, 393  
 Lead scoring, 170, 196, 205  
 Learning by example, 5  
 Least absolute shrinkage and selection operator (LASSO), 183, 213, 242, 243  
 Leave-one-out (LOO), 264  
 Leiden, 93  
 Leiden algorithm, 457  
 Lemmatisation, 43, 318, 319, 326, 327, 368  
 Lemmatising, 379, 383  
 Levenshtein Edit Distance, 409  
 Lexicon-based approach, 365, 366, 369, 371, 372  
 Lift curve, 266  
 LightGBM, 284  
 LIME, 281–282, 286, 294–297, 299  
 Linear kernel, 191  
 Linear regression, 212, 213, 221–223  
 Links, 454–459, 462, 464  
 Ljung–Box, 469  
 Local interpretable model-agnostic explanation (LIWC), 367  
 Local tourism boards, 11  
 Logistic regression, 97, 98, 182, 183, 194, 201, 203, 204  
 Long-short-term-memory (LSTM), 347, 349  
 Louvain, 93, 94  
 Louvain algorithm, 147  
 Lowercasing, 316, 379, 383

- L1 regularization, 100  
 L2 regularization, 100  
 LSA, *see* Latent semantic analysis (LSA)
- M**
- Machine learning (ML), 5–7, 9–11, 85–101, 532, 539  
 Machine Learning for Language Toolkit (MALLET), 377, 389  
 MAE, *see* Mean absolute error (MAE)  
 Mahalanobis distance, 131  
 Manhattan, 160  
 Manual Search, 235, 236, 240, 242, 246  
 MAPE, *see* Mean absolute percentage error (MAPE)  
 Margin, 189, 190, 215  
 Market segmentation, 129, 130  
 Markov clustering, 93  
 Mathematical modeling, 454  
 MATLAB, 547  
 Matplotlib, 533  
 Matthews correlation coefficient (MCC), 175, 176  
 Maximum likelihood estimator, 114  
 MCC, *see* Matthews correlation coefficient (MCC)  
 MDS, *see* Multi-Dimensional Scaling (MDS)  
 Mean absolute error (MAE), 44, 211, 248, 261, 262  
 Mean absolute percentage error (MAPE), 44, 211, 248, 471, 474–476  
 Mean Decrease Accuracy (MDA), 282  
 Mean square error (MSE), 153, 211, 213, 217, 222, 242, 246, 247, 261, 262  
 Measures, 454–457, 459, 463  
 Metadata, 87  
 Microsoft Power BI, 533  
 Minibatches, 195  
 Minkowski, 160  
 minPts, 134, 135, 142, 144  
 Misclassification rate, 171  
 Missing data, 112, 113, 117  
 Missing values imputation, 181  
 ML, *see* Machine learning (ML)  
 MLBox Auto-ML, 102  
 MLPs, *see* Multi-layer perceptrons (MLPs)  
 MNIST, 153  
 Mobility, 482, 483, 486, 491, 494  
 Model evaluation, 41, 42, 44, 253–273  
 Model fitting, 89  
 Model tuning, 41, 42  
 Modelling, 38–40, 43, 44
- Modifiable Aerial Unit Problem (MAUP), 519, 524  
 Modular structure, 456  
 Modularity index, 456, 463  
 Monge-Elkan string comparison, 409  
 Moral evaluation, 52  
 MSE, *see* Mean square error (MSE)  
 Multi-class classification, 170  
 Multi-Dimensional Scaling (MDS), 152–154, 157, 160, 165, 166  
 Multi-layer perceptrons (MLPs), 97, 193  
 Multinomial, 370, 372  
 Multinomial logit, 98  
 Multivariate, 528, 529, 537, 542, 543  
 Mutual information, 114, 115
- N**
- Naïve Bayes, 96, 365, 368, 370  
 Naïve Bayes classifiers, 184  
 Named entity recognition (NER), 310, 319, 320, 328, 429, 430, 437  
 Natural Language Generation (NLG), 309  
 Natural language processing (NLP), 4, 8, 10, 12, 89, 100, 307–330, 335, 341, 342, 346–348, 357, 376, 539  
 Natural Language Toolkit (NLTK), 317, 319, 323, 368  
 Nature-based tourism, 130  
 NDCG, *see* Normalized Discounted Cumulative Gain (NDCG)  
 Nearest neighbor, 132, 135, 144  
 Nearest neighbor classifier, 181, 182, 202  
 Negative predictive value, 175  
 Neo4j, 432, 444  
 NER, *see* Named entity recognition (NER)  
 Network (graph), 454, 455  
 Network study, 456  
 Neural networks, 90, 94, 96, 98–100, 336, 341, 342, 347, 349, 351  
 NLP, *see* Natural Language Processing (NLP)  
 NLTK, *see* Natural Language Toolkit (NLTK)  
 NMF, *see* Non-Negative Matrix Factorisation (NMF)  
 Nodal properties, 456  
 No free lunch theorem, 89  
 Noise points, 134, 144  
 Nonlinear relationships, 484  
 Nonlinearity, 481, 483, 486, 494  
 Non-Negative Matrix Factorisation (NMF), 375, 376, 379–381, 390  
 Normalization, 113, 180–183, 187–189, 192, 195, 202, 204

- Normalize, 138, 139, 142
- Normalized Discounted Cumulative Gain (NDCG), 444
- Notebooks, 547, 558
- NSGA-II, 283
- Nuisance variations, 109–112, 114
- Numerical feature, 171, 184, 187–189
- $\nu$ -SVR, 215, 216, 224
- NVD3, 534
  
- O**
- Octoparse, 73
- ODD+D protocol, 502
- One-dimensional regression, 210, 216, 218, 224
- One-hot encoding, 116, 117, 179, 336
- Online feedback data, 130
- Online search data, 130
- Online travel agencies (OTAs), 62
- OOB estimates, *see* Out-of-bag (OOB) estimates
- Open data, 69, 71
- Opportunity scoring, 170, 205
- Orange, 548, 550, 552
- Orange 3, 41, 43, 44, 102
- Ordinary least squares (OLS), 517, 518, 524
- OTAs, *see* Online travel agencies (OTAs)
- Outliers, 134, 138, 139, 142–144
- Out-of-bag (OOB) estimates, 188, 215
- Overfitting, 96, 100, 101, 254, 263, 272, 371
- Over/underfitting, 57
- OWL, *see* Web Ontology Language (OWL)
  
- P**
- Pandas, 196, 218
- Parameters, 173, 179, 181, 183, 188, 191, 195
- Partitional clustering, 136, 139, 140
- Partitioning, 132, 134, 140, 144
- Partitioning clustering, 144
- Part-of-speech (POS), 319, 368
- Pattern recognition, 9
- PCA, *see* Principal component analysis (PCA)
- Personal attributes, 55
- Personalized way, 196, 217
- Phillips–Perron, 469
- Pipeline, 120, 123
- Platt scaling, 254
- Polynomial kernel, 191
- Porter Stemmer, 318, 337
- Positive predictive value, 175
- Posterior class probability, 179, 183, 187, 192
  
- Power-law, 457, 463
- P-R Curves, 258
- Precision, 44, 175, 257, 258, 265, 437, 442, 444
- Prediction, 57, 61, 170, 182, 188, 195, 196, 199, 205, 210–212, 214, 217, 221, 222, 224, 225
- Pre-processing, 38, 41–43, 136, 138, 139, 141–143, 307, 314, 315, 318–320, 322, 331
- Principal component analysis (PCA), 94, 152, 153, 155–157, 159–162, 165, 180, 340, 348
- Privacy, 8, 9, 12
- Privacy rights, 54
- Probabilities, 254–256, 258–261, 266, 267
- Problem framing, 7, 8
- Profiling, 135
- Projection, 518, 520, 521
- Pruning, 185
- Python, 102, 375, 385–390, 398, 532, 533, 539, 540, 542, 544
- PyTorch, 204, 225, 284, 299
  
- Q**
- Qlik, 533
  
- R**
- Radial basis function (RBF) kernel, 98, 191, 215
- Radio-frequency identification (RFID), 6
- RandomForest, 284
- Random forest (RF), 86, 97, 98, 115, 123, 124, 187–189, 201, 203, 214, 215, 221, 222, 224, 233, 238
- Random sampling, 263
- Random Search, 235–237, 240, 243–246
- RapidMiner, 102, 129, 136, 139–143, 548
- Raster, 515, 516, 519, 522
- RBF kernel, *see* Radial basis function (RBF) kernel
- RDF, *see* Resource Descriptipn Framework (RDF)
- Recall, 44, 175, 177, 256–258, 265, 272, 437, 440, 442, 444
- Receiver operator characteristic (ROC) curves, 177, 178, 202, 203, 258–260, 265
- Recommendation systems, 10
- Record linkage, 405
- Rectified Linear Units (ReLU), 194
- Recurrent neural networks (RNNs), 99, 100, 193
- Red, 125

- Regression, 209–226, 253, 254, 260, 262, 264, 269, 270, 272, 273
- Regression function, 210–212, 214–216, 221, 224
- Regression trees, 213–215, 221, 222
- Regressor, 210, 215, 216, 221, 223
- Regularization, 171, 183
- Reinforcement based, 5
- Reinforcement learning, 89, 98
- ReLU, *see* Rectified Linear Units (ReLU)
- Reputation monitoring, 10, 12
- Residuals, 211
- ResNet50, 94
- Resource Description Framework (RDF), 427, 435
- RF, *see* Random Forest (RF)
- RF's feature importance, 115
- R/Julia, 102
- Ridge regression, 183, 213
- RMSE, *see* Root mean square error (RMSE)
- RMSProp, 195
- RNNs, *see* Recurrent neural networks (RNNs)
- RoBERTa, 347
- Robotics, 6, 10
- ROC-AUC, 178
- ROC curves, *see* Receiver operator characteristic (ROC) curves
- Root mean square error (RMSE), 248, 261, 262, 471, 474–476
- Ruby, 556
- S**
- Sample (Pearson) correlation coefficient, 212
- SAS, 555, 556
- Scatterplot, 199, 529, 534
- SciBERT, 348
- Scikit-learn, 201, 204, 221, 225
- Scrapestorm*, 73
- Scraping, 67–80
- Scrapy, 75, 77, 79
- Seaborn, 533
- Seasonal naïve, 471, 472, 475, 476
- Security, 8
- Selenium, 72, 76, 77
- Self-organizing maps (SOMs), 147
- Semantic, 337, 339–342, 345
- Semantic Web (SW), 425–427, 432, 442, 444
- Sensitivity, 175, 177, 256–260
- Sensors, 5–7, 10, 11, 68
- Sentiment analysis, 10, 44, 45, 309–314, 320, 363–372
- SentiWordNet, 367, 369, 370
- Sequence-to-sequence (Seq2Seq), 349
- Sequential pattern mining, 143, 145
- SES, *see* Single exponential smoothing (SES)
- SGD, *see* Stochastic gradient descent (SGD)
- Shadow model, 280
- SHAP values, 115, 125
- Shapley additive explanations (SHAP), 284–286, 289–294, 296, 297, 299
- Short-text analysis, 383
- Sigmoid, 182, 183, 194
- Selenium, 75
- Silhouette* scores, 133, 135
- Similarity measure, 131, 138, 142
- Single exponential smoothing (SES), 468, 471, 472, 476
- Single feature transformations, 116
- Single linkage, 132, 133, 140, 141
- Singular Value Decomposition (SVD), 152, 153
- Siri, 85
- Skip-gram, 341, 343–345
- Smart city, 8, 11
- Smart destinations, 87
- Smart tourism, 87
- Smith, A., 52
- Snowball Stemmer*, 318
- Social networks, 6
- Soft-margin SVM, 190
- Softmax, 194
- SOMs, *see* Self-organizing maps (SOMs)
- Soundex, 408
- spaCy*, 312, 319, 320, 328, 329, 350
- Spatial analysis, 513, 522
- Spatial autocorrelation, 514, 516
- Spatial database, 515
- Spatial interpolation, 517
- Spatial regression, 514, 517
- Specificity, 175, 177, 258–260, 263
- Speech recognition, 10
- Splitting criterion, 185, 213
- Squared Euclidean distance, 132
- Stakeholders, 11
- Standardization, 113
- Standards, 7, 9, 10
- Statistics, 35, 36, 38, 40, 42–44, 46
- Stemming, 311, 312, 318–319, 337, 368, 378, 379, 381, 383
- Stochastic gradient descent (SGD), 195
- Stop word removal, 368
- Strawman imputation, 181
- Streams, 7, 9
- Structured data, 7
- Student's t-distribution, 154

Supervised, 5  
 Supervised FE, 112  
 Supervised learning, 89, 95, 96  
 Support vector machines (SVMs), 96–98, 189–192, 202–204, 233, 276, 365, 368, 370–372  
 Support vector regression, 215  
 Support vector regressors, 215, 216  
 Surrogate, 280–282, 296  
 Sustainable development, 11  
 SVD, *see* Singular Value Decomposition (SVD)  
 SVMs, *see* Support vector machines (SVMs)  
 SW, *see* Semantic Web (SW)  
 Swift, 556  
 Switch, 434  
 Systems thinking, 483, 487

## T

Tableau, 533, 539, 540, 542  
 Target value, 210, 215, 227  
 t-Distributed Neighbor Embedding (tSNE), 154, 155  
 t-Distributed Stochastic Neighbor Embedding (tSNE), 94, 95, 348  
 TDM, *see* Text and data mining (TDM)  
 TensorFlow, 204, 225, 284, 299, 547  
 Term Frequency-Inverse Document Frequency (TF-IDF), 97, 338–339, 350–353, 370  
 Test sets, 172, 234  
 Testing data, 89, 263  
 Text and data mining (TDM), 73, 74  
 Text summarisation, 309, 310, 312  
 Theano, 547  
 3D representations, 535  
 Ties, 455  
 Time series analysis, 97  
 Tokenization, 43, 312, 315, 316, 323, 327, 368, 383, 390  
 Top-down clustering, 139, 140  
 Topic modelling, 375–398  
 Topology (structure), 457  
 Top2Vec, 375, 376, 381–382, 390  
 Torch, 547  
 TourBERT, 348  
 Tourism destinations, 12  
 Tourism (mobile) apps, 11  
 Tourism2Vec, 350  
 TourMIS, 470  
 TPOT, 240, 246  
 Tracking technologies, 11  
 Trafacta, 539  
 Trainable FE, 112

Training, 86, 88, 89, 95, 96, 98, 100, 101  
 Training data, 254, 263, 264, 267, 272  
 Training set, 172  
 Transformation, 136  
 Transformers, 342, 347–349  
 TripAdvisor, 37, 44, 69, 72, 75, 76, 78  
 True negative, 255, 256, 258  
 True negative rate, 175  
 True positive rate, 175, 177, 203  
 tSNE, *see* t-Distributed Stochastic Neighbor Embedding (tSNE)  
 Turing complete, 98  
 Twitter, 69, 72

## U

UDPipe, 319  
 UMAP, *see* Uniform Manifold Approximation and Projection (UMAP)  
 Underfitting, 100, 101  
 Unethical, 52, 57, 61  
 Uniform Manifold Approximation and Projection (UMAP), 94, 95, 155–157, 159, 163–165, 348, 382  
 Univariate, 528, 529  
 Unstructured, 7, 9  
 Unsupervised, 5  
 Unsupervised learning, 89, 91, 92, 95  
 Unsupervised machine learning, 146, 147  
 User-generated content (UGC), 307  
 Utilitarianism, 52, 53

## V

Valence Aware Dictionary for Sentiment Reasoning (VADER), 369, 370  
 Validation, 234  
 Value, 7, 8  
 VAR, *see* Vector autoregressive models (VAR)  
 Variety, 5–7, 10  
 Vector, 335–337, 339–342, 344–349, 351–354, 357  
 Vector autoregressive models (VAR), 468  
 Vector data, 515, 516  
 Vectorisation, 43  
 Vega-Lite, 534  
 Velocity, 6  
 Veracity, 7  
 Vertices (nodes), 455  
 Virtual assistants, 10  
 Visualization, 38, 43, 44  
 Visualization techniques, 527, 528  
 Volume, 6

**W**

Ward's linkage, 133  
Wearables, 68  
Web navigation data, 130  
Web Ontology Language (OWL), 427, 437  
Web tracking, 196, 217  
WEKA, 550, 554, 555  
Wheel of emotions, 366  
White-box models, 185, 214  
Wikidata, 428, 435, 437, 441, 443, 444  
Within-cluster variation, 130, 131, 138  
Word embeddings, 320, 335–357  
Word intrusion, 385, 397  
Word2vec, 339, 341–347, 350, 351, 353–354,  
410

WordNet, 319, 321  
Wrapper methods, 115

**X**

XGBoost, 223, 284  
XLNet, 347  
*x*-means, 147

**Z**

Z-score standardization, 138