Analysis of Customers' Reviews Using Soft Computing Classification Algorithms: A Case Study of Amazon

Koushal Kumar and Bhagwati Prasad Pande

Abstract Recently, text and sentiment analysis has received tremendous attention, especially due to the availability of gigantic data in the form of unstructured text available on social media, E-commerce websites, E-mails, blogs, and other similar sources. It involves analyzing large volumes of unstructured text, extracting relevant information, and determining people's opinions and expressions like positive, negative, neutral etc. Nowadays, the majority of business firms are using text and sentiment analysis techniques to understand the feedbacks of their customers and to gain information about the degree of customers' inclination towards their products and services. Therefore, sentiment analysis provides valuable insights and helps the firms to formulate effective business strategies. However, the massive data derived from social media and other sources are unstructured, highly dimensional, and involve uncertainty and imprecision. Thanks to soft computing techniques, we are equipped to handle uncertainty imprecision, partial truth, and approximation. The present chapter is based on text and sentiment analysis of customers' reviews collected from the *Amazon* customer review portal. We propose a three-tier model that takes raw data from this portal as input and generates a comparative report over certain parameters. We fetch data variables from this portal, apply data preprocessing and cleaning techniques to repair and/or remove dirty data in the first phase. In the second phase, we filter out those input variables which exhibit the strongest relationship with output variables using statistical feature selection techniques. In the final phase, we analyze processed dataset using machine learning algorithms to classify positive, negative and neutral reviews. For classification, we apply Random Forest, Naïve Bayes, and Support Vector Machine algorithms in particular. These algorithms are applied to the processed dataset to study a few parameters like accuracy, precision, F-measure, true positive, false negative, etc. Finally, our study compares the outputs of these three classifiers over the above-mentioned parameters.

K. Kumar
Sikh National College, Qadian, Guru Nanak Dev University, Amritsar, Punjab, India

B. P. Pande (✉)
Department of Computer Applications, LSM Government PG College, Pithoragarh, Uttarakhand, India

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

S. Dash et al. (eds.), *Advanced Soft Computing Techniques in Data Science, IoT and Cloud Computing*, Studies in Big Data 89, https://doi.org/10.1007/978-3-030-75657-4_15

Keywords Text and sentiment analysis · Social media · Customers' reviews · Amazon · Statistical feature selection · Machine learning · Classifiers

1 Introduction

Computing in terms of computer technology refers to the process of executing tasks with the help of a computer device. A few characteristics of the computing process are: solutions must be precise and valid, there should be unambiguous and correct control sequences, and finally, formulation of the mathematical solutions of the problems should be easy. Computing methods are categorized into two folds: hard computing and soft computing. Hard computing is the traditional practice that draws on the postulates of precision, certainty, rigour, and inflexibility. It requires a well defined analytical model and often takes a significant amount of computation time. On the other hand, soft computing is different from conventional computing, it includes the concept of approximate models and provides solutions to tricky real-world problems. Unlike hard computing, soft computing deals with imprecision, uncertainty, approximations, and partial truths. Soft computing incorporates modern theories and practices such as expert systems, fuzzy logic, genetic algorithms, artificial neural networks, and machine learning. Some of the key differences between hard and soft computing are: hard computing draws on binary (two-valued) logic and deterministic in nature, while soft computing works upon formal (multi-valued) logic and stochastic reasoning; hard computing requires exact data for its mechanism, while soft computing can tackle ambiguous and noisy data; hard computing executes sequential computations, on the other hand, soft computing is capable of performing parallel computations; hard computing requires explicit programs to be written, while soft computing can emerge its own programs.

Sentiment analysis is one of the soft computing techniques that perceives positive, negative, or neutral opinions, known as *polarity* within a piece of text. This text can be a clause, sentence, paragraph, or a whole document. Let us take customer feedback as an illustration, sentiment analysis weighs the inclination of customers towards a product or service, which they express in textual form as comments or feedbacks. For example, consider the following feedbacks by two different customers (Table 1).

The goal of sentiment analysis is to take a piece of text as input, analyze it, and returning a metric or score that estimates how positive or negative the text is. The process can be understood as context-based mining of text to identify and

Table 1 Comments and corresponding sentiments

Customer's comment (text)	Sentiment
"The watch is great! Delivery was fast, and the customer care representative was very friendly"	Positive
"I will not recommend this watch to anyone. The material is cheap, and it is truly a wastage of money"	Negative

extract subjective information from source data. It helps businesses to judge the social opinions of their products, brands, and services by monitoring the online activities of their customers. However, the analysis of web and social media platforms is limited to trivial sentiment analysis and count-based metrics: akin to engraving the surface and overlooking other important insights that ought to be discovered.

Sometimes, sentiment analysis is coupled with *text analytics* and people often consider them as the same or related processes. Though both procedures extract meaningful ideas from customer data, both are the essential constituents of the *customer experience management module*, but, they are not the same thing. As we know, the former classifies a piece of text or expression as positive, negative, or neutral and determines the degree of this classification, the latter is concerned with the analysis of the unstructured text, extracting apt information, and converting it into productive business intelligence. Text analytics deals with the *semantics* of the text: involving the grammar and the relationships among the words. In general terms, text analytics draws out the *meaning*, while sentiment analysis develops an insight into the *emotions* behind the words. Sentiment analysis has an upper hand over text analytics that the former can be applied to non-text feedbacks such as *emoticons* or *emojis*. A '*grinning face with big eyes*' emoji is coupled with a higher sentiment score than the emojis of '*frowning face*' and '*zipper-mouth face*'.

Recently, text and sentiment analysis has received tremendous attention, especially due to the availability of gigantic data in the form of unstructured text available on social media, E-commerce websites, e-mails, blogs, and other similar web resources. This requires analyzing large volumes of unstructured text, extracting relevant information, and determining people's opinions and expressions. Nowadays, the majority of business firms are using text and sentiment analysis techniques to understand the feedbacks of their customers and to gain information about the degree of customers' inclination towards their products and services. Therefore, sentiment analysis provides valuable insights and helps the firms to formulate effective business strategies. However, the massive data derived from social media and other sources are unstructured, highly dimensional, and involve uncertainty and imprecision. This kind of massive text usually contains white spaces, punctuation marks, special characters, @ links, hashtag links, stop words, and numeric digits etc. This unstructured data must be cleaned before being fed to the classification models. These types of unnecessary expressions or characters can be removed using data pre-processing libraries available in *Python*. Thanks to soft computing techniques, we are equipped to handle uncertainty imprecision, partial truth, and approximation.

The rest of the chapter is organized as follows: Sect. 2 covers the literature review, followed by Sect. 3, data collection and methodology, Sect. 4 presents experimental results and discussions, followed by the final concluding section.

2 Literature Survey

Pang and Lee [20] presented an exhaustive survey on opinion mining and sentiment analysis. They explored research works that promise to directly enable opinion-oriented information-seeking systems. They focused to give more attention to contemporary challenges raised by modern sentiment-aware applications rather than already available traditional fact-based analysis models. Prabowo and Thelwall [21] proposed a hybrid approach to sentiment analysis based on rule-based classification and machine learning. They proposed a complementary and semi-automatic approach where every classifier supports other classifiers. They tested their hybrid model over movie reviews, product reviews, and *MySpace* comments and reported that a hybrid model is capable of improving classification effectiveness in terms of micro-and macro-averaged F1 measure. The authors suggested that in real-world applications, it would be better to have two rule sets: the original and induced rule sets. Barbosa and Feng [5] investigated the writing pattern of *Twitter* messages and meta-information of the words that constitutes them. Based on this data, they proposed the automatic detection of sentiments on tweets. They utilized biased and noisy labels of tweets provided by a third party and used this source as training data. They combined these labels by utilizing various strategies and compared their model with already existing techniques. The authors claimed that the solution proposed by them can handle more abstract representation of tweets and proved to be more robust and effective. Agarwal et al. [1] studied *Twitter* data for sentiment analysis. They proposed two models: one binary model to classify tweets as positive and negative and one 3-way model to classify them as being positive, negative, and neutral sentiment. They performed experiments with the unigram model, feature-based model, and kernel-based model. The authors used the unigram model as a baseline and reported an overall gain of 4% for these classification tasks. They claimed that the feature-based and tree kernel-based models outperformed the unigram baseline. In their concluding remarks, the authors stated that the sentiment analysis for *Twitter* data is the same as sentiment analysis for other genres. In their work, Gräbner et al. [12] proposed a classification system of the reviews of hotel customers employing sentiment analysis. Given a corpus, they designed a process to collect words that are related semantically and developed a domain-specific lexicon. This lexicon served as the key resource to develop a classifier for the reviews. The authors claimed to achieve a classification accuracy of 90%. Liu [14] presented a minutely detailed work on sentiment analysis and opinion mining. The author gave an in-deep introduction and presented a thorough survey of the available literature and the latest developments in the realm. This work presents an excellent qualitative and quantitative analysis of opinions and sentiments and stands as a distinguished literary resource for practical applications. The author endeavored to develop a common framework to bring different research works under a single roof and discussed the integral constituents of the subject like document-level sentiment classification, sentence-level subjectivity and sentiment classification, aspect-based sentiment analysis, sentiment lexicon generation, opinion summarization, and opinion spam detection. Bagheri et al. [3] proposed an unsupervised and domain-and language-independent model for analyzing online customers' reviews for sentiment analysis and opinion mining. Their generalized model was equipped with a set of heuristic rules to detect the impact of opinion word/multi-word. They presented a novel bootstrapping algorithm and proposed a metric to detect and score for implicit and explicit aspects of reviews. They claimed that their model can be used in a practical environment where high precision is required. Medhat et al. [17] gave a detailed analysis of sentiment analysis algorithms and their applications. Their work can be considered as the state of the art in the domain. The authors categorized a large number of research articles according to their participation in sentiment analysis techniques for real-world applications. They suggested that further research is needed to enhance Sentiment Classification (SC) and Feature Selection (FS) algorithms. They commented that the *Naïve Bayes* and *Support Vector Machine (SVM)* serve as the base or reference techniques for comparing novel SC algorithms. Fang and Zhan [10] proposed a general process for sentiment polarity categorization thereby giving detailed descriptions. They studied online product reviews from *Amazon.com* over the following major categories: beauty, books, electronics, and home appliances. Each review includes rating and review text among other data. The authors used a part of speech (POS) tagger at the preprocessing step and then computed sentiment score. They used the F1 measure to evaluate the performance of their proposed classification process and reported that the *SVM* model and the *Naïve Bayes* model performed almost the same. Mozetič et al. [19] exploited a big set of tweets from different languages. The tweets were labeled manually and they exploited them as training data. They proposed automatic classification models and reported that the performances of the top classification models are not statistically different. The authors concluded that it would be good to give more attention to the accuracy of the training data than the genre of the model being employed. They found that on applying to the three-class sentiment classification problem, there is no correlation between the accuracy and performance of the classification models. From the literature available on sentiment analysis, this work on human annotation is very unique. Saad and Saberi [22] presented a survey of sentiment analysis and opinion mining techniques, their applications, and challenges. The authors classified such techniques into three groups: machine learning approach, lexicon-based approach, and combination method. They collected data from blogs and forums, reviews, news articles, and social networks (*Twitter* and *Facebook*). They concluded that the unstructured data is a big hurdle in sentiment analysis and stated that algorithms of sentiment classification and opinion mining need further research for improvement. Ghag and Shah [11] mentioned that the *bag-of-words* is a popular tool of sentiment analysis. The authors classified the sentences extracted from the sentiments by reviewing their syntactic and semantic structures. They proposed some metrics like relative frequency, term frequency, and inverse document frequency to improve accuracy. They used text preprocessing techniques and claimed to achieve 77.2% classification accuracy.

Employing automatic text clustering and manual qualitative coding, Mäntylä et al. [16] analyzed around seven thousand research articles from *Scopus* and *Google scholar* and presented a computer-assisted literature review for sentiment analysis. They highlighted a very interesting fact that automatic sentiment analysis had been possible only with the availability of online subjective texts and therefore, 99% of the research work in this domain took place after 2004. According to the authors, computer-based sentiment analysis started by analyzing product reviews available over the web, and it is now being applied over a wide range of domains like social media texts (*Twitter*, *Facebook*, etc.), stock markets, elections, disasters, medicine, and cyberbullying. They stated that sentiment analysis involves a multitude of data sources like tweets, comments, chats, emoticons etc. Alsaeedi and Khan [2] investigated applications and results of various sentiment analysis techniques over *Twitter* data. They explored machine learning, lexicon-based approaches, ensemble approaches, and hybrid approaches. They reported the following conclusions of their research work: when multiple features were taken, machine learning techniques resulted in the greatest precision; lexicon-based techniques performed good but they require manual efforts to create the archive, and the ensemble and hybrid-based algorithms performed better than supervised machine learning algorithms. Tyagi and Tripathi [26] also collected *Twitter* data and performed sentiment analysis. The authors extracted the features through the N-gram modeling technique and exploited the *K-Nearest Neighbor* algorithm to categorize sentiments into positive, negative, and neutral. Bhagat et al. [6] studied online product reviews, general tweets in *Twitter*, and movie reviews and carried out sentiment analysis of text messages using supervised machine learning techniques. They preprocessed the messages and applied *Naïve Bayes*, *Decision Tree*, and *Support Vector Machine (SVM)* techniques for their research. They proposed a three-tier framework: the first layer is the initialization layer for data collection and message preprocessing, the second layer is the learning layer which splits preprocessed data into training and test datasets and develops three machine learning models, the final layer evaluates the performance of the models based on precision, recall, F1-measure, etc. The authors concluded that the *Decision Tree* and *SVM* can be considered as good classifiers with lower mean square error.

3 Data Collection and Methodology

As we all know, *Amazon* is one of the leading E-commerce websites, where a large number of users' reviews can be found. After purchasing the products, customers can post their reviews directly on the *Amazon* review portal. With such a massive amount of customers' reviews, this provides an opportunity to study and investigate feedbacks of the customers about a specific product [8]. All such comments or feedbacks help the sellers and other potential customers comprehend product-related public opinions. In the present case study, we are taking reviews of *Amazon* customers for sentiment analysis. We propose a *three-tier model* that takes raw data from the *Amazon* portal as input and generates a comparative report over certain parameters. In the first phase, we fetch data from the portal, apply data preprocessing and cleaning techniques to repair and/or remove the dirty data. In the second phase, we apply *TF-IDF* and *Skip-Gram* models for statistical feature selection. This step filters

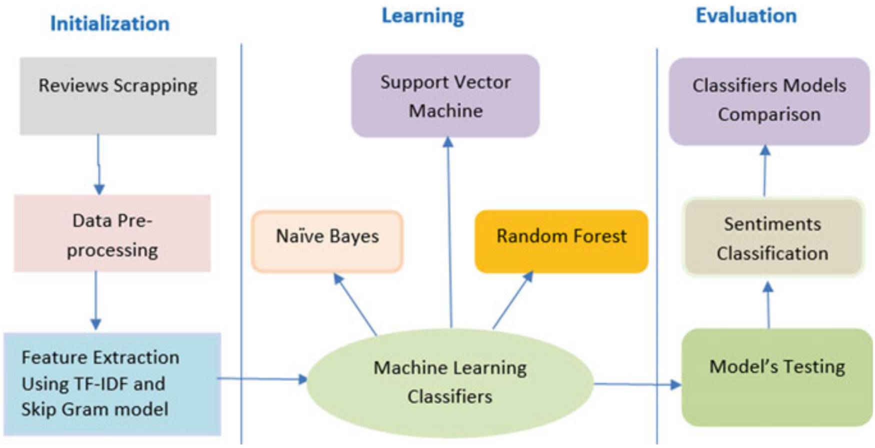

Initialization
Learning
Evaluation
Reviews Scrapping
Data Pre-processing
Feature Extraction Using TF-IDF and Skip Gram model
Support Vector Machine
Naïve Bayes
Random Forest
Machine Learning Classifiers
Classifiers Models Comparison
Sentiments Classification
Model's Testing

Fig. 1 Proposed model and its constituents

out those input variables which exhibit the strongest relationship with output variables. In the final phase, we apply machine learning (ML) algorithms say, *Random Forest*, *Naïve Bayes*, and *Support Vector Machine* analyzing the processed dataset and to classify the customers' reviews into the genres of positive, negative, and neutral. These algorithms are applied to the processed dataset to study the following performance parameters: accuracy, precision, recall, F measure, true positive, and false negative. Finally, our study compares the outputs of these three classifiers over the above-mentioned parameters. Figure 1 depicts the elements of our proposed model.

We hypothesize a four-fold methodology for our present research work: (a) data collection, (b) data preprocessing, (c) data representation, and (d) data classification. We now discuss the above steps in detail below.

3.1 Data Collection and Preprocessing

To conduct this case study, we gathered data from *the Amazon* web portal using an automated technique known as *Scrapping*. Scraping is a data extraction technique used for data collection from different websites. *Scrapy* is a free and open-source web-crawling framework which is written in *Python* and it is used for extracting data from websites. We applied the scraping process to extract *Amazon* reviews using *the Scrapy* library which permits the programmers to extract the data as per their requirements [18]. In our practical experiments, the scrapped data set consists of 300,000 mobile phone reviews from *the Amazon* review portal for various international brands. However, these reviews are in unstructured and unlabeled text form which requires pre-processing treatment. This is an essential step of the whole process as the accuracy of machine learning models depends on the quality of data we feed into them.

The scrapped data set used in our research had many missing or null values. We dealt with these issues by utilizing the *Imputation* technique, a widely used tool in the realms of machine learning and data mining. The basic principle of the technique is to replace each missing value of an attribute with the mean of the observed values of the attribute, known as Mean Imputation (MEI), or a nominal attribute with its most commonly observed value, known as Most Common Imputation (MCI). For each attribute f_i with missing values, the classifier C_i (. . .) takes as input the values of the other (*n* – *1*) attributes $\{f_j \mid j \neq i\}$ for an instance, and returns the value for f_i for this instance [4, 15, 24]. Other preprocessing treatments applied to this data before feeding the data to the machine learning models are spellings corrections; stop words removal; removal of special characters and punctuations from text data; removal of multiple spaces; removal of numeric digits from the review texts; removal of all URLs, hashtags, and E-mail addresses; upper to lower case conversion; contraction to expansion; substitution of any non-UTF-8 character by space; stemming; and removal of rare words. To improve the performance of the classifier's models, some of the irrelevant attributes (like reviews.dateSeen, reviews.sourceURLs, reviews.title, reviews.username, etc.) have been dropped after pre-processing.

Ultimately, after applying all of the above-mentioned preprocessing treatments, we receive accurate, useful, and clean text suitable for analysis and classification of sentiments. Table 2 presents the final extracted attributes and their description.

Every product rating is based on a 5-star scale ranged from 1-star to 5-star with no existence of a half-star or a quarter-star. Figure 2 depicted below shows the distribution of reviews based on *Amazon's* 1–5-star rating scales.

As shown in the above figure, the most frequent review rating in our dataset is 5 stars, with more than 30% share in the entire dataset. Figure 3 illustrated below shows the attributes which are of the numerical type and their distribution in the data set.

It is clear from the above figure that *reviews.numHelpful* is a valuable attribute in our dataset, so we kept only those instances in the dataset for which more than 75 people found the review helpful. On the other hand, in *reviews.rating* attribute, the distribution is skewed towards 5 stars rating. The last two attributes, *reviews.userCity* and *reviews.userProvince* have *NaN* values i.e., a numerical value that is undefined or not present. Therefore, we have dropped these attributes from our dataset. One important attribute that is used for product identification is *Amazon Standard Identification Number* (*ASIN*). Our dataset has 35 different products which possess unique *ASIN* values and are used for training our classifiers. After analyzing *ASINs* and product name attributes, we observed that there's a one to many relationships between the *ASINs* and the product names, i.e., a single *ASIN* is linked with one or more product names. Figure 4 shown below visualizes the individual *ASIN* and product reviews in a bar graph representation.

The above figure clearly shows that certain products have significantly more reviews than other products, which indicate a higher sale of those products. Based on this *ASIN* attribute frequency graph we can easily decide which products should be kept or dropped. Now, for better insight into the data or corpus, the *Wordcloud* visualization is an excellent tool in practice. The word that appears more prominent

Table 2 Features information of *Amazon* reviews dataset

S. No.	Attributes name	Description	Data type
1	id	This attribute represents a unique reviewer id number being assigned to each reviewer	Int64 (integer)
2	name	This attribute represents the product name	String
3	ASIN	This attribute stands for *Amazon Standard Identification Number*, a unique number assigned to each product	Object (integer and characters)
4	brand	It represents the brand name of the product	String
5	categories	It shows the category the product belongs to	String
6	manufacturer	This attribute represents the manufacturer of the product	String
7	reviews.date	This attribute expresses the date on which review posted on the *Amazon* portal	Float64 (floating point number)
8	reviews.doRecommend	This attribute shows whether a particular review is recommended by other reviewers or not	Boolean (yes/no)
9	reviews.numHelpful	The number of reviewers who consider a particular review useful	Int64 (integer)
10	reviews.userCity	This attribute represents reviewer city	String
11	reviews.userProvince	This attribute represents reviewer province or state	String
12	review rating	This attribute expresses the rating (1–5 stars) assigned to a particular product by a reviewer	Float64
13	review text	This attribute contains the actual review text posted by reviewers on the *Amazon* portal	Object (integer and characters)

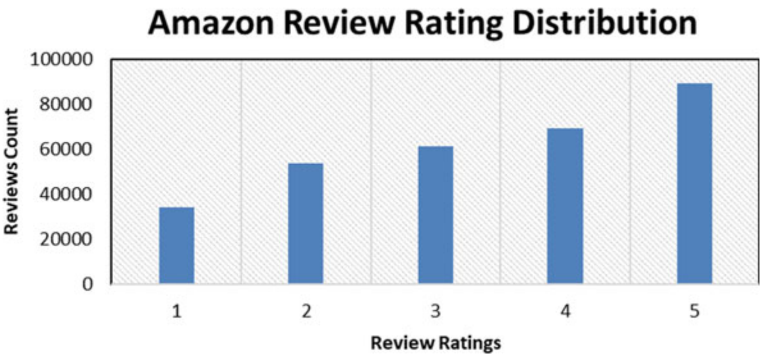


Fig. 2 Distribution of *Amazon's* star-rating scores

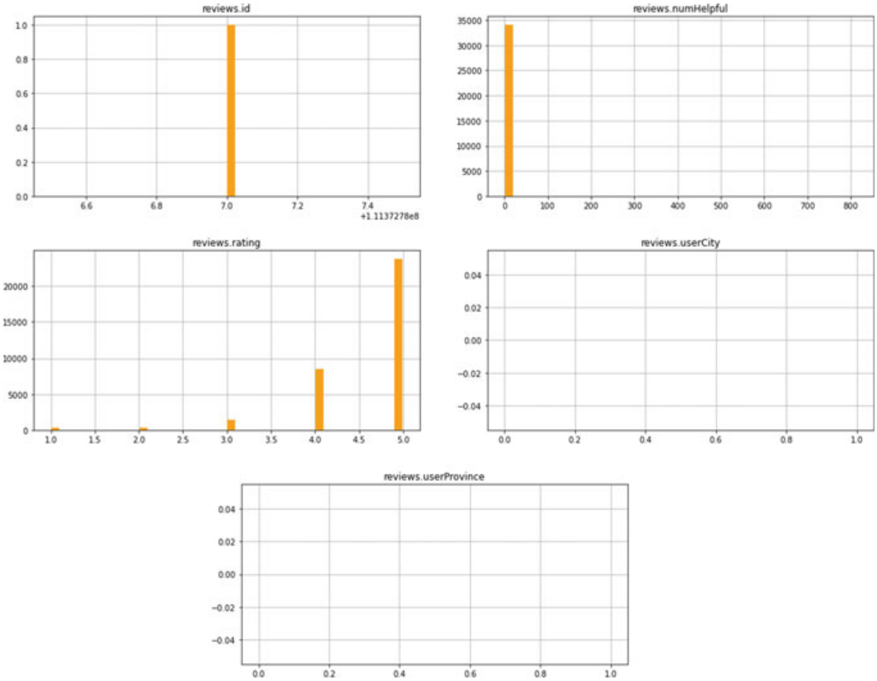


Fig. 3 Distribution of numerical data in the dataset

based on the frequency or importance in the text data is displayed with the bigger size in the *Wordcloud* visualization. In simple words, the word with larger size has more weight than the word with smaller size. After the completion of pre-processing of the dataset, we visualize words from the reviews' text using *Wordcloud* feature as shown in Fig. 5 below.

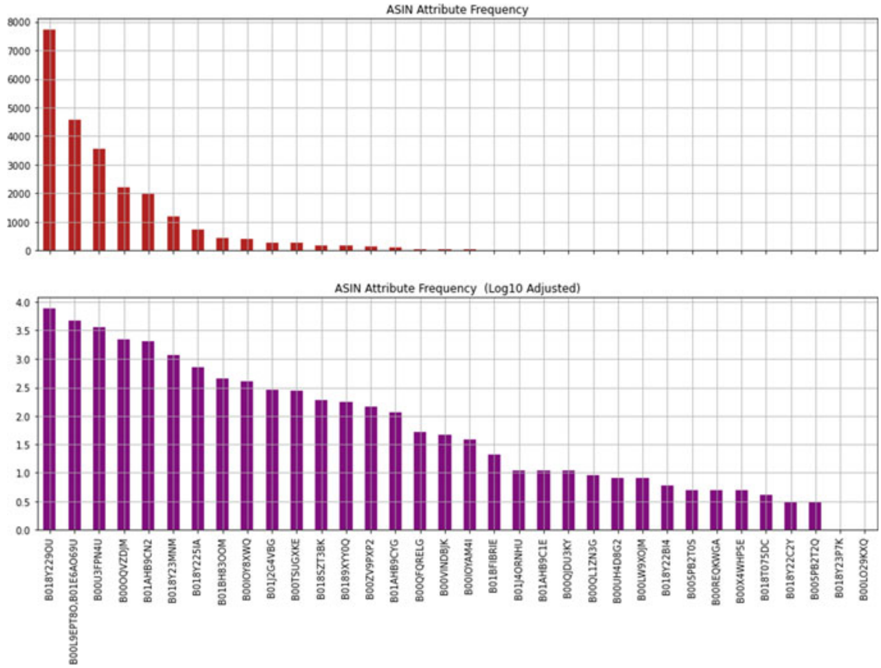


Fig. 4 Review ratings and ASIN frequencies

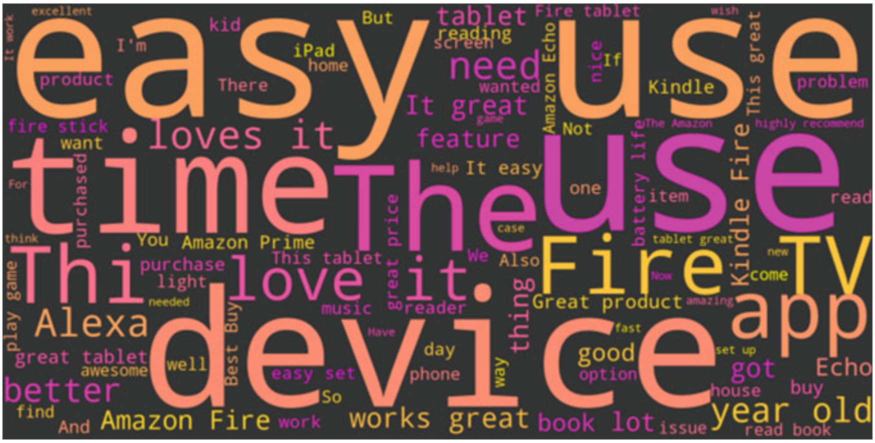


Fig. 5 *Wordcloud* of reviews' text

3.2 Data Representation

After the pre-processing of the unstructured text, the data representation is a vital step in sentiment classification. The extracted pre-processed reviews are mainly in text format but numerical representation in terms of metrics is needed to classify sentiments using the machine learning algorithms. Therefore, we have applied two different approaches to convert text data into some suitable form to be fed into the machine learning classifiers. The first approach is *word embedding* and second is the combination of *Term frequency and inverse document frequency (TF-IDF).* For word embedding, we applied the *Word2vec* model with *skip-gram* architecture. The *skip-gram* model predicts the source context words given a target word. It works as an unsupervised learning technique that is used to find the most suitable and related words for a given target word [13]. *Skip-gram* architecture provides more accurate and effective results when we have a corpus of bigger size, because, in the *skip-gram* approach, each context-center pair is considered as a new observation. The word vectors are adapted using Eq. (3.1), as given below:

$$w_{i,j}(k+1) = w_{i,j}(k) - s\frac{\partial J}{\partial w_{i,j}} \tag{3.1}$$

where $w_{i,j}(k)$ is word vector value in step k of the optimization process, j is our *optimization function* and s is the chosen step size. The *optimization function* is applied for selecting those words which can be represented using the Eq. (3.2) given below.

$$J = \sum_{i,j=1}^{V} f\left(X_{i,j}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{i,j}\right)^2 \tag{3.2}$$

where V is the number of word tuples with the non-zero co-appearance count, $X_{i,j}$ is the count of co-appearances, w_i is a word vector and $\tilde{w}_j$ is word vector's context, b_i and $\tilde{b}_j$ are biases (again every word has two of them: one for the word and other for the context) and function f is a weighing function. The *skip-gram* architecture is illustrated in Fig. 6 given below.

The *TF-IDF* algorithm is based on words' statistics for feature extraction and represents how important a word or a phrase in a corpus. *TF-IDF* assigns a unique score to each word using a hybrid statistical method, in terms of the product of *term frequency* (TF) with *inverse document frequency* (IDF). The TF denotes the total number of times a given term occurs in the dataset against the total number of all words in the document, and the IDF measures the amount of information word provides [23]. In our case study, TF assigns a score to most frequently occurring words in the mobile review dataset and IDF assigns weight to the least frequent words in the same dataset. The *TfidfVectorizer* from the *sklearn python library* is used to fit the vectorizer on the corpus of the review texts to extract features and the model will transform the text data into the *TF-IDF* representation. Using the *TF-IDF*

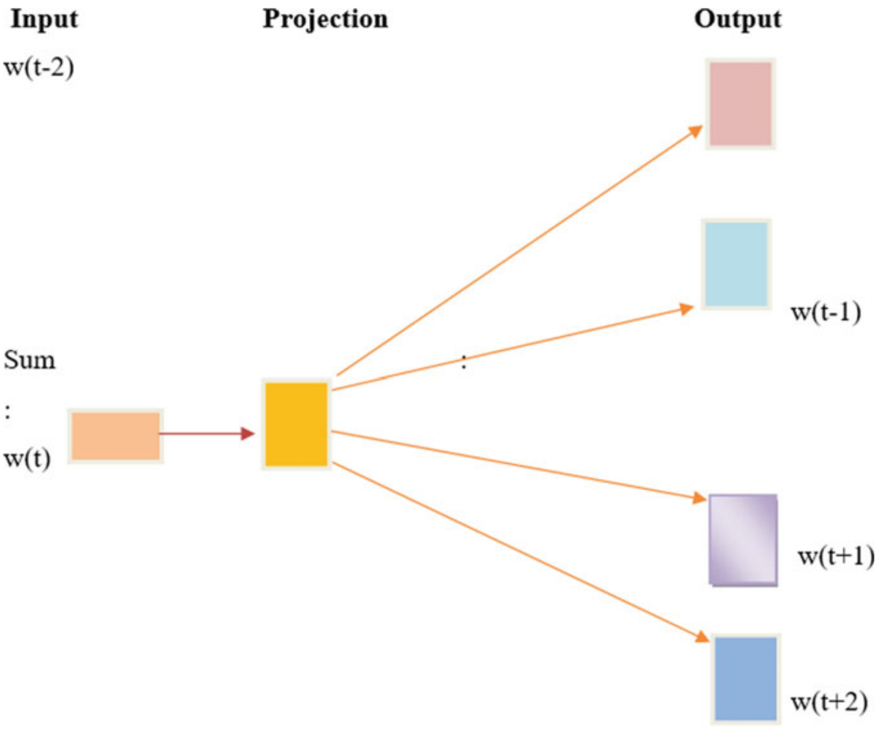


Fig. 6 Architecture of word to vector skip-gram model

approach in a normalized data format, each corpus word can be represented using the following equation:

$$W_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i}\right) \tag{3.3}$$

where, tf_{ij} is the number of occurrences of the ith word in the jth review and df_i is the number of reviews containing i and N is the total number of reviews.

3.3 Classifications

Classification is a supervised machine learning process that generally focuses on predicting a qualitative response by recognizing and analyzing given data points. This case study is focused on sentiment analysis by classifying the reviews into 1–5 ranked scales. To carry out this task, we have applied three different supervised classifiers: *Naïve Bayes*, *Support Vector Machine*, and *Random Forest*. These classifiers are then evaluated to provide a comparative analysis of various parameters for classifying reviews into positive and negative genres. The literature suggests that reviews with ratings 4 and 5 should be categorized as positive reviews while reviews with rating 3 should be labeled as neutral and reviews with ratings 2 and 1 should be treated as negative reviews. Since here we are interested in analyzing only positive and negative reviews, so we have neglected neutral reviews form the data set. The working of the individual classifier is explained in the next sections.

3.3.1 Naïve Bayes Classifier

The *Naïve Bayes* (NB) classifier is a simple and robust probabilistic classifier algorithm that is based on the *Bayes* theorem. It assumes that attribute values are independent of each other given the class. This assumption is known as the *conditional independence assumption.* Therefore, applying changes in one feature does not affect other features of the class [7]. Let D be our *Amazon* review data set for training the model then each tuple in the dataset is defined with n attributes and it is represented by: $X = \{a_1, a_2, a_3, \ldots, a_n\}$. Let there be m classes represented by: $\{C_1, C_2, C_3 \ldots, C_m\}$. For a given tuple X, the classifier predicts that X belongs to the class having the highest posterior probability, conditioned on X. The *Naïve Bayes* classifier predicts that the tuple X belongs to the class C_i if and only if $P(C_i|X)$ is maximum among all i.e.:

$$P(C_i|X) > P\big(C_j|X\big) for 1 \le j \le m, i \ne j \tag{3.4}$$

Since we want to maximize $P(C_i|X)$, the class C_i for which $P(C_i|X)$ is maximized is called the *maximum posterior hypothesis*. According to the *Bayes* theorem,

$$P(C_i|X) = \frac{P(X|C_i) \cdot P(C_i)}{P(X)} \tag{3.5}$$

If the attribute values are conditionally independent of one another (*Naïve Bayes* condition), then

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i) \tag{3.6}$$

where x_k refers to the value of the attribute A_k for the tuple X. If A_k is a categorical attribute, then $P(x_k|C_i)$ is the number of tuples of class C_i in D having the value x_k for A_k. The classifier predicts the class label of X is in the class C_i if and only if,

$$P(X|C_i) \cdot P(C_i) > P(X|C_j) \cdot P\big(C_j\big) \tag{3.7}$$

3.3.2 Support Vector Machine

Support Vector Machines (SVM) is a supervised machine-learning based classification algorithm which widely deals with predictive and regression analysis. *SVM* algorithm aims to find a *hyperplane* in an N-Dimensional feature space that distinctly classifies the data points, while maximizing the marginal distance for the two classes (positive and negative) and minimizing the classification errors. The marginal distance for a class is the distance between the decision hyperplane and its nearest instance which is a member of that class [25]. The data points that lie closest to the decision surface (or hyperplane) are called support vectors and these points help us in building the *SVM* model. The loss function that helps in maximizing the margin is given below.

$$C(x, y(f(x)) = f(x) = \begin{cases} 0, & x < 0 \\ 1 - y * f(x) & else \end{cases} \tag{3.8}$$

$$C(x, y(f(x)) = (1 - y * f(x))+ \tag{3.9}$$

The equation of the line in 2D space is $y = a + bx$. By renaming x with x_1 and y with x_2, the equation will change to $ax_1 - x_2 + b = 0$. If we specify $X = (x_1, x_2)$ and $w = (a, -1)$, we get,

$$F(x) = w \cdot x + b \; where \; w, x \in R^n \; and \; b \in R \tag{3.10}$$

The above Eq. (3.10) is called the equation of the hyperplane, which linearly separate the data.

The hypothesis function h in SVM classifier can be defined as:

$$h(x) = \begin{cases} +1 \; if w.x + bx \geq 0 \\ -1 \;\; if w.x + b < 0 \end{cases} \tag{3.11}$$

The point above or on the hyperplane will be classified as class $+1$, and the point below the hyperplane will be classified as class -1. SVM classifier amounts to minimizing an expression of the form given below:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - \mathrm{yi}(\mathrm{w.x_i} - \mathrm{b}))\right]$$

The reason to choose this classifier in the present research is its robustness. It provides an optimal margin gap to separate hyperplanes and it gives a well-defined boundary for easy classification.

3.3.3 Random Forest

Random Forest (RF) machine learning algorithm is also known as a tree-based ensemble learning, which creates a forest of many decision trees. RF ensures that the behavior of each decision tree produced is not too correlated with the behavior of any other decision tree in the model. This final prediction can simply be the mean of all the observed predictions [9]. Therefore, the different decision trees obtained by using the RF algorithm are trained using different parts of the training dataset, which is the reason behind its unbiased nature and superior prediction accuracy.

4 Experimental Results and Discussions

For conducting the practical implementation of this case study, we used *Jupyter Notebook* with Python version 3.8. Various Python libraries have been used for data pre-processing and visual representation such as *pandas, numpy, scrapy, matplotlib, seaborn, spacy,* etc. For training and testing of machine learning classifiers, the corpus is divided into two subsets with a train-test split of 75–25% respectively. To evaluate the performances of the classifiers, the main parametric metrics employed in this research are *accuracy, precision, recall, F-measure, true positive* and *false negative*. In classification problems, precision (also called positive predictive value) is the fraction of relevant instances that are retrieved, while recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Accuracy of the model can be defined as the ratio of the correctly labelled attributes to the whole pool of variables. F-measure is a weighted average of precision and recall. As *ASIN* and review ratings are two important attributes available in our dataset, therefore we explored their relationship visualized in Fig. 7 below.

The above figure clearly reveals that the most frequently reviewed products in our case study have their average review ratings above 4.5. On the other hand, *ASINs* with lower frequencies in the bar graph have their corresponding average review ratings below 3. For analyzing the classification performance of machine learning models, we applied *Naïve Bayes*, *Support Vector Machine* and, *Random Forest* algorithms to the pre-processed dataset. The performance evaluation results of machine learning classifiers using *Skip-gram* and *TF-IDF* feature extraction techniques are shown in Table 3 and Table 4 respectively.

The results of the Table 3 above, clearly show that the *Random Forest* classifier achieves maximum accuracy in skip-gram model.

Table 4 reveals that *TF-IDF* significantly improves the accuracy along with other important parameters of *Naïve Bayes* and *Support Vector Machine* classifiers but does not perform well enough with *Random Forest*. After comparing the results of Table 3 and Table 4, it is clear that the *TF-IDF* approach improves the accuracies of *Naïve Bayes* and *Support Vector Machine* classifiers by 6% and 3.9% respectively but deteriorates the accuracy of *Random Forest* by a margin of 0.8%. Figure 8 compares

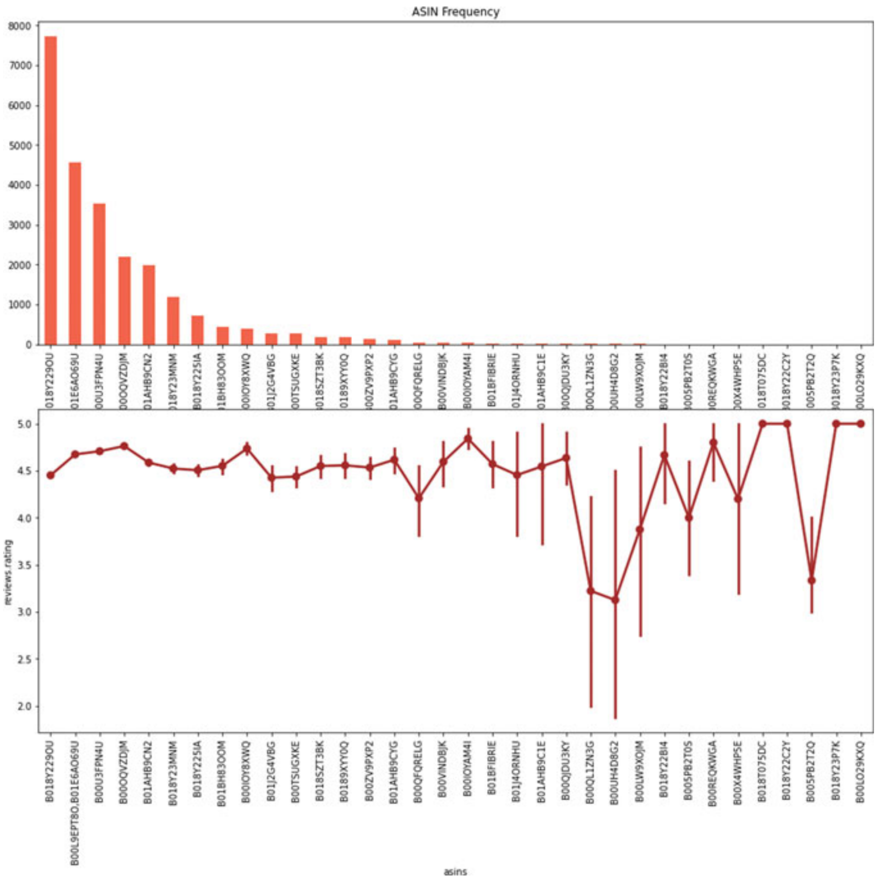


Fig. 7 Relationship between *ASIN* and review ratings

Table 3 Classifiers comparison using Skip-gram feature extraction method

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	89.3	0.893	0.892	0.892
Random Forest	95.2	0.952	0.95	0.951
Support Vector Machine	93.3	0.916	0.93	0.927

Table 4 Classifiers comparison using *TF-IDF* feature extraction method

Classifier	Accuracy	Precision	Recall	F-measure
Naïve Bayes	95.33	0.953	0.952	0.953
Random Forest	94.63	0.946	0.941	0.946
Support Vector Machine	98.61	0.991	0.99	0.990

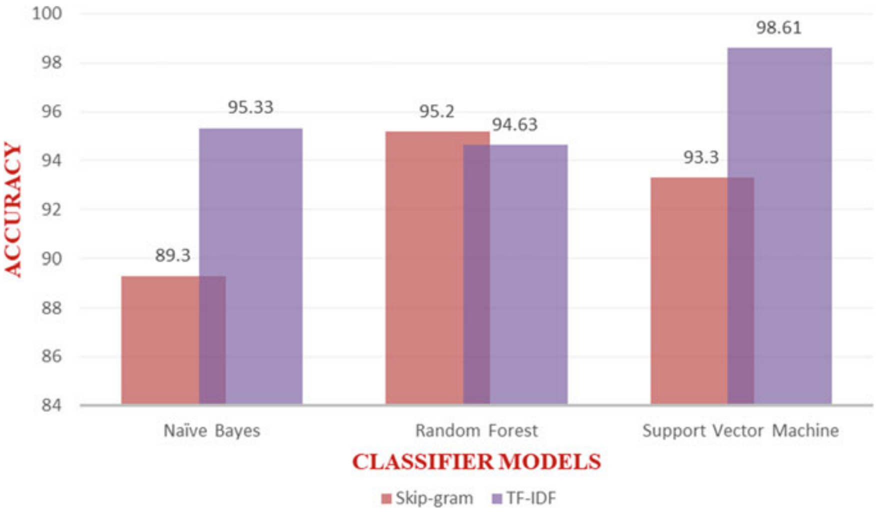
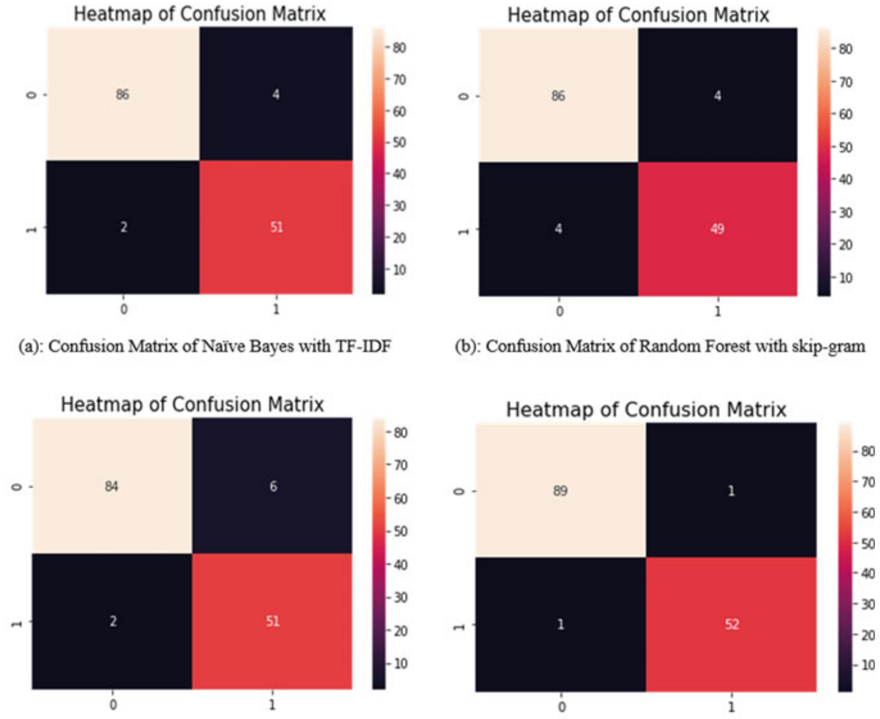


Fig. 8 Accuracies comparison of *Skip-gram* and *TF-IDF* techniques

the classification accuracies of individual classifiers against the *Skip-gram* and *TF-IDF* feature extraction techniques.

As shown in the above figure, it is evident that in the case of *Naïve Bayes* classifiers, the classification accuracy obtained using *TF-IDF* is better than the value obtained using the *Skip-gram* technique. The *Naive Bayes* algorithm follows a probabilistic approach, where the attributes are independent of each other. Therefore, when the analysis is performed using a single word (unigram) and double word (bigram), the accuracy value obtained with *TF-IDF* is comparatively better than that obtained using *Skip-gram*. Similarly, it is clear that in the case of *Random Forest*, the classification accuracy value obtained using the *Skip-gram* technique is a little better than the value obtained using *TF-IDF*. As we know, *Random Forest* is an ensemble tree-based classifier and it aggregates the output obtained from different decision trees, the *Skip-gram* model which can predict the source context words given a target word gives better results. In the case of *Support Vector Machine* classifiers, the classification accuracy attained using *TF-IDF* is better than that obtained using the *Skip-gram* approach. *Support vector machine* is a non-probabilistic linear classifier and the trained classifier is used to find hyperplane for dataset separation, the *TF-IDF* which analyses the corpus word by word gives better results as compared to the *Skip-gram* model. Figure 9 shown below presents the *Heatmaps* of confusion matrices obtained.

The above figure depicts the four best confusion matrices obtained from various classifiers, which is a summary of the prediction results in our classification problem. The part (a) above, shows the trained model predicts *True Negative* of 86%, *True Positive* of 51%, *False Positive* of 4% and *False Negative* with 2%. Therefore, 86% and 51% are the correct predictions and 4% and 2% are incorrect predictions. We



(a): Confusion Matrix of Naïve Bayes with TF-IDF

(b): Confusion Matrix of Random Forest with skip-gram

(c): Confusion Matrix of Random Forest with TF-IDF

(d): Confusion Matrix of SVM with TF-IDF

Fig. 9 Heatmaps of the confusion matrices

can see that we have a good ratio of correct predictions. Similarly, we can interpret the confusion matrices of other classifiers. Note that the SVM with TF-IDF shows maximum accuracy, as shown in part (d) of Fig. 9.

5 Conclusions

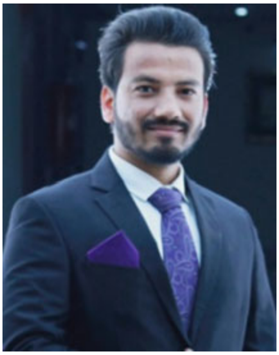
Analysis of sentiments is crucial for any online retail business enterprise to understand the opinions and feedbacks of its customers. This case study analyses the sentimental polarities of the scrapped user-reviews of *Amazon* customers through machine learning classifiers. The dataset used in the present chapter is collected from the *Amazon* review portal using the well-known *scrappy* library available in Python. We scrapped 300,000 mobile phone reviews from *Amazon* review portal for various international brands. This unstructured dataset had to be preprocessed first to convert it into a legitimate form so that machine learning classifiers can process it smoothly. The null and missing values were dealt with *imputation* technique and several preprocessing techniques like stop word removal, spelling corrections, stemming, special character handling, etc. were also exercised. Now, this preprocessed dataset which was in text form needed to be converted into numerical scores before submitting it into machine learning models. We employed two techniques for this preliminary step: *Skip-gram* and *TF-IDF*. After the above treatments, we put the processed dataset to three different classifiers: *Naïve Bayes*, *Support Vector Machine*, and *Random Forest*. The above-mentioned machine learning classifiers are then evaluated over some standard parameters say, accuracy, precision, recall, F-measure, true positive and false negative. The empirical results found to be very satisfactory.

The present sentiment analysis case study of the *Amazon* reviews can be considered a kind of novel work where various machine learning classifiers have been compared against two different feature engineering techniques. Empirical results reveal that all models are able to classify the user reviews into negative and positive classes with relatively high accuracy and precision. Calculated results exhibit that the *Support Vector Machine* model achieved the highest classification accuracy (98.61%) with *TF-IDF* feature extraction method. Next, the *Naïve Bayes* model with *TF-IDF* achieved the classification accuracy of 95.33%. And, the *Random Forest* model with the *Skip-gram* technique acquired 95.2% accuracy at the third position.

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of Twitter data. In: Proceedings of the Workshop on Languages in Social Media (LSM 2011), pp. 30–38. Association for Computational Linguistics (2011)
2. Alsaeedi, A., Khan, M.Z.: A study on sentiment analysis techniques of Twitter data. Int. J. Adv. Comput. Sci. Appl. **10**(2), 361–374 (2019)
3. Bagheri, A., Saraee, M., Jong, F.D.: Care more about customers: unsupervised domain-independent aspect detection for sentiment analysis of customer reviews. Knowl.-Based Syst. **52**, 201–213 (2013)
4. Bai, B.M., Mangathayaru, N., Rani, B.P.: An approach to find missing values in medical datasets. In: Proceedings of the International Conference on Engineering & MIS 2015, Istanbul Turkey, Article No.: 70, pp. 1–7 (2015). https://doi.org/10.1145/2832987.2832991
5. Barbosa, L., Feng, J.: Robust sentiment detection on Twitter from biased and noisy data. In: Conference: COLING 2010, 23rd International Conference on Computational Linguistics, Beijing, China, Poster Volume, pp. 36–44 (2010)
6. Bhagat, A., Sharma, A., Chettri, S.K.: Machine learning based sentiment analysis for text messages. Int. J. Comput. Technol. **7**(6), 103–109 (2020)
7. Celin, S., Vasanth, K.: ECG signal classification using various machine learning techniques. J. Med. Syst. **42**(12), 1–11 (2018). https://doi.org/10.1007/s10916-018-1083-6
8. Chaffey, D.: Amazon's business strategy, revenue model and culture of metrics: a history. Digital Marketing case studies (2020). https://www.smartinsights.com/digital-marketing-strategy/online-business-revenue-models/amazon-case-study/. Accessed 25 July 2020
9. Dimitriadis, S.I., Liparas, D.: Alzheimer's disease neuroimaging initiative. How random is the random forest? Random forest algorithm on the service of structural imaging biomarkers for Alzheimer's disease: from alzheimer's disease neuroimaging initiative (ADNI) database. Neural Regen. Res. **13**, 962–970 (2018). http://www.nrronline.org/text.asp?2018/13/6/962/233433

10. Fang, X., Zhan, J.: Sentiment analysis using product review data. J. Big Data **2**(5), 1–14 (2015)
11. Ghag, K.V., Shah, K.: Conceptual sentiment analysis model. Int. J. Electr. Comput. Eng. **8**(4), 2358–2366 (2018)
12. Gräbner, D., Zanker, M., Fliedl, G., Fuchs, M.: Classification of customer reviews based on sentiment analysis. In: Fuchs, M., Ricci, F., Cantoni, L. (eds.) Information and Communication Technologies in Tourism 2012, pp. 460–470. Springer, Vienna (2012). ISBN 978-3-7091-1141-3
13. Krishna, P.P., Sharada, A.: Word embeddings—skip gram model. In: Gunjan, V., Garcia Diaz, V., Cardona, M., Solanki, V., Sunitha, K. (eds.) ICICCT 2019—System Reliability, Quality Control, Safety, Maintenance and Management. ICICCT 2019. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-8461-5_15
14. Liu, B.: Sentiment Analysis and Opinion Mining. Morgan&Claypoo, SanRafael, CA (2012)
15. Madhu, G., Rajinikanth, T.V.: A novel index measure imputation algorithm for missing data values: a machine learning approach. In: IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, pp. 1–7 (2012). https://doi.org/10.1109/iccic.2012.6510198
16. Mäntylä, M.V., Graziotin, D., Kuutila, M.: The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. Comput. Sci. Rev. **27**, 16–32 (2018). ISSN 1574-0137
17. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. J. **5**(4), 1093–1113 (2014)
18. Mitchell, R.: Web Scraping with Python, 2nd ed. O'Reilly Media, Inc. (2018). ISBN: 9781491985571
19. Mozetič, I., Grčar, M., Smailović, J.: Multilingual Twitter sentiment classification: the role of human annotators. PLoS ONE **11**(5), (2016). https://doi.org/10.1371/journal.pone.0155036
20. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retr. **2**(1–2), 1–135 (2008)
21. Prabowo, R., Thelwall, M.: Sentiment analysis: a combined approach. J. Inform. **3**(2), 143–157 (2009)
22. Saad, S., Saberi, B.: Sentiment analysis or opinion mining: a review. Int. J. Adv. Sci. Eng. Inf. Technol. **7**(5), 1660–1666 (2017)
23. Sidorov, G.: Vector space model for texts and the tf-idf measure. In: Syntactic n-grams in Computational Linguistics. Springer Briefs in Computer Science. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14771-6_3
24. Su, X., Greiner, R., Khoshgoftaar, T.M., Napolitano, A.: Using classifier-based nominal imputation to improve machine learning. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) Advances in Knowledge Discovery and Data Mining. PAKDD 2011. Lecture Notes in Computer Science, vol. 6634. Springer, Berlin, Heidelberg (2011)
25. Sweilam, N.H., Tharwat, A.A., Moniemc, N.K.A.: Support vector machine for diagnosis cancer disease: a comparative study. Egypt. Inform. J. **11**(2), 81–92 (2010)
26. Tyagi, P., Tripathi, R.C.: A review towards the sentiment analysis techniques for the analysis of Twitter data. In: Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) (2019)



Er. Koushal Kumar has 8 years of teaching experience at both graduate and postgraduate levels. He attended several professional development seminars, national and international conferences, Workshops, and faculty development programmes. He also cleared India's prestigious UGC-NET examination in 2013. He has been actively performing reviewer duties in an IEEE journal (Internet of things) since 2017. He has more than 130 citations and 12500 reads on world most reputed researcher platform Research Gate. He has guided many dissertations at the master's level. He is a keen researcher and has published many research publications in international journals and conferences. His research interests include Machine Learning, Data science, Internet of Things, Fog computing and Deep Learning. He is the author of a book titled "*Rules extraction from trained neural networks using decision trees*."

Dr. Bhagwati Prasad Pande has experience of more than 8 years in teaching undergraduate and postgraduate courses of Information Technology and Computer Science. He holds Ph.D. degree in IT and his research interests include Information Retrieval, NLP, Optimization Techniques and Emoticons. He also cleared India's prestigious UGC-NET examination in 2013. He published many research articles in reputed publication houses like John Wiley & Sons, Oxford University Press and Glottotheory. He recently registered copyright in *The United States Copyright Office*. He also served a few journals as a reviewer. He has guided a number of dissertations at the master's level and industrial projects at the undergraduate level.