

# Modern Machine Learning: Applications and Methods



Xabier Cid Vidal, Lorena Dieste Maroñas, and Álvaro Dosil Suárez

**Abstract** Machine Learning (ML) is now omnipresent in most fields of human knowledge. Despite this, it remains mysterious and, to some extent, a black box of statistical methods many have heard of, but few know in detail. Although arguably some minimal degree of experience in statistics is needed to master ML, it is not impossible at all to understand many important concepts with no mathematical formalism. In this chapter, we aim at providing a description as complete as possible of what ML is and which is the main methodology used nowadays, from the most basic methods, such as support vector machines, to the very popular neural networks, which appear in some of the trendiest applications of ML. The second part of the chapter will revolve around how ML is used in different areas of knowledge, from industry to basic science. This will be done, once again, with no mathematical formalism, but instead referring to the methods presented in the first part of the chapter. Our goal is that, once a reader has gone through the full chapter, they have a basic idea of what happens behind the scenes when a ML algorithm is run, which are the main types of algorithms that one can use, and they are familiar with the extremely wide range of applications of ML in the modern world.

## 1 Machine Learning in Modern Days

It does not matter whether you work in industry, in science or as an engineer, Machine learning (ML) is becoming ubiquitous in almost every field associated with human progress (Brink, 2016). ML has been among us from many years ago, in applications such as image recognition, product recommendation, fraud detection... even

---

X. Cid Vidal (✉) · L. Dieste Maroñas  
Instituto Galego de Física de Altas Enerxías (IGFAE), Universidade de Santiago de Compostela,  
15782 Santiago de Compostela, Spain  
e-mail: [xabier.cid@usc.gal](mailto:xabier.cid@usc.gal)

L. Dieste Maroñas · Á. Dosil Suárez  
Triple Alpha Innovation, Cuns, 4, 15286 Outes, A Coruña, Spain

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022  
D. Carou et al. (eds.), *Machine Learning and Artificial Intelligence  
with Industrial Applications*, Management and Industrial Engineering,  
[https://doi.org/10.1007/978-3-030-91006-8\\_2](https://doi.org/10.1007/978-3-030-91006-8_2)

without us knowing it, and now is one of the boosting technologies of the 4th industrial revolution. The main goal of this chapter is making you understand why this is the case and quickly guiding you through the main methodology of ML, while providing examples of applications in several areas of knowledge. ML is defined by the Cambridge dictionary as “the process of computers changing the way they carry out tasks by learning from new data, without a human being needing to give instructions in the form of a program”. While this is a relatively simple definition, as we shall see it gathers a lot of important information about how ML works. For the moment, let us simply say ML is a group of powerful statistical techniques which, using the simplistic concept of “learning through examples”, can achieve tasks as amazing and diverse as allowing a car to drive itself, optimizing the performance of an assembly line or allowing the discovery of a new subatomic particle.

You might have used ML in the past, even without being aware of it. Imagine you have a doughnut production line, and you want to eliminate the faulty ones. You look at a few examples of those you consider not suitable for selling, and you visually find some patterns: they have a blackish color, their hole is not as perfect as it should be, or their size is not appropriate. Based on these criteria, you write down some guidelines so your workers can discriminate against those doughnuts you do not like to sell. As simple as it is, this is ML. You have learned from examples on your own data, found patterns, and developed a method to select what you want in independent data sets! This kind of task is called “classification”, and it will be reviewed in the next section. As a different approach, say you want to study how the average temperature in different world regions of the same longitude from the northern hemisphere changes with latitude at the same time of the year. As a rule of thumb, you could expect the temperature goes down with latitude (the more to the North, the lower the temperature). A scatter plot of different measurements (temperature vs latitude) can be very helpful and easily show such a correlation. You take one step further, assume the correlation is linear, and perform a regression analysis to model it. Using this model (no need to enter in this example on how accurate this is) and given a latitude or temperature, you can estimate which is the other expected magnitude in the pair. And surprisingly, or not again, that is ML! You have taken some data, used some simple statistics to understand it, and provided a model you can use to produce independent predictions. As we shall see, this is what we generally call “regression”.

This chapter is divided into two main parts. In the first one, we will provide a definition of the types of learning used in ML, such as supervised and unsupervised learning. Furthermore, we will briefly summarize some of the most popular methodologies, without providing any mathematical detail. This will include concepts such as boosting, support vector machines, or the very popular artificial neural networks (and associated methods). In the second part, we will cover the main uses of ML in many different aspects of the modern world, from engineering, manufacturing, and finance to user interface, medicine, and science.

## 2 How Can We Teach a Machine to Learn?

When we talk about ML we are referring to the process through which a computer learns how to solve a problem. This learning can be classified in two main different types of learning: supervised and unsupervised (Géron, 2019).

### 2.1 *Supervised Learning*

In supervised learning a human directs the algorithm learning by giving it examples of the problem and the desired solutions (Pedregosa et al., 2011). Basically, the person must provide to the algorithm a dataset well labeled, i.e., the training examples must be tagged with the right answer, so the algorithm can learn what are the properties that define each label are. The right answer can be a class from a classification problem or a specific number as a result of a regression. After that, the supervised learning algorithm examines the training data so, when the machine is provided with a new set of unlabeled data, the algorithm will be able to produce a correct outcome.

There are dozens of supervised learning algorithms, but mostly all of them can be included in one of these groups: classification or regression. The goal of classification is to label an input, although regression aims to predict a quantity.

#### 2.1.1 **Classification**

Classify means essentially to group some elements by their properties. Humans like to classify everything since it creates order that helps us to understand the world. So, this is a very interesting feature of ML for us. In ML classification problems, the goal will always be to group the inputs in different categories depending on their characteristics. A typical example of classification is the email spam detection. In this problem, there are only two different output classes, spam or legitimate. Since we only have two options, this is called a binary classification problem. Every time we receive a new mail in our inbox, the ML algorithm must classify it as spam or not-spam. However, we can find classification problems with any number of classes. For example, if we want to classify a book by its genre, we have a lot of different output classes: romance, thriller, biography, or adventure.

There are tens of classification algorithms to choose from, but here we are going to mention only some of the most important ones: Logistic Regression, Decision Trees or Neural Networks (NN), and Support Vector Machines (SVM).

### **2.1.2 Regression**

Unlike classification, in regression problems the result is a number. In this kind of problem, the goal is to predict a numerical property of the input based on its properties. One of the most used examples to explain this method is the house price prediction. As its name indicates, the algorithm must predict the market price of a house based on some of its inputs such as its size, date of construction, location, and many others. Some of the main methods for regression are lineal and no-lineal regression, Decision Trees Regression or Neural Networks, and Support Vector Machines.

## **2.2 *Unsupervised Learning***

Opposite to supervised learning where a human must teach the algorithm how to perform, the most defining characteristic of an unsupervised learning algorithm is that it must figure the problem and find the solution by itself. That is, it must recognize the data structure and what the outcome of the algorithm is supposed to be. These kinds of algorithms are very useful to discover hidden patterns in data or to feature learning and, also, since they do not need human intervention, they can improve their results over time by themselves.

One of the main examples of unsupervised learning is clustering. The main goal of a clustering model is to group the elements by its features. Objects in the same groups will be similar, while objects in different groups are dissimilar. Two elements of the same group do not need to have identical values of a feature but will be like a third element of another group.

The ML models create a spatial representation of the features and then evaluate the similarity of the elements by the distance between them. The smaller the distance, the similar the objects are and, hence, they probably belong to the same group. In the end, the result is that the distances between neighbors within one cluster are smaller than between objects from different clusters.

## **2.3 *Others***

### **2.3.1 Reinforcement Learning**

We also must mention a fourth type of learning called reinforcement learning (RL) (Osiński & Budek, 2018). In this case, the algorithm must operate into an environment without any instructions of how to do it. Supervised and unsupervised algorithms need to be fed with data first; nevertheless, this step is completely skipped in reinforcement learning. Instead, the data is generated from trial and error during training, and it is tagged at the same time.

The goal of these algorithms is to perform a task as well as they can. The algorithm is connected to an environment where it can accomplish the task, and the only interaction with the trainer is that the algorithm receives a feedback or reward at certain times every time it gives a try. So, the aim of the algorithm is to increase the reward as much as it can, creating long-term strategies.

There are two kinds of reinforcement learning methods: positive and negative. The positive reinforcement maximizes the actions that increase the reward. This type of reinforcement maximizes performance and sustains change over a longer period, but excessive reinforcement may lead to over-optimization, affecting outcomes. Negative reinforcement, on the other hand, is defined as any alteration in behavior caused by a negative condition that should have been avoided. Thus, the disadvantage of this method is that it provides enough to achieve the minimum objectives, but does not provide the optimal solution. Some applications of RL can be:

- Robotics for industrial automation.
- Creation of training systems that provide custom instructions.
- Business strategy planning.
- Aircraft control and robot motion control.
- Machine Learning and data processing.

### 3 Different Algorithms and Methods Used in ML

At the moment of writing this book, we can count tens of ML methods, being even difficult to enumerate all of them. To avoid leaving the scope of the book, we are going to review the main methods that are being used nowadays.

Before we get down to business, some terms are worth describing for a better understanding of the chapter. A dataset is a collection of objects, and an object can be anything we want to use to solve a problem. For example, different flowers, emails, different animals... The objects are described by features. In the flower example, some features would be the color, the size, and different properties we can use to describe them. These features can be of different types, and the most used ones are numerical, categorical, and binary. The numerical type is used when the features are described by numbers, as, for example, the age of a flower. Categorical features usually describe a property that is not a number, as the color of the flower. Finally, we use a binary feature when we want to describe a property with only two options, like true or false.

All the objects and their features together form what we call a dataset. During the ML model training, the dataset will be divided into two smaller sets, the train set, those objects we will use for training our model, and the test set, those objects we will use to “test” the final model.

Once the two datasets with their objects and features are defined, the next step is to describe the problem we want to solve and, afterwards, the algorithm we will use. The algorithm is a mathematical application that will map our dataset to the

solution we want to achieve. There are a lot of different algorithm options we can choose from, some of the most known ones are Logistic Regression, Support Vector Machines, Neural Networks, and many others.

Once we run the algorithm on data, we create the ML model, i.e., the model is the algorithm parametrized after the algorithm training. It represents the rules, numbers, and any other data structures required to make predictions. The model gives us an approximation of the result, the closer the result is to the real one is a measurement of how good our model is. One of the most difficult parts of creating a ML solution is choosing the right algorithm, which means applying the right assumptions about the data we have.

After achieving a good enough result during training, it is time to evaluate the model and check how good it really is. In order to do that, we need to evaluate it using a set of objects different from those used for training. This dataset is called the test set.

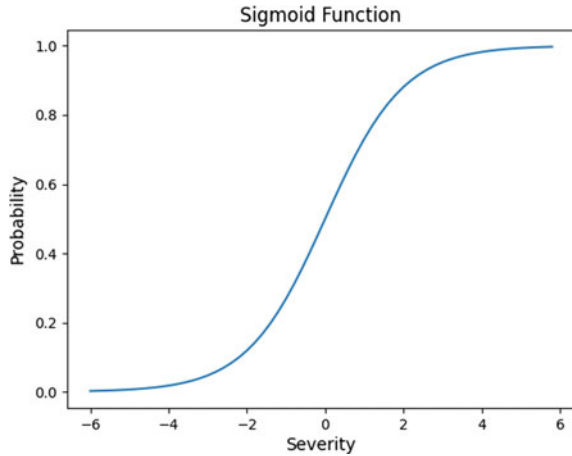
At this moment, some training errors can arise. Two of the most common problems are overfitting and underfitting. Overfitting means that the model can approximate the training data almost perfectly, although when tried with new data the accuracy decreases. In this case, the model is said to have high variance. The reason is that the model learns the noise in the training set. This noise can obscure the true relationship between features and the response variable. Overfitting is more likely when there are many features available, or a complex model is used. The more the features, the bigger the chance of discovering a fake relationship between the features and the result. Complex models develop more complex hypotheses about the relationship between features and the result. When a model underfits it is not able to fit the training data and it is said to have high bias. This means that the model is not complex enough, in terms of the features or the type of model being used.

### ***3.1 Logistic Regression***

Logistic regression is an ML technique from the field of statistics. From its name one might think that it is an algorithm to be applied in regression problems, however it is a method for classification problems, in which a binary value between 0 and 1 is obtained. Thus, logistic regression allows to establish the possible relationship between a dependent variable with one or several independent variables through a logistic function that determines the probability that the dependent variable is related to the independent variables (Osiński & Budek, 2018). The independent variables represent the features of the objects we want to classify, and the dependent the variable, the class the object belongs in.

The linear regression model has a quantitative variable as the output variable, nevertheless for classification we need a qualitative output. To exemplify this, let us think of a variable called “severity” that indicates if a patient’s condition in a hospital is “serious” or “not-serious”. Thus, we have two groups, 0 = “not-serious” and 1 = “serious”. Logistic Regression manages to transform the output variable with the

**Fig. 1** Graphic representing a sigmoid function



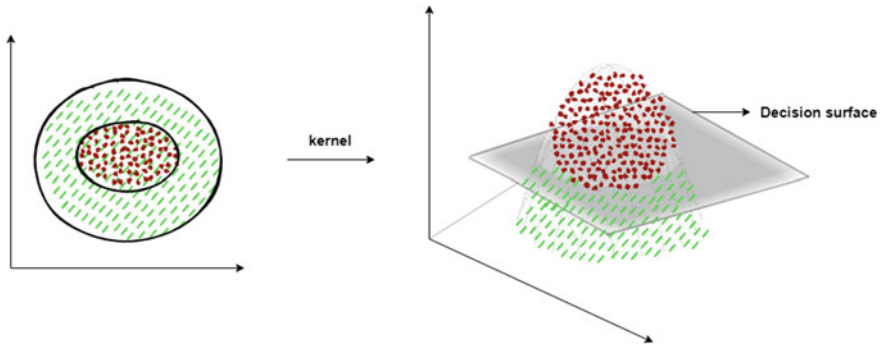
logistics operator, also called sigmoid function. This mathematical operator converts the independent feature in a probability, ranging between 0 and 1 and representing how likely an instance is of being 0 = “not-serious”. The sigmoid function is an S-shaped curve with values between 0 and 1 (see Fig. 1).

This probability must be translated into binary values, for which a threshold value is used. For probability values above the threshold value the statement is true and below it is false. A true positive is an object that was classified in the correct class ‘0’ and a false positive is an object that was classified in the class ‘0’ but it belonged in the class ‘1’. With false negatives the idea is the same, an object that belongs to the class ‘0’ is wrongly classified as class ‘1’. In this way we can use the same structure as linear regression. We are simply converting the response variable, which is qualitative, into a probability, which is quantitative.

Logistic regression is a technique widely used by data scientists because of its efficiency and interpretability. In addition, it does not require extensive computational resources for training or execution. The performance of logistic regression, like linear regression, is better when using attributes related to the output. It is also important to eliminate features that show great multicollinearity with each other. Therefore, the selection of these features before the training of the model is a key. Because the expression that makes the decision is linear, the model is not able to solve non-linear problems directly and it is better to use other models such as decision trees.

### 3.2 Support Vector Machines

Support Vector Machine (VVM) can be defined as a supervised method mostly used to solve classification problems, but also applied in regression problems. In this last case, we talk about Support Vector Regression (SVR). When working on classification problems the algorithm learns how to separate classes creating decision



**Fig. 2** Example of a kernel transformation. (Left) the data as we have it in a two dimensions space. (Right) The data after some transformation to be linearly separable in a three-dimension space

boundaries or hyperplanes (Scikit-learn, n.d.). The points that define the maximum margin of separation from the hyperplane are called support vectors. They are called vectors, instead of points, because they have as many elements as there are dimensions in our input space.

Sometimes it is impossible to find a hyperplane to separate two classes, so these two classes are not linearly separable. To tackle this problem, we have to use a kernel. This consists of inventing a new dimension in which we can find a hyperplane that separates the classes. In Fig. 2 you can see an example of how the kernel works.

Basically, we transform the two-dimensional space without a linear separation, into a three-dimensional space. In this new space we clearly observe a plane separating the two classes.

As the reader may record, we said that a regression problem is based on looking for the curve that models the trend of the data and, according to it, predicting any other data in the future. In principle this definition is not very compatible with the support vector machines but doing some simple changes we can get it ready. The SVR uses the same principle of SVM and if the problem is not linear, it adds a new dimension to try to find a linear separation.

### 3.3 *Decision Trees and Random Forests*

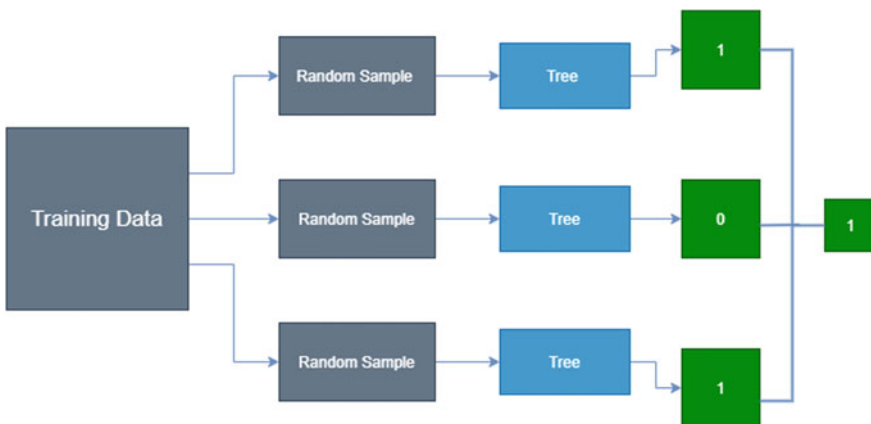
A decision tree is a method to separate a set of objects into several distinct subsets through binary decisions about the properties of its elements. Each binary decision consists of a comparison involving one or more variables, and it is taken in the Decision nodes (Gupta, 2017). The output of the node are two new decision nodes or two terminal nodes and depending on the result the child goes to the left or to the right. The classification starts at the root of the tree, in the node called Root node and finishes on the Terminal nodes that do not split and that determine the result of the operation.



The main advantage of trees is that they represent “rules” which can be understood by humans with the advantage that the knowledge is generated by the tree itself and not based on the premise of an expert on the subject. Another advantage of this algorithm is that it is easy to use, everybody can understand the way it works and plan a model to solve problems. It is versatile, since a lot of different problems of different fields can be solved using this modeling. Moreover, compared with other algorithms it requires less effort for processing the data, this means no normalization, not filling missing data and not scaling among others. However, this type of solution also has disadvantages, mainly because it requires long training times for the models and is often unsuitable for predicting continuous values, and is mainly used for classification problems.

There are a lot of options in order to improve the algorithm and achieve a better result for the model, one of the most known ways of improving a Decision tree is the combination of several of them called ensemble learning and resulting in a new method called Random Forest (Glen, 2019). Ensemble-type methods are made up of a group of predictive models that allow better precision and model stability to be achieved since the group compensates for the errors in the predictions of the individual ones. Also, Random Forests use a technique called bagging, which means that different trees see different portions of the data. This way we can prove that each tree is different from the others. At the end we have many versions of the algorithm trained on different subsets of the dataset so that the biases cancel each other. Although each decision tree has a high variance, when combining them the total variance is low as each decision tree is perfectly trained and therefore the outcome does not depend on one decision tree but on multiple decision trees.

An example of the process can be seen in Fig. 3, where a schema of how a Random Forest algorithm with three Decision Trees would work is shown.



**Fig. 3** Schema of how Random Forests works. It combines three different regular Decision Trees to obtain the final result using soft-voting

Random Forest can be used both for classification and regression solutions. The process is similar for both, being the combination of the result the main difference. In classification problems, the results of decision trees are often combined using *soft-voting*; however, we can also come up with our own way of combining the results of a random forest. In soft-voting, more importance is given to some of the decision trees of the forest. Regarding regression problems, the most common way to combine the results of the decision trees is by taking their arithmetic mean.

### 3.4 *Boosted Decision Trees*

Another improvement of the performance of a Decision tree can be what we call BDT or Boosted Decision Tree. This is another ML technique used for regression analysis and for classification problems. The name boosting is an approach of creating a highly accurate capability of prediction by combining weak and imprecise predictors. BDT algorithms consist of a set of decision trees, each with an assigned weight. Each tree is created iteratively using the output of the previous tree, and the tree's output is given a weight relative to its accuracy. To calculate the final output, the outputs of the trees are linearly combined with their weights. After each iteration every data sample is weighted proportionally to the frequency of its misclassification. The final goal is to minimize the loss function. A schema of how the model uses the residuals of the previous model, and the combined solution is shown in Fig. 4.

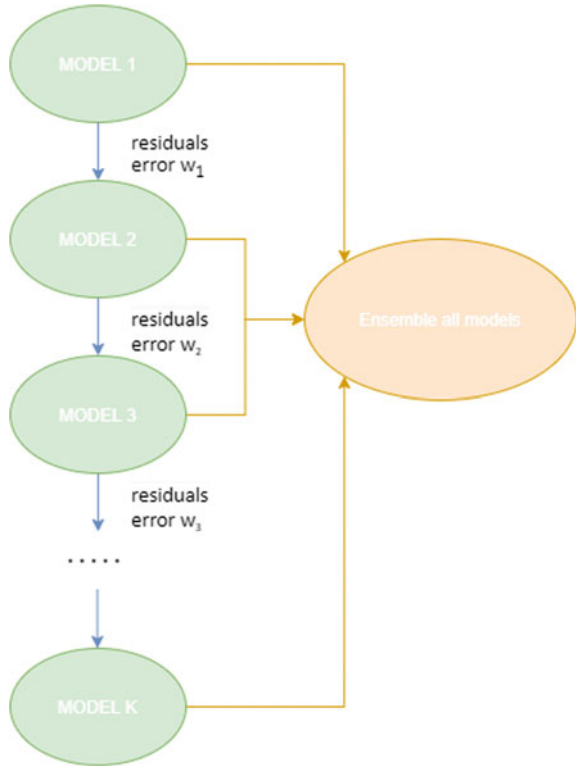
This new method brings some new parameters that we will have to take into consideration in order to create the best possible model. The most important ones are:

- **Loss function:** the loss function indicates the difference between the real data and the prediction.
- **Learning rate:** the learning rate indicates the step to adjust data weights in each iteration. The smaller the more precise, but also slower. Normally this parameter takes values close to 0.1.
- **Subsample size:** as data samples are randomly selected in each iteration, the subsample size is the parameter that indicates how many samples to train in each new tree.
- **Number of trees:** the total number of trees that can be created to solve the problem. Usually a big number is better, but it could lead to overfitting.

As every other algorithm BDT has advantages and problems. We can highlight that this algorithm is fast, for both training and prediction, easy to tune and it is not sensitive to scale, allowing a mix of numerical or categorical features. Also, it has good performance, and it is very commonly used so there is a strong community behind it. On the other hand, BDT is an algorithm sensitive to overfitting and noise.

So far, Random Forest and BDT seem similar, but they have two important differences. The first one is related to how the trees are built. While random forests build each tree independently, the BDTs build one tree at a time, and it introduces weak

**Fig. 4** Schema of how boost decision tree works. Example using k models combined

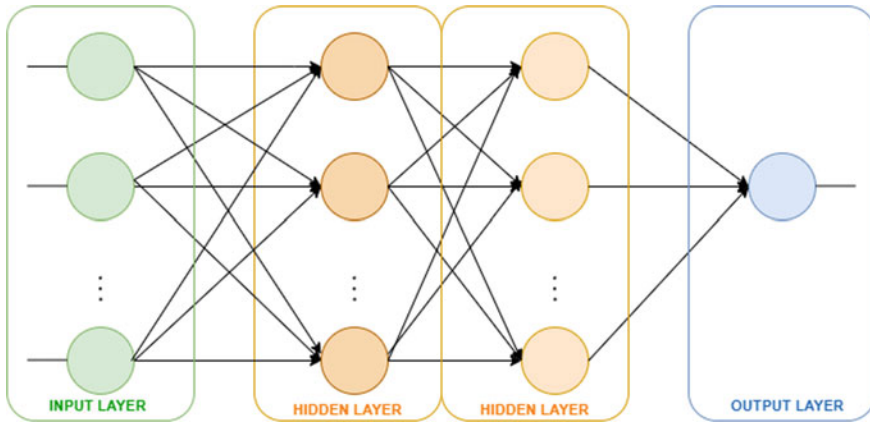


learners to improve the deficiencies of the previous ones. The second one is the way they combine the results. Random forests combine the results at the end of the process, while BDT combines results after each iteration. Hence, BDT can result in higher performance. Nevertheless, the final performance will depend on the problem. In a problem with very noisy data, the BDT may not be a good option as it can result in overfitting. Also, it is usually more difficult to tune than Random Forests.

There are many types of boosting algorithms, some of the more common ones are AdaBoost or Adaptive Boosting, Gradient Boosting, and XGBoost. The main difference among the algorithms is the loss function they use to calculate the weights of the trees and the data.

### 3.5 Neural Networks

Neural Networks try to mimic the network of neurons in a human brain and its behavior. Basically, the structure of a neural network is a bunch of neurons connected among them and working with a common goal, without having individually a specific task. The neurons can be grouped in layers that are connected among them. Typically,



**Fig. 5** Neural Networks graph representation. Example with 2 hidden layers

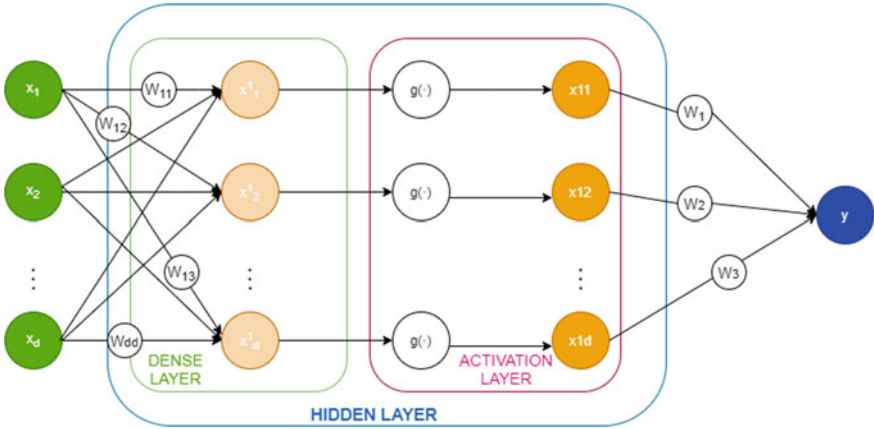
the NNs are formed by three layers. An input layer, with neurons representing the input fields (Goodfellow et al., 2016). One or more hidden layers, and an output layer, with one or more neurons representing the result. An important characteristic of neural networks is that every input to a neuron is weighted, which is a crucial aspect when training the network. An example of a diagram of a neural network can be seen in Fig. 5.

### 3.5.1 Deep Neural Networks

One of the multiple possible classifications of the NN is by the number of their hidden layers. Using this classification, we find Shallow neural networks that have only one hidden layer between the input and output, and Deep Neural Networks (DNN), that have multiple hidden layers. One example of DNNs is the Google LeNet model for image recognition that has 22 layers (Alake, 2020).

In DNNs, each network layer can learn increasingly complex features of the data. Each layer integrates a deeper level of knowledge. A neural network with five layers can learn more complex features than one with only two layers. In this kind of networks, the hidden layer is divided into two phases. The first is called dense layer, and it applies a nonlinear transformation to the input data. The second phase improves the model with a derivative function, in what is called the activation layer. The neural network repeats these two steps many times until the result is similar enough to the one desired. Each repetition of this two-phase is called an iteration. An example diagram of a neural network can be seen in Fig. 6.

Deep neural networks provide higher accuracy in many tasks compared to any other method, for example, in object detection or speech recognition. Also, they can learn autonomously, without any human knowledge transference.



**Fig. 6** Neural Networks graph representation of the hidden layer. The  $W_{ij}$  are weights that represent the importance of every input when feeding the next neuron. Example with  $d$  features

Deep learning is called this way because it makes use of deep neural networks. This type of learning can be supervised, semi-supervised, or unsupervised.

### 3.5.2 Convolutional Neural Networks

A Convolutional Neural Network (ConvNet/CNN) was born as a necessity of improving the DNN when they interact with images (IBM Cloud education, 2020).

A color image can be represented as a mix of three matrices, one for each main color, Red, Green, and Blue, in which the number of pixels indicate the size of the matrix. For example, a color image with  $34 \times 34$  pixels is represented as three matrices of  $34 \times 34$  units each. In Fig. 7 we can see how a regular picture is represented by three images, Red, Green, and Blue.

This means that images are high-dimensional objects that demand powerful techniques for efficient processing. With all this, CNN is a type of Artificial Neural Network that using supervised learning imitates the visual cortex of the human eye. In CNNs hidden layers are specialized in one specific task and are sorted hierarchically. The first layers can recognize simple shapes such as lines or curves. Deeper layers can detect much more complex shapes such as faces or landscapes.

CNN requires much lower pre-processing compared to other algorithms. This is what we call the “distinctive processing” of a CNN. That is, the so-called “convolutions”. These consist of multiply scalar, a group of nearby pixels from the input image by a kernel. The kernel goes through all the input neurons creating a new hidden layer. In Fig. 8 a schema of how an image is treated by a CNN is shown.

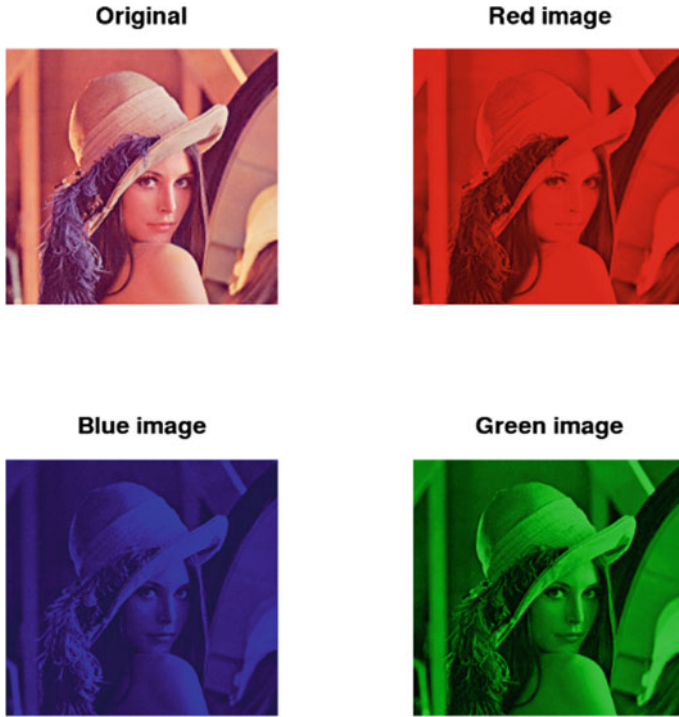


Fig. 7 Red, Green, and Blue example of how an image is shown in the RGB representation

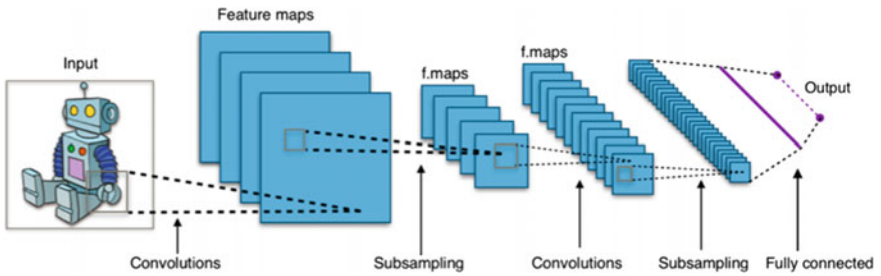
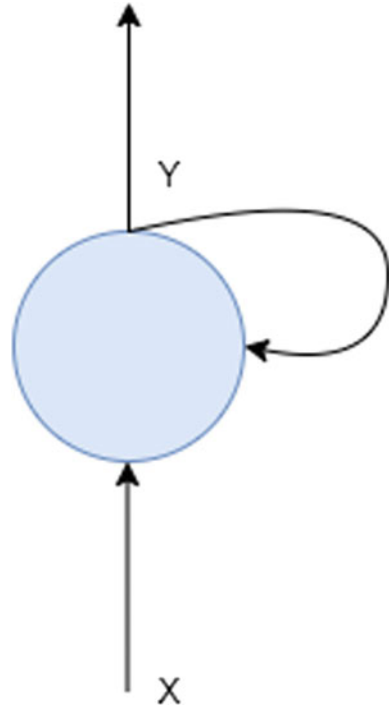


Fig. 8 CNN schema of how a convolutional neural network works. This file is licensed under the Creative Commons Attribution-Share Alike 4.0 International license

### 3.5.3 Recurrent Neural Networks

So far, we have seen networks whose activation function only acts in forward direction, from the input layer to the output layer, that is, they do not remember previous values. A Recurrent Neural Network (RNN) is similar, but includes connections that point backwards, a sort of feedback between neurons within the layers. These networks are called recurring because they perform the same task for each element

**Fig. 9** Single RNN neuron with feedback of the output



of a sequence, and the output depends on the calculation explained above. To see how this works let us imagine the simplest possible RNN. This is the case of a single neuron that receives an input, generates an output, and sends back that output to itself. This idea is shown in Fig. 9, in which a single neuron with a regular input and the previous output are shown as the inputs for the process.

At each instant (also called timestep in this context), this recurring neuron receives the  $x$  input from the previous layer, as well as its own output from the previous time instant to generate its output  $y$ . Following this same idea, a layer of recurring neurons can be implemented in such a way that, at each instant of time, each neuron receives two inputs, the corresponding input from the previous layer and in the output from the previous instant of the same layer.

Given that the output of a recurrent neuron at a certain moment is a function of the inputs of the last moments, we could say that these kinds of neurons have memory. The cell state is preserved over time in an internal memory called memory cell. This internal memory makes the RNNs a very interesting method to apply on ML problems involving sequential data. This internal memory allows RNNs to remember relevant past input information, which allows them to make better predictions and keep contextual information. These algorithms are used for temporal problems, such as natural language processing, speech recognition, and language translation. Examples of applications are voice search, Alexa and Google Translate, detect fraudulent credit-card transactions, etc.

One of the disadvantages of a simple RNN is that it has a short memory. However, it has shown remarkable success in natural language processing, especially with its Long Short-Term Memory (LSTM) variant that can look back longer than RNN and, therefore, solves the problem of the short memory.

### 3.5.4 Natural Language Processing

Natural Language Processing (NLP) is an area that provides machines with the ability to comprehend the human language. NLP makes it possible for computers to read a text, listen to a spoken voice, extract the meaning, determine which parts are important, and even measure the sentiment. Virtual assistants or chatbots are one of the best-known utilities of the NLP, but they are not the only ones. In addition, it is important to understand that NLP does not give a chatbot intelligence, it only gives it the ability to process and generate human language. To provide intelligence to a virtual assistant, the use of systems such as neural networks is necessary.

NLP has many parts and many steps to succeed and being able to understand the human language. Usually, NLP starts dividing the text into elements (phrases, words, etc.) and trying to understand the relationships between them. One of the most popular models to divide the text in words is called Bag of Words. This model is very used. It counts all words in a text and creates an occurrence matrix neglecting word order and grammar. In this process it is also useful what is called Tokenization. This consists of cutting a text into pieces called tokens and removing some characters without interest for the analysis, as for example, punctuation characters. This part is what we call Natural language understanding (CLN or NLU). It is the part of natural language processing that is responsible for interpreting a message and understanding its meaning and intention, just as a person would. In order to get the system working, you need datasets in the specific language, grammar rules, semantic theory and pragmatics, etc. In Fig. 10 we can see one of the simplest examples of tokenization.

Another popular part of NLP is Speech to text or STT. It is based on the conversion of audio to text, and it is a task to value the audios, which once converted into texts can be processed with other NLP techniques. Once processed it is possible to return an audio using the text to audio conversion (Text to Speech or TTS). Both tasks, STT and TTS, have become truly relevant with conversational systems with a prominent level of quality.

To sum up, the applications of NLP are huge, and the fields of application increase every day. Let us mention some examples:

```
Can you read this text?  
['Can', 'you', 'read', 'this', 'text', '?']
```

**Fig. 10** Tokenization of the sentence “Can you read this text?”



- Organizations can extract valuable information about customer choices and, mainly, their decision drivers. Also, they can determine the feeling about a product or service by analyzing the sentiment analysis, for example, social media.
- From the email text analysis using NLP, your email provider can classify your emails and stop spam before entering your inbox.
- NLP is also being used in talent recruitment. It permits to identify the applicant skills in the selection phase.
- Automatic Machine Translation is a field of research within computational linguistics that studies systems capable of translating messages between different languages or languages. For example, Google is one of the companies that has invested the most in machine translation systems, with its translator using its own statistical engine.
- Autocorrect and autocomplete text systems also use Natural Language Processing.

More examples of the different uses of NLP are given in the next section of ML applications.

### 3.5.5 Generative Adversarial Networks

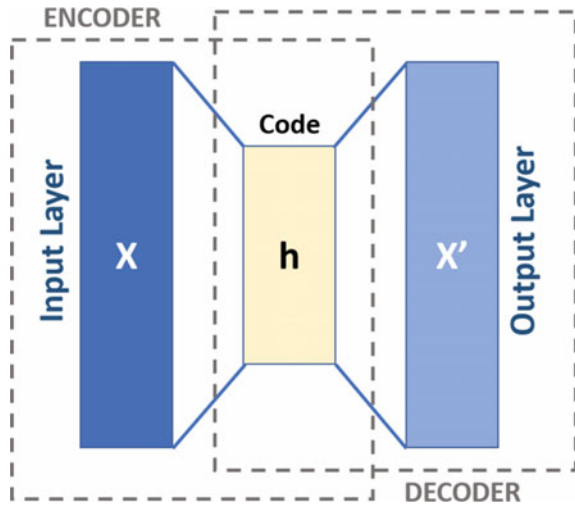
As its name may suggest, generative adversarial network (GAN) algorithm consists of two neural networks competing one against the other. One of the networks, called the generative, produces samples of what we try to create, and the other one, the discriminating network, examines the samples and determines if it fits the requirements. If the samples do not fit these requirements, they are discarded, and the generative is notified about that, forcing it to try again. This process can be repeated thousands or even millions of times until the discriminating network agrees with the result. In this process, the generating network learns what the discriminating network is looking for.

One of the latest and most surprising applications of this technology is the fake human face generators. An example can be the DCGAN from Nvidia. This tool allows to generate hyper-realistic faces that do not correspond to any real person. Another use of this technology is to generate samples of photorealistic images of industrial design, interiors, clothing and accessories, or elements for scenes from computer games.

### 3.5.6 Autoencoders

The most representative characteristic of autoencoders is that in this kind of networks, the input is the same as the output. Their objective is the generation of new data. Autoencoders compress, or encodes, the input into a lower dimensional latent representation code and then reconstruct the output from it (Dertat, 2017). This type of network consists of three parts, Encoder, Code, and Decoder. The encoder is the input part of the network, and it compresses the input into a space of latent variables.

**Fig. 11** Autoencoder representation. It has three parts, the encoder, the code, and the decoder. This file is licensed under the Creative Commons Attribution-Share Alike 4.0 International license



The product of this process is the Code. Decoder is the part that tries to reconstruct the input based on previously collected information, the Code. These three parts are connected as it is shown in Fig. 11.

If we observe the code, we will see that at this point the Autoencoder has a compact representation of the input. That is, the data obtained is a compressed version of the input and therefore contains a smaller amount of data. The representation obtained in the code is known as latent space and is the result of training, where the network learns how to extract the most relevant information from the input data. What we hope is that, when we train an autoencoder and copy the input, the code can take on useful characteristics for us. To find this compact representation, the Autoencoder is trained in an equivalent way to a Neural Network. However, in this case the error function used to update the autoencoder coefficients is simply the result of comparing point by point the reconstructed data with the original data. Taking this into account, we can say that the Autoencoder is an example of unsupervised learning, because during the training we do not really define the category to which each entry belongs.

Some of the most popular uses of Autoencoders are dimensionality reduction, removing noise in images or detecting anomalies in a series of data.

### 3.6 *K-Nearest Neighbors*

K-Nearest Neighbors (kNN) is the main example of a clustering algorithm. As we mentioned above, clustering is a type of unsupervised ML. This means that in the dataset we just have objects and features, and we do not know the classification of the objects.

These algorithms take a new element and calculate the distance from it to every other element. Depending on the distance to  $k$  neighbors the new element is classified into a group or another. The selected group will therefore be the one with the highest frequency with the shortest distances. To decide the group the point belongs to, there are two values that we must define beforehand. The value of  $k$ , which is the number of elements per class, and the formula to calculate the distance between the points. The most popular ways to “measure the closeness” between points are the Euclidean distance or the Cosine Similarity (it measures the angle of the vectors, the smaller, and the similar). The value of  $k$  is especially important, as this will almost end up defining how many groups are and to which group the points will belong. The best choice for  $k$  depends on the data. Normally higher values of  $k$  reduce the way noise effects’ classification. However higher values of  $k$  make the boundaries between classes less differentiable.

Despite its simplicity, it is used in solving a multitude of problems, such as recommender systems, semantic search, and anomaly detection. The main advantage of this algorithm is that it is simple to learn and implement. Its disadvantages are that it is very memory and processing resources (CPU) demanding, since it uses the entire dataset to train each new point. So, because of this, KNN is best suitable for small datasets and with a low number of features.

## **4 Machine Learning Applications in the Real World**

As we already said at the beginning of this book, ML is being used in many and many applications, so it is impossible to condense in a chapter a full comprehensive review of the ML applications. Instead, this chapter aims at showing the palette of current applications, ranging from engineering to fundamental science. We provide significant examples in each area, focusing on the methodology, and relating every topic to the techniques discussed above.

### ***4.1 How Machine Learning Improves Engineering***

The work of engineers has not escaped the current revolution, and the way their jobs are done has had to dramatically align with the most modern trends in ML. The introduction of ML and all its tools bring significant improvements to what we are capable of on our own (Smith, 2020). One of the most distinct aspects of ML incorporation in engineering has to do with the workflow management. ML goes well beyond the designing, into data management and interoperability. As one could expect, ML helps to manage engineering data in a more optimal way, effectively adapting in Industry 4.0, solving the problem of big data, and making other advancements easier to deal with (Hulten, 2019). Moreover, ML is also allowing a better communication between different departments to perform different tasks. If computers and systems

are capable of learning over time, then they can improve and automatically carry out many daily efforts. In summary, ML is perfect to improve our capabilities in engineering, given as we have seen, it is able to take advantage of computers by imitating our own learning.

We all know engineering problems which solve traditional computational techniques to the limit. Many times, they are solved because there are experts capable of reaching the most appropriate solution, and then checking it with conventional calculation methods. In this context, ML is trying to capture the essence of human cognition to accelerate the resolution of these complex problems. ML has been used intensively in engineering research in recent decades. Newer methods such as pattern recognition, ML, and deep learning are emerging methods in this field of engineering. These emerging techniques could learn complicated dependencies between parameters and variables and hence allow facing more efficiently a variety of obstacles that otherwise could not be dealt with by means of traditional methodology.

One of the main aspects of ML is improving our lives and engineering in general is when it is applied in Big Data (Hasan et al., 2014). For some years the amount of data existent on the Internet has grown exponentially. This means we can find huge amounts of data with different structures, which appear in business at almost any time. The key with this data is not so much the amount of it we have, but rather how companies use it. In this regard, it becomes essential to analyze it in detail to later take good business decisions that, e.g., improve their income. And it is in this part of the process in which ML starts to be useful for engineers. Some examples of uses are:

- **Tourism:** the tourism industry is essentially about keeping customers happy. However, quantifying this happiness is often hard. Furthermore, the detection of dissatisfaction must be done as quickly as possible, otherwise there might be no time of reverse a bad customer experience. Big data provides handles to companies to tackle these problems, by providing advanced analytics of customer data, helping to prevent potential problems well in advance. Recommendation systems built by NN are one of the main tools Tourism uses to promote and reach the targeted population, to increase the benefits.
- **Administration:** administration must tackle the challenge of maintaining excellence and productivity with a cost as low as possible. This is even more true for justice.
- **Autonomous car:** the era of the autonomous car will be a reality in the not-so-distant future. In fact, there are already concrete steps in this direction. Driving without having to do anything at all will become a very common habit in a few years. Facing that horizon, more and more brands announce new research. These are focused on security systems that include the most amazing technologies for when the autonomous car arrives. In this scenario, ML already shines and will be one of the great protagonists. Specially DNN and CNN are used to work with the information the diverse sensors of the car recollect (Amezquita-Sanchez et al., 2020). This information changes from temperature, speed to images of the road, making the use of complex algorithms a necessity.

However, ML not only improves data management and the way Big Data offers solutions. ML can also be applied to electrical and computer systems. ML makes it possible for logic control to create rules that enable machines to react against inputs of very different kind, rather than simple binaries (Aguilar Vidal et al., 2020). In general, it makes it possible for systems to discover different arrangements, find inferences, and understand how to perform tasks with no specific guidance. In addition, ML has a high potential in the field of computational mechanics, given how these methods can solve complex problems through the so-called Internet of Things. This introduces one of the main developments of the area, the time for smart infrastructure, cities, homes, or structures. Smart Cities are not a utopia. In fact, smart cities are a reality whose basis is the use of infrastructure, innovation, and technology to reduce energy consumption, reduce CO<sub>2</sub> emissions, and increase the quality of life of the population. The parameters that serve to qualify a city as Smart City are its commitment to the environment, its urban planning, its public management, its mobility and transport conditions, its efforts to facilitate social cohesion, and human and economic investments to improve its operation. Following the concept of Smart City, the idea of Smart Home is introduced as those that incorporate a system that allows automating many tasks, as well as having full and live control over what happens in our home (Riquebourg et al., 2007). The principal goal is to incorporate automation to manage air conditioning, lighting, security, audio or video. All these systems are interconnected and can be controlled remotely. Smart homes are also committed to efficient and intelligent consumption, always adjusted to the needs of users.

Finally, we must take the job evolution into consideration. We have a wealth of examples in the past of how innovation usually becomes a new sector of research and work, and this time it will not be different. In this regard, most of our innovations in engineering currently have ML as a key feature. Our workflows will evolve, new fields of study will appear, and new types of industry are to be developed so that existing engineers will need to adapt to. We can already see how data and ML engineers are the new fashion profession in the business and research sector.

However, we still encounter limitations in the use of these emerging methods. These limitations include how difficult it is to select the most optimal ML method, the fact that the repercussions of noisy or incomplete data as well as the efficiency of computation are not considered, or the non-reporting of the accuracy of the data. Other examples are the classification without examining different solutions to increase the yield, and the inadequacy of the presentation of the process to select the best parameters for the ML problem under scrutiny. Yet despite these limitations, ML, pattern recognition, and DL are posited as pioneering methods for increasing the efficiency of many applications in modern engineering, as well as for developing new possibilities.

The automatization achieved by ML models, such as NNs, usually provides systems that can make better and faster decisions than humans. Since these systems will continue changing, we expect them to be able to transform our competence to exploit information at different scales.

## 4.2 *The Advantages of Using Machine Learning in Finance*

As we mentioned before ML is expanding to more and more sectors of our day to day, and finance is not an exception. Currently, ML is seen as a key resource in finances, including aspects such as asset management, risk level assessment, credit score calculation, and even loan approval (Israel et al., 2020). The industry of financial services usually deals with huge data volumes, related to daily transactions, vendors and customers or payments. As we have seen, this is an ideal environment for ML. This results in more optimal processes, lower risks, and better designed portfolios.

Some of services being improved by ML are:

- Algorithmic trading

The use of different types of algorithms to choose the best trading decisions is usually known as Algorithmic trading. Mathematical models to scan trading news and activities are developed by traders, with the goal of identifying the actual facts that affect prices (Jansen, 2020). In opposition to humans, these algorithms are capable of assessing vast amounts of data at the same time, and therefore make thousands of operations every day. Moreover, very often humans make decisions biased by their own emotions or feelings, which make them judge incorrectly. Algorithmic trading does not have this limitation. Different financial institutions employ algorithmic trading in order to automate their trading activities. Some of the most used algorithms are SVM and DNN (Abedin, 2021).

- Automated trading

Robo Advisor and Quant Advisor are both online investment platforms that provide an automated portfolio management service. That is, they offer a computerized advisory service based on algorithms without any human intervention in the investment decisions. Mostly they are based on NN and RL. They do not depend on the bias or fears that people may have. The main difference between these two fintech products is as follows:

- The Robo Advisor performs passive management through ETFs and index funds that try to replicate an index. Robo-advisors are online programs built using ML, which provide automatic advice to investors. The applications use algorithms to enact a financial portfolio depending on the investor risk tolerance and actual goals (Rosov, 2017).

Robo-advisors tend to be cheaper than their human equivalent, requiring lower investments. Robo-advisors ask the investors to fill in their investment and goals, so they automatically find out which can be the highest returns or investment occasions.

- The Quant Advisor proposes an active management, looking for opportunities that the market generates at any moment of the economic cycle. They are based on quantitative analysis and are uncorrelated with the markets. Quant-advisors seek to provide tools that allow obtaining positive returns without the exposure

to the risk of volatility of the market being high. This way of investing avoids the high sensitivity of other methods to both bullish and bearish movements in the market. The instruments most used by quant-advisors tend to be listed futures, either on equity indices, on fixed income bonds, commodities, or currency.

- Fraud detection

Given the transactions from third parties, as well of the number of users doing these transactions, keeps on increasing, the protection against the security threats appearing in finance is becoming more and more and important. The losses from banking related frauds account for billions of dollars each year and therefore an efficient detection of these frauds is something the finance industry certainly looks forward to. One of the main sources of risk in this case arises from the fact that companies store most of their data online, which make security breaches easier to appear (Corporate finance institute, [n.d.](#)). Previous fraud detection systems were based on simple rules, which could be broken smoothly by many modern fraudsters. Therefore, ML is becoming the most common solution against this type of fraudulent financial transactions. The simplest solutions are created without the development of more capabilities, taking instead different data types to identify anomalies. More complex algorithms can provide data codes and/or scores that a real-time engine uses to decide. Mostly ML works by going through large data sets to detect deviations to be further investigated by a human expert. These algorithms are based on checking features of transactions, such as IP address, physical location, or the history of the account. This way the algorithm can find out if the transaction under scrutiny is compatible with the behavior of the holder of the account. Some ML algorithms even block the transaction entirely, if there is at least a 95% chance that it is fraud. ML algorithms need only seconds to evaluate a transaction. The identification of fraud is a binary classification which uses one of the most efficient binary classifiers, such as SVM, Decision Trees, or Logistic Regression. CNN and image identification are some of the other big improvements in fraud detection.

- Loan/Insurance underwriting

In the banking and insurance industry, ML is changing the way insurance companies and banks serve their users. Like many other industries, insurance and loans are data-driven. Companies access vast amounts of data from consumers, which can be later used to train ML algorithms that facilitate the underwriting process. ML algorithms are capable of deciding on underwriting and credit scoring, to help companies saving money and time. This means algorithms can be trained to assess consumer data, compare it to stored examples, look for similar cases, and finally decide if it is a good idea to accept a client for a loan or insurance. Examples of items these algorithms can use are the age, credit behavior, job or income.

Using ML, banks and insurance companies are realizing that the benefits are increasing, some of the most visible ones are:

- Increase in loyalty. Its use entails an increase in the speed of the response and the success in the resolution of incidents.

- Efficiency in claim procedures. They send people to the corresponding area or department to be attended quickly.
- Success in resolving incidents. ML makes an exhaustive analysis of the most successful claims and resolutions to offer the solution that best suits the client's needs. Correct management gives greater confidence in the insurance brand.
- Fraud prevention and detection. It is also used to calculate, based on a series of predictive algorithms, the probability of a customer trying to commit fraud in a claim, since it extracts data from internal and external sources.
- Offer customization. ML helps to predict better loss risk, which means cost savings and custom premiums.
- New ways of assigning policy prices. Obtaining quality information has become a much simpler and more accurate task. Sensors in vehicles, buildings and even the possession of smartwatches, allow insurers to be more exact in measuring and forecasting a risk. The information reveals data on the customer's lifestyles, driving habits, and so on. As a result, insurers can set prices without running risks and in turn allow consumers to purchase services according to their needs.
- Process automation

Process automation through ML appears in almost every field of knowledge nowadays. The reason for this is that the type of ML algorithms we have been discussing in this chapter allows the replacement of several manual tasks, avoiding repetition and helping to increase productivity. This topic will be further discussed in the following section *Machine Learning to automatize interface with final users*. The automation of processing in banking shows how ML can improve finances. Examples of these new applications are:

- Using NLP, a bank designed a contract intelligence program to deal with legal documents and automatically obtain data from them. If this were to be done manually, the reading of 12,000 business credit agreements would need around 360,000 h, compared to just a few hours using this type of ML tools.
- Another bank uses a chatbot controlled by ML with NLP through the Facebook Messenger tool to connect with users and aid with accounts and passwords.
- A Ukrainian bank made use of chatbot assistants on its online platforms. Chatbots accelerated the answer to customer inquiries of every type and decreased the need for human interaction.

Despite all the improvements ML can bring, even resource-rich companies often find it difficult to make the most out of this technology. It is not enough to have an adequate software infrastructure. Instead, it requires a clear idea, good technical skills, and courage to achieve worthwhile ML-related goals. Even so, many economists and financiers predict that within a few years a large part of financial processes will be developed through ML.



### 4.3 *Industry 4.0 and Machine Learning*

The goal of the so-called Industry 4.0 is to transform industry and make it intelligent. Thanks to the use of Cyber-Physical Systems, the Internet of Things (IoT), Cloud Computing, and Big Data; it is possible to collect, store, and compute a large amount of data (Datta & Davim, 2021). Due to the amount of data, manufacturing is one of the main industries that uses ML technologies to its fullest potential and, to achieve it, the industry needs to process all this data in an efficient way. In order to remove the manual data collection, the use of sensors was added. Some examples of these sensors are: humidity and temperature, cameras that allow recognition of shapes and objects, location and position sensors in space, pressure sensors, temperature sensors, etc. Another important part in this procedure is digital image processing. This technique allows the analysis and processing of digital images with computers to extract useful information from them. In manufacturing it can be applied to a lot of tasks. However, the application of ML algorithms is required to create models that give value to this data and facilitate decision-making.

There are many and diverse applications of ML in the industry such as:

- Production

In production, vision systems and robotics are combined with ML algorithms to improve processes and increase productivity. In fact, it is possible to automate tasks with such a variability that a traditional robot could not carry out on its own: recognizing and locating types of parts, processes and variable trajectories, etc (Monostori et al., 1996). For this reason, it allows in many cases to reduce costs and increase the competitiveness of companies. Moreover, improving sustainability is currently one of the challenges for manufacturers (López-Manuel, et al., 2020). In this regard, a full understanding of the processes, as well as finding simple ways to improve them, is a must. This is complicated in some cases, given how large the lines of equipment and processes can be, so it is crucial that manufacturers can figure out the best ways to implement these. For this, ML can provide interesting handles, such as NNs, which can be useful to obtain the most optimal methodology to carry out a process. The main advantage of this automatization is to find out which issues the production can face and how to tackle them. Some of the production issues that are specially solved by ML are:

- Failure rate reduction

They allow the detection of failures and their reduction, which has a direct impact on the quality of the production and its improvement. Mistakes from the past help to improve the process.

- Stock optimization

Stock optimization pursues one objective: maximize profit or minimize costs. With this clear, it is necessary to consider a series of restrictions that can influence costs (maximum stock, transport cost, delivery times, etc.). ML is usually applied to predict

the sales of a product and hence predict the stock and a possible stock shortage. Also, the stock and sales prediction help to set the prices and offers for the products.

- Process automation

With these algorithms it is possible to automate processes that would not be possible without learning-based systems: variable inspections, changing environments, etc.

- Quality

Quality improvement investigates the relationship between the features of a product result and how well it matches its design and desired performance. The quality index can be objective (a physical or chemical test) or subjective (a human test). For this type of problems, the sensor's information is used by a NN, that can be either a DNN or a CNN to process images, to determine if the quality check is passed or not.

ML can be used to predict the quality of a product in two separate ways:

- (a) Based on its design, modeling the result allows to simulate assorted products and adjust the design to improve quality.
- (b) An anomaly detector algorithm can automatically indicate issues in the production line that can result in quality flaws. For example, to detect defects in parts, manufacturing surface defects, paint, etc. They also allow quality checks to be carried out in an assembly process, the presence or absence of parts, the inspection of welds, etc. These mistakes can be of different range but, in any case, have an overall influence in the production so that removing them in the initial stages would certainly help to save resources.

- Logistics

The supply chain generates a huge amount of both structured and unstructured data every day that can only be exploited thanks to artificial intelligence. Logistics is based on physical and digital networks that cannot be optimized by humans due to their high complexity (Kotsiopoulos et al., 2021). Therefore, the goal of ML is to transform reactive behaviors into proactive ones, manual into automatic, and standardized into personalized ones. For autonomous transport to be accepted, it must exceed the capabilities of a human behind the wheel, starting with the perception of the environment and its ability to predict changes in it. This is possible thanks to the combination of technologies that build a three-dimensional map of the environment. DN is responsible for processing data to identify traffic signs, detect obstacles, and other cars on the road, as well as comply with traffic laws. These algorithms are essential for the machine to acquire knowledge, since humans are not capable of programming all the possible situations that may occur on the road.

- Maintenance

In recent years, the cheaper sensors and their smaller size have facilitated the obtaining of valuable information on the state of the machines. Specifically, by measuring different points and characteristics of a machine, it is possible to have an almost real-time view of its status. By analyzing data obtained from different

machines, models capable of predicting failures can be generated. Likewise, this helps to improve processes, avoid failures before machines break down, avoid production stoppages and reduce preventive maintenance time. Thanks to ML, models are created from this data in order to detect possible anomalies before they happen. This is called predictive maintenance, and its application is one of the most reliable ways to prevent machines from failing and damaging the production process (Chuprina, 2020). Although the concept of predictive maintenance is not new, the development of data collection, storage technologies and ML applications have contributed to create a new perspective on this term. In this way, the contribution of data from many sources is treated with more complex algorithms that have allowed a reduction in maintenance costs of between 10–15%. Although ML manages to analyze data and learn extremely effectively, it is important to highlight the role of humans when developing and improving tools related to maintenance. Both the experience of the workers and the historical data of the machines are basic information for the systems to work. In addition, feedback from an experienced person is needed to adjust the algorithms and validate the results they show. After all, decisions are made by humans based on predictions made by ML.

- Ergonomics

Regarding the working conditions in the production and assembly positions, Motion Analysis Systems (MAS) are used to create detailed reports on both the productive and ergonomic performance of the worker. This is achieved with a hardware called MOCAP (Motion Capture) integrated with software based on neural networks specialized in the treatment of images and videos. The adoption of MOCAP technologies in industry has grown in importance with the advent of smart factories. MOCAPs were originally designed to recognize movement in video games, but now the benefits of using the same principle to study and improve manual activities have been seen. Another way of applying Machine Learning in ergonomics is achieved with the sensors equipped in smartphones. These sensors provide information on the location and movements carried out by the user, such as the step counter. Unlike other devices, smartphones are cheap, easy to use and do not require a lot of maintenance.

- Security

The increased use of the Internet, both in social and work life, completely changes the way people learn and work. However, the number of cyberattacks is increasing in the same way. In the period of digital transformation in which companies find themselves, a technological incident of this type can put an end to the continuity of the company. For this reason, cybersecurity arises as a set of technologies and processes designed to protect networks, computers, programs, and data from possible attacks and threats. Powerful Machine and Deep Learning algorithms in cybersecurity are mainly used for malware analysis and intrusion detection and prevention. The development of these algorithms is driven by the need to anticipate a cyberattack and restrict access to infected files or programs.

- Product classification

Finally, we have a point that we can relate to some of the previous ones, such as quality control and artificial vision (Wuest et al., 2016). And it is that, both the sensors and the cameras help to identify aspects that will be decisive for the classification of the products based on the measured parameters.

The biggest companies in the world have been utilizing ML in manufacturing and investing millions in related developments. Some of them even have been digitizing the factories and buildings for many years. Although this concept sounds very avant-garde, the truth is that it has been in place for a long time. In fact, it is not the future of the sector, but the present. What ML proposes is to take the company to a much more advanced level of digitization.

#### ***4.4 Machine Learning to automatize interface with final users***

Whether we want it or not, ML is behind the curtains every time we do a browser search, an online purchase or even take a picture with our phone. In the last decade ML has improved a lot our relations with machines and allow them to achieve more often what we want from them (Dudley and Kristensson, 2018). Of course, many of the advances in this area are very well protected and licensed by software companies and we do not intend here to disclose any of those. Instead, we will briefly go through some examples to show how ML affects the user interface, and how it relates to the different methodologies we are discussing in this chapter.

- Searching

Our favorite search engines have ML algorithms behind their success. We use the most popular engine, Google, to briefly explain the interaction between ML and internet searches. Google searches do not work with strings per se, but “entities”. Entities are unique items based on semantic analysis and they are formed when the strings in a search are grouped according to aspects such a context of the search, history, or ranking. Entities can be linked through relations that exist before any actual search is done. In the actual search, entities can be connected through contents. For instance, in the question “is Brazil beautiful?”, “Brazil” and “beautiful” are the entities and “is” the content. “Brazil” and “beautiful” were already connected entities through links given that is a statement that appears often!

Apart from entities, the other element in Google searches where ML plays an essential role is RankBrain. RankBrain is a system that connects different searches, where similar entities are involved, also accounting for additional information such as who is making the search, from which device or its location. Furthermore, RankBrain is continuously learning these connections through an iterative process, where it keeps improving to become more accurate. With this, it can provide a user optimized ranking whenever a search is performed. RankBrain runs

in Tensor Processing Units (TPUs) (<https://cloud.google.com/tpu/docs/tpus>) which are application-specific integrated circuits developed specifically by Google for ML applications.

- E-mail filtering

E-mail filtering is usually given as one of the typical examples of ML application, especially around natural language processing. However, this is still an area where developments keep being made, given phishing, spam, or in general unwanted messages threaten our mailboxes increasingly often (Karim et al., 2019). Apart from annoying for the average user, spam mail causes millionaire economic losses across the world. An e-mail is composed of a TCP/IP header, a Simple Mail Transfer Protocol (SMTP) envelope, an SMTP header, and the body. All these elements can give us hints of whether an email is spam and are therefore tackled by the spam filter algorithms. The first obvious solution against spam does not involve ML per se but focuses on ensuring the sender/receiver of the mails are authenticated and authorized. For instance, the use of encrypted mails is becoming the norm. Other solutions involve *hashing* of e-mails or filtering based on regular expressions.

Focusing on ML, although both supervised and unsupervised solutions are possible, the latter remain the most accurate and frequent, with algorithms focusing on SMTP headers and body of emails. The most popular solutions are currently SVMs and Naïve Bayes, although Neural Networks are becoming more and more employed. For instance, CNNs can be used to detect spam based on images associated with the emails. As explained, this is an area that keeps improving and with a lot of work ahead: as spammers become increasingly sophisticated, so must be the methods to detect their mails.

- CAPTCHAs and cybersecurity

CAPTCHA stands for “Completely Automated Public Turing Test to Tell Computers and Humans Apart” and represents a particularly smart use of ML for user interface. CAPTCHAs were designed for cybersecurity purposes, as a barrier against hackers, to ensure online interactions are with humans and not with machines (Yang, 2018). We have all experienced CAPTCHAs when browsing online, but what many might not know is that by entering CAPTCHAs we are training a ML algorithm. The first CAPTCHAs were used to directly ML algorithms to “read” texts more accurately. More recently, they are becoming a way to train CNNs for image recognition. The fact that CAPTCHAs are becoming harder and harder for humans to solve is a sign of how hard it is becoming to find tasks that machines are not able to perform.

Other than helping with the defense, ML can also be used to perform cyberattacks. For instance, GANs have been shown to improve brute force for cracking passwords, based on the real passwords obtained from actual password leaks (Hitaj et al., 2019). Other possibilities include the use of RL to design evasive malware or RNNs for preparing advanced phishing attacks based on information retrieved online.

- Translation

An automatic and accurate translation between languages is one of the most ambitious targets of ML algorithms. This problem is usually targeted through RNNs, since these somehow emulate the reading process, keeping a memory of the processing steps, so that outputs from one step can serve as input for the same step at a later stage. An alternative is the use of CNNs, as in some commercial solutions, such as DeepL Translation.

Other recent developments are not directly involving CNNs or RNNs, but instead rely on novel NN architectures. For instance, a new network based on “attention mechanisms” (a type of encoder-decoder architecture) has been able to improve the scores achieved by human professionals in Czech–English translation of texts (Popel, 2020).

Two more areas of work are related to translation, although they are not as much progressed since they involve additional degrees of complication. The first is speech recognition and translation, which would help to achieve online translation of conversations. The second is augmented translation of texts, such as translating a written panel live just by pointing our phone cameras to it. Both are combinations of different ML techniques. The first layer involves the speech or image recognition, explained later in this subchapter, and the second the translation itself, as discussed here. In particular, the online translation of conversations is especially challenging and rewarding at the same time, so a lot of developments are expected in this area in the future.

- Advertising

We have all experienced it: doing an internet search for a product, and then spending several days watching ads for related products even when you are not interested anymore. Private companies keep learning increasingly about us, and this information can be more effective in tempting us to buy what they have to offer. ML is excellent at deciding which is the best way to trick us, building publicity based on our browsing history, interests, or even traces of our personality companies has been able to learn (Hwang, 2019). Moreover, the ability of ML to find hidden patterns is especially useful in this context, connecting products with customers that would have not been otherwise targeted.

This can be taken one step further, by finding out how certain images can be more appealing on marketing depending on the viewer characteristics or even designing the right ad for the right person. For instance, a study has designed a model that is trained to tailor personalized ads based on simple LASSO regression.

Another possibility is the use of so-called “contextual relevance”, based on natural language processing and DL. In this case, ads are designed to be displayed only in certain contexts. For instance, this can be used as an aggressive strategy to have your ads placed next to those of your competence. A final example involves ultra-personalized publicity. This involves creating different versions of ads to be shown based on who is watching them and in which context (i.e., next to a video, in a

newspaper website). This approach targets not only our personalities but also our moods when we are reading or watching specific types of content.

- Customer support

The use of ML in customer support revolves around speech recognition and natural language processing. Companies are starting to use ML to provide a quick and direct way to most of the issues customers raise, as well as to save costs by reducing the need for human intervention. The huge amount of client interactions that big companies must deal with, while a burden, is also interesting from a purely ML point of view, since it generates data for ML algorithms to be trained.

The first line of attack for ML in customer support is the automatic identification of customer issues. The same issue can be phrased in an almost infinite number of ways, and being able to recognize this similarity is an important challenge for natural language processing algorithms. An additional input to recognize issues might come from “social listening”, which relates to marketing and refers to the tracking of online activities that users have performed prior to their queries. A variation of this task is the assignment of agents to customers. In that case, the idea is not so much providing a solution to the issue but identifying the topic of the inquiry to find the right human expert. This is becoming more frequent in call centers, which, as with translation, adds an additional layer of complication through speech recognition. In this case, not only *what* is being said is of interest, but also *how*. For instance, measuring the volume or pitch of the voice can be used to determine the priority of the call: companies want to listen first to angry clients!

A more ambitious approach is that of the famous chatbots. Those intend to provide a full service to customers (intend-based chats), even attempting to establish conversations (flow-based chats). Chatbots turn out to be one of the most demanding applications of natural language processing but is an area where a lot of research is being done and significant improvements are expected in the near future.

- Speech recognition

We have already seen in this chapter how speech recognition has multiple applications in the user interface. From translation to customer support, speech recognition is a pillar in which many ML applications stand. Other important examples involve virtual assistants (such as Alexa or Siri), enhanced biometry using voice recognition, transcription of meetings or interviews, language learning, or automatic subtitling of contents (Kamath et al., 2019).

The first step for speech recognition is sampling, i.e., taking pieces of the audio input and digitizing them. Sampling usually also involves a mathematical operation called “Fourier transform” which helps to disentangle the frequencies involved. Note sampling is a classical problem, but it has lately become a focus of ML algorithms themselves, such as CNNs.

The next step is the conversion of the sampled input in the actual text outcome. For this, RNNs are usually employed since, as discussed above, they are good at tasks related to “reading”. In particular, the so-called long short-term memories RNNs (LSTMs), specialized in time series data, are frequently part of this processing step.

Large datasets of written texts are an important additional source of information here, which can help to better interpret what comes out of the speech. For instance, if the network doubts between “apple” and “attle”, this will give a much higher score to the first one, given it is a word that appears more often in English.

As in other cases in ML, speech recognition models are constantly “learning” from us. Note every gadget now has virtual assistants whose interaction with us is constantly used to improve their response through training!

The final challenge is finding end-to-end models that include all these intermediate steps in a single trainable algorithm. An exceedingly popular solution, applied both in natural language processing alone, but also in speech recognition, is Connectionist Temporal Classification, which provides a sequence of continuous output probability labels.

- Image recognition and manipulation

Image recognition is also present in many aspects of our daily lives, such as when social networks recognize our faces in pictures. Increased instances of use appear every day to make our interaction with machines easier. Recent examples involve the recognition of artworks or the visual search for products for purchase.

The ML analysis of images can involve classification, detection, and recognition (Voulodimos et al. 2018). These are obviously correlated, but not the same. The first, classification, is about categorizing images, such as grouping all pictures where a firefighter is located, among a group of pictures with different professionals on them. The second, detection, is about locating specific elements in an image, for instance, finding out if there is any dog in a picture with several components on it. The third, recognition, is somehow a combination of the other two, and becomes closer to what a human can do. For instance, it would imply finding out if a particular person, with a known face, is present in a photograph.

All three of classification, detection, and recognition heavily rely on CNNs. As with speech recognition, the preprocessing of the images is crucial to begin with, with decoding and resizing as frequent additional steps. Then, the processed image is fed into a DNN for the required task. This will usually use image patterns to achieve its goals. Several pre-trained models, such as AlexNet, ResNet, SqueezeNet, or DenseNet, exist in the market, making the analyses easier. Models for specific tasks can make use of transfer learning to make the training quicker and more reliable.

Another recent and related late development is that of manipulation of images or video through DNNs. When this manipulation involves humans, this results in the popular “Deep fakes”. These have had applications in films or social media, but also fraudulent ones, such as blackmail or fake news. From a formal point of view, the manipulation of images involves the use of autoencoders or GANs. The first encoded images into a latent space, which can be then decoded through a pre-trained model to imitate, e.g., someone else’s characteristics. As for GANs, a first network generates fake images that a second network tries to discern from real ones, creating a zero-sum game that provides increasingly perfect results.



- Transportation: dynamic pricing and routing

Many of the advances regarding transportation happening in the last years are also related to the development of new ML models (Fitzek et al., 2020). This goes from the fluctuations in the price of plane tickets to the choice of the best route to go from one place to another in our car.

As for the price fluctuations, they arise from a well-known law of the market that the price of items sold will increase if there is more demand for them. This principle can be extended to other aspects important for flag carriers, such as competitor prices, price of fuel, market movements, or even the habits of customers.

Dynamic pricing consists of building models that consider all these aspects to decide the best fares at any moment. The same principles used in plane tickets also apply to ride-hailing, hotel bookings, or in general e-commerce. Even if it is possible to establish rule-based systems for dynamic pricing, the best solutions come, as one could expect, from ML models. As we are often seeing in this chapter, ML algorithms can find hidden patterns and determine the best pricing strategy for a company. LSTMs are the most frequent type of models for dynamic pricing.

The custom models designed by companies provide a great degree of flexibility, since pricing can be modified according to expectations from customers and the allowed margin of profit that the company might have at a specific moment. This can be all done real-time, with no need for manual adjustments!

Then, concerning routing, the existing algorithms, such as Google Maps, are able to provide increasingly accurate answers about aspects like best route or time of arrival. For this, Google Maps feed several types of NNs with information regarding the type of road (highway or not, straight or curved, quality) as well as, of course, the traffic flows. For this, Google uses the so-called “Supersegments”, which are ensembles of roads that share their traffic volume. Then, to estimate the time needed to transverse each Supersegment uses a type of NN called “Graph Neural Network” (<https://deepmind.com/blog/article/traffic-prediction-with-advanced-graph-neural-networks>). These extend CNNs and RNNs by including the concept of proximity, loading the network of roads into a graph with connections and edges (Fig. 12). This type of NN has been able to improve the accuracy in the time of arrival provided by Google maps up to 50% in some places!



**Fig. 12** (Left) Example of CAPTCHA. CAPTCHAs are used in cybersecurity as way to distinguish humans from machines, but they are also useful to train ML models. (Right) Example of a “Deep fake” image, generated using adversarial learning. Free images to use a prior [[https://en.wikipedia.org/wiki/File:Unexpected\\_CAPTCHA\\_encountered.png](https://en.wikipedia.org/wiki/File:Unexpected_CAPTCHA_encountered.png)] and [<https://commons.wikimedia.org/wiki/File:Sw-face-swap.png>]

#### 4.5 How Machine Learning is revolutionizing Medicine

The use of ML in medicine is becoming increasingly popular, and the possibility to save lives makes any progress in this area particularly attractive (Rajkomar et al., 2019; Goecks et al., 2020). The scalability of ML algorithms makes those especially useful for medical applications, and an innovative way to make the most of some of the vast existing datasets. The fact that in ML a model, rather than having to follow strict rules, learns from examples can make these models very flexible, going beyond many of the existing traditional solutions. In this regard, ML algorithms are normally able to find hidden patterns in data, which might be challenging for health professionals. Finally, the fact that ML algorithms can be run instantaneously, once they have been trained, are a key aspect for certain applications, when a fast decision might be the key to avoid permanent harm.

Now we discuss the current trends of ML in medicine, providing specific examples in diagnostic, treatment, and health management.

- Diagnostic

ML has been shown to perform as well as medical professionals across different specialties in terms of diagnostic, and a joint human–computer diagnostic is currently regarded as the optimal solution for the future. Computers would offer potential diagnostic options, and it would be left to the health care professional to make the final decisions on how to proceed. Examples of use of ML in diagnostics are:

- Imaging: the use of CNNs is becoming very common in computer vision applications, e.g., cancer diagnostic, Optical Coherence Tomography (OCT) in eye conditions such as diabetic retinopathy (DR), radiological images or magnetic resonances (Esteva et al., 2019). As explained above, the use of ML can be sometimes especially useful when an urgent diagnostic is required, such as radiological images of the brain, when patients have limited time before permanent damage occurs. A remarkable alternative to CNNs around medical imaging is Deep Belief

Networks, made by accumulating autoencoders, which have been applied, e.g., to magnetic resonance images to detect signs of schizophrenia.

- Molecular tests: DL can be used for the interpretation of genetic data, for instance, for phenotype prediction, or in genome-wide association studies, making use of the ability of this type of algorithms to make complex associations that are hard to be found for humans.
- Treatment and prognosis

ML algorithms are excellent at accounting for multifactorial effects, overcoming the limitations of practitioners. This is often crucial to decide the best treatments to follow, or in general, to generate suggested therapies for experts to choose. Moreover, the capacity of ML to learn patterns from existing databases can be also useful to provide predictions about the evolution of patients or, more specifically, about the outcome of a disease.

A new remarkable use of ML in medicine has led to the discovery of new antibiotics (Stokes et al., 2020). For this a DNN was trained to predict how well different molecules inhibited the growth of *Escherichia coli* using a collection. Then, this network was applied to different chemical libraries, comprising hundreds of millions of molecules, to identify those with potential as antibiotics. The best candidates were then tested in mice, confirming this potential. Another powerful example is the use of RL for robotic-assisted surgery, where the RL algorithms learn from the motion of human surgeons.

- Health management

The most promising area for using ML in health management revolves around electronic health records. The reason for this is that ML is often useful to deal with massive amounts of data that are exceedingly difficult to interpret coherently as a whole (Ngiam and Khor, 2019). Apart from this, natural language processing tools, such as RNNs, are starting to be used to help in the analysis of the texts in, e.g., medical reports. The same goes with speech recognition techniques to transcribe conversations with patients. Another interesting example are unsupervised learning algorithms, which might be particularly useful here. For instance, tools such as autoencoders can learn representations to then reconstruct unlabeled data and perform diagnoses.

Other than this, wearable devices can be used to generate data for ML applications. This data can be used in health management to provide real-time health-care suggestions. The use of smartphones as an additional source of health information can be immensely powerful in this regard.

The use of ML tools in medicine is a rapidly moving field. Notably, in 2018 the American Food and Drug Administration (FDA) approved for the first time a ML-based DR diagnostic system (Abràmoff et al., 2018). This is significant since DR is one of the leading causes of blindness in the world. This automatic system, using CNNs to interpret OCT images, achieved a sensitivity of 87.2% and a specificity of



**Fig. 13** (Left) Magnetic resonance of a human brain. CNNs are capable of finding patterns that might be invisible even to the expert eye, and can do so much faster. (Right) Electronic Health record. EHRs are becoming ubiquitous in medicine, and their use is the great interest in ML to facilitate the access to medical data. Free image to use a priori [<https://pixabay.com/service/license/>]

90.7% when compared to the standard method used so far, using human readings of the OCT images.

This progress in the use of ML in medicine has inherent challenges, some specific to this area, related to the management of data and ethics. For the first, the main issue is the collection of formatted, unbiased and uniformized data. A workaround for this comes from ML itself, using data curation, i.e., algorithms that can pre-classify data well enough to make ML algorithms more performant. Moreover, concerning the bias, this might arise from the use of data produced in merchandised health systems. In those, for instance, sometimes unnecessary care is given, while, on the other hand, data from certain population segments, such as those with no insurance, is not available. Finally, for ethics, some of the issues to be addressed are the privacy, security, and control of patients' data, given biomedical information might be especially sensitive, so hard to share. A novel solution to this is federated learning, where just the algorithms have full access to data that is split in several independent nodes.

ML should in general not be seen as a replacement to human medicine practitioners, but rather to assist them. Instead of having them as black boxes, it is important that practitioners and patients understand how ML algorithms do their predictions. This might be crucial, for instance, to establish the liability in cases of medical errors (Fig. 13).

#### ***4.6 The Rise of Machine Learning to Accelerate Science***

Although ML algorithms seem to naturally fit in the category of “applications”, their use in natural science has exploded in the last few years. The capacity of ML for interpreting data and finding hidden patterns can provide a lot of advantages to understand the Nature around us. One of the most interesting aspects when using ML in natural science is the availability, in some cases, of exceptionally large datasets.

Spectacular examples are those of colliders in particle physics, producing millions of events per second, or the catalogues of molecules in chemistry, with hundreds of millions of samples. We next show some examples of the use of ML in science, focusing on biology, chemistry, physics, and geoscience.

- **Biology**

The increase in the availability of large datasets from biological systems in the last few years has accordingly led to incredibly significant expansion in the use of ML in this field (Tang et al., 2018). These datasets comprise aspects such as molecular variables, genetic variation, or microbiome composition. As in other areas, ML is especially good at deriving nonlinear relations in biology datasets that would otherwise remain hidden. The main applications of ML in biology appear in genetics and biochemistry, such as for genome annotation or the study of metabolic functions. However, other priori simple aspects, such as the study of cells, have also benefited from ML. For instance, the free tool Cell Profiler (<https://cellprofiler.org/>), which was developed to measure cell properties, has recently incorporated DL to record and combine features and produce new features for cell analysis.

As mentioned, ML is starting to be one of the main tools in genetics. For instance, since RNNs are particularly good at dealing with sequential data, they are becoming key in the understanding of DNA arrays or genomics sequences. RNNs, in combination with CNNs, have been used in genome-wide analysis or to predict gene expression. Likewise, other DNNs have been employed to build predictive models of RNA to identify potentially pathogenic mutations. Another interesting example is the prediction of the structure of expressed bioproteins through autoencoders.

Biological networks are also a good target for ML algorithms, given they involve large multi-dimensional datasets (Camacho et al., 2018). The study of these networks relates to the interactions between cellular systems through biomolecules, which are responsible for the structure and behavior of living cells. For example, in plant studies, SVMs have been used to study pre-microRNAs and mature microRNAs, which help to provide a better defense against pathogens. In the same regard, SVMs have also been applied for the development of disease-resistant plant varieties.

As for disease biology, DNNs are being employed to learn the key features that allow the recognition of healthy states, identifying the interactions and biomolecules that define these. A similar principle can be applied to the discovery of new drugs, through the prediction of drug toxicity for tackling cancer cells. These developments make use of the advantage of ML algorithms when dealing with multi-label datasets. As a final significant example in this area, transfer learning from other well-studied microorganisms has been used to help understanding the human microbiome, composed of the microorganisms inside the human body.

ML has exciting potential for other future applications in biology. A remarkable example is the use of DNNs for creating synthetic gene networks from biomolecules so that these would serve as “circuits” to modify the cells behavior, acquiring new capabilities of interest. For this progress to happen, the main challenges are achieving

large enough biological datasets for training new ML algorithms and the interpretation of the consequent ML models to help understanding the underlying biological mechanisms that are at play.

- Chemistry

Chemistry is no exception in the surge of ML in fundamental science. From chemical sensing or the design of experiments to the discovery of new materials or molecules, the use of ML algorithms is becoming increasingly frequent by chemists (Rodrigues Jr et al., 2019; Cova & Canelas Pais, 2019). The recurrent presence of patterns in Chemistry (in, e.g., combinations of functional groups or crystalline structures) is of special interest here, since this can be exploited by ML algorithms to find hidden properties. The source of the chemistry datasets needed by ML algorithms is varied: this might come from computer simulations, diverse types of sensors, specific experiments, or public datasets (such as in crystallography or materials data).

The discovery of new materials can be hard and expensive through traditional methods and is therefore a good target for the use of ML algorithms. In this case, it is essential to have at hand software to make proper simulations of materials and databases with their properties, together with an adequate representation of the relevant information. An example of this is the SMILES representation, which encodes molecular structures, and which has turned out to be especially useful when used in association with different ML tools.

For instance, autoencoders can be used to convert SMILES (Weininger & Smiles, 1988) representations in a “continuous” space, allowing to obtain the most optimal chemical properties. Similarly, RNNs can be also used in association with SMILES representations to generate new molecules with better properties. An original development in this area, which is very correlated to ML, is the so-called “genetic algorithms”. In this case, compositional and structural features of materials are interpreted as “genes”. The domains of these genes are then scanned to search for the adequate “phenotypes”, with the goal of achieving the target properties.

Another interesting approach in this area is the use of NNs or DNNs in synthesis prediction, with the network being trained to identify the sequence that is most likely to produce a specific compound and the optimal conditions for this to happen. In the same regard, RL algorithms have also been used for synthesis planning. Finally, crucial progress in this area is expected to come from computer-aided drug design. This might especially be important for the discovery of drugs tackling neglected diseases. Here, there is an obvious overlap with biology and medicine, described above. Advanced methodologies, such as graph-based structural signatures, DNNs or self-organizing maps (a type of NN) have already started to be used for drug design. These are especially useful, e.g., to predict the activity of bioactive molecules or minimize the toxicity of drugs.

Not surprisingly, ML has also appeared as a valid solution in quantum chemistry. These applications are correlated with some appearing physics, which we will be reviewing next. ML can supplement or directly replace complex calculations in quantum chemistry and predict quantities such as spin angular momenta or bond energies, as well as more complex aspects such as modeling electronic quantum

transport. Moreover, the combination of density-functional theory (DFT), which is a quantum mechanical method widely used in calculations to model the structure of matter, with ML has turned out to be very efficient. For instance, DNNs or even more simple methods, such as random forests, manage to significantly reduce the computation time when dealing with DFTs. However, one of the main challenges that ML faces in this area, when compared to, e.g., computer aided drug design, is the lack of abundant training datasets.

Developing new materials for chemical sensing is becoming easier thanks to ML. The easiest example is the use of different classifiers, such as Decision Trees or kNNs, to achieve simple quality control for distinct types of compounds in medicine, food industry, or cosmetics. In the same regard, in agriculture ML can be used for disease control, to differentiate samples that have been affected by disorders. Finally, more advanced techniques with immense potential could involve microfluidic sensors to detect gluten in food or even its taste. For this, techniques such as, Random Subspace Methods, based on ensemble learning are especially promising. As for biology, the future of chemistry seems to be intricately linked to that of ML. Progresses with chemistry or bio sensors will be essential to provide the large datasets that ML algorithms require. Examples of areas where developments are expected and which were not covered here are analytical chemistry or catalysis.

- Physics

From a more fundamental perspective, tackling the statistical side of ML, to a more applied one, looking into how ML algorithms could be run in future quantum computers, ML is becoming omnipresent in physics (Carleo et al., 2019). The initial relation between ML and physics comes from statistical mechanics, given some of ML algorithms, such as Boltzmann machines, were an application of physics concepts to data science. Currently, the understanding of different ML algorithms, such PCAs or even NNs from a physics perspective, is an active field. This is important to fight against the usual view of methods in ML as black boxes and help interpret the data connections one can learn when running these algorithms. Conversely, ML also provides interesting handles to study physics systems, such as the use of RNNs in non-linear dynamical systems.

The application of ML has experienced a boom in particle physics, astrophysics, and cosmology in the last decade. The first example involves the use of BDTs for classification and regression in particle physics. This has lately evolved into other methods, such as CNNs in classification of particle showers (jets) increased trigger selections rely now on ML. For instance, the LHCb experiment at CERN has developed several of these selections, such as one that relies on the use of NNs specialized in quick evaluation in real-time environments (Benson et al., 2019). As for cosmology, methods based on NNs and BDTs have been developed to measure the photometric redshift and CNNs to estimate several cosmological parameters based on different dark matter measurements. As a final example, different ML methods, such as GANs, are also being currently used to denoise and deconvolute reconstructed parameters in particle physics and cosmology.

ML is also suitable to deal with the interaction of multi-body quantum states. A recent remarkable development in that regard is that of Neural-network quantum states (NQS), which are a NN representation of the many-body quantum wavefunction (Carleo & Troyer, 2017). NQS are useful both for supervised and unsupervised tasks in the analysis of quantum systems. Another important associated aspect is the interrelation between ML and quantum computing. Apart from how quantum computers could enhance the training and application of ML algorithms, ML itself is also promising for the development of quantum computers. For instance, the tomography of quantum states, crucial in quantum information, can be done through deep learning approaches. Similarly, the preparation of qubits of control could be also improved via RL.

Apart from this all, there are several research areas in physics where ML is starting to be widely used, such as optics, climate science, or even in searches for exoplanets. Another remarkable area of research in physics is the use of new hardware platforms for ML application. An example is that of optical processing units.

- Geoscience

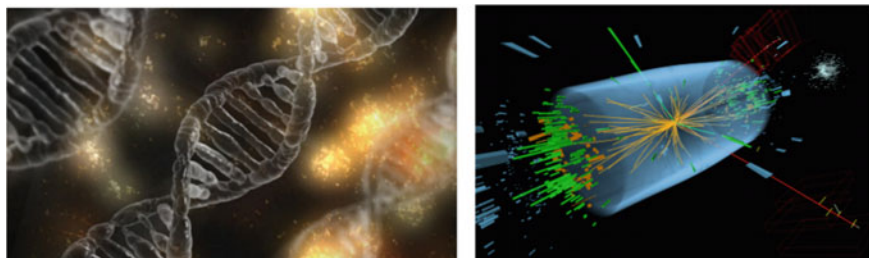
Geoscience is also starting to widely adopt ML techniques. Fields such as seismology or geomorphology are already using different variants of ML algorithms, and this is expected to increase in the future (Dramsche, 2020). The first uses of ML in geoscience involved, e.g., the use of SVMs for automatic seismic interpretation or of random forests for seismological applications such as event classification or localization in volcanic tremors.

As in other cases, DL is beginning to dominate the field of geoscience. Apart from their application in areas where other methods had been used before, such as seismic interpretation, this had led to a significant increase of applications. A powerful example is that of semantic segmentation for fault interpretation and salt detection.

More lately, U-nets, a type of CNN, have been used to interpret satellite data, to then prevent landslides or predict the arrival of earthquakes. Similarly, LSTMs, introduced above, are being applied to monitor volcanic activity.

Finally, although still at early stages, GANs are being applied with seismic data, for instance to help generate seismograms (Fig. 14).





**Fig. 14** (Left) DNA helix. RNNs are good at dealing with DNA sequences, helping in genome-wide analyses. (Right) Particle collision at the Large Hadron Collider at CERN. BDT and DNNs are used to classify and help reconstruct the particles produced after the collision. Left, free image to use a priori [<https://pixabay.com/service/license/>]. Right image, image courtesy of CERN [CC-BY-SA-4.0 license <http://cds.cern.ch/record/1606502>]

## References

- Abedin, M., Hassan, M. K., Hájek, P., & Uddin, M.M. (2021). *The essentials of machine learning in finance and accounting*. Routledge.
- Abrahamoff, M. D. et al. (2018). Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. *NPJ Digital Medicine*, 1, 39.
- Aguilar, V., Wu, H., & Montgomery, J. (2020). Machine Learning in Foundation Design and More. *TRB Centennial Circular*, 103.
- Alake, R. (2020). Deep Learning: GoogLeNet Explained. Available at: <https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765>
- Amezquita-Sanchez, J., Valtierra-Rodriguez, M., & Adeli, Hojjat. (2020). Machine learning in structural engineering. *Scientia Iranica*, 27, 2645–2656.
- Benson, S., et al. (2019). Real-time discrimination of photon pairs using ML at the LHC. *SciPost Phys.*, 7, 062.
- Brink, H., Richards, J., & Fetherolf, M. (2016). *Real-world machine learning*. Simon and Schuster.
- Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C., & Collins, J. J. (2018). Next-generation machine learning for biological networks. *Cell*, 173(7), 1581–1592.
- Carleo, G., & Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, 355, 602.
- Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., ... & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4), 045002.
- CellProfiler (n.d.). Cell image analysis software. Retrieved in October 2021 from <https://cellprofiler.org/>
- Chuprina, R. (2020). AI and Machine Learning in Manufacturing: The Complete Guide. Available at: <https://spd.group/machine-learning/ai-and-ml-in-manufacturing-industry/>
- Cova, T. F., & Pais, A. A. (2019). Deep learning for deep chemistry: optimizing the prediction of chemical patterns. *Frontiers in chemistry*, 7, 809.
- Corporate finance institute (n.d.). What is Machine Learning (in Finance)? Retrieved in October 2021 from <https://corporatefinanceinstitute.com/resources/knowledge/other/machine-learning-in-finance/>
- Datta, S., & Davim, J. (2021). *Machine learning in industry*. Springer.
- Dertat, A. (2017). Available at: Applied Deep Learning - Part 3: Autoencoders. <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>
- Fitzek, F., Granelli, F., & Seeling, P. (Eds.). (2020). *Computing in Communication Networks: From Theory to Practice*. Academic Press.

- Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.
- Glen, S. (2019). Decision Tree vs Random Forest vs Gradient Boosting Machines: Explained Simply. Available at: <https://www.datasciencecentral.com/profiles/blogs/decision-tree-vs-random-forest-vs-boosted-trees-explained>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Google (n.d.). Cloud Tensor Processing Units (TPUs). Retrieved in October 2021 from <https://cloud.google.com/tpu/docs/tpus>
- Gupta, P. (2017). Decision Trees in Machine Learning. Available at: <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>
- Hasan, S., Shamsuddin, S. M., & Lopes, N. (2014). Machine learning big data framework and analytics for big data problems. *International Journal of Advances in Soft Computing and its Applications*, 6(2).
- Hitaj, B., Gasti, P., Ateniese, G., & Perez Cruz, F. (2019). Passgan: A deep learning approach for password guessing. *International Conference on Applied Cryptography and Network*.
- Hulten, G. (2019). *Building Intelligent Systems: A Guide to Machine Learning Engineering*. Apress.
- IBM Cloud education (October 20th, 2020). Convolutional Neural Networks. Available at: <https://www.ibm.com/cloud/learn/convolutional-neural-networks>
- Israel, R., Kelly, B., & Moskowitz, T. (2020). Can machines 'Learn' finance? *Journal of Investment Management*. Available at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3624052](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3624052)
- Jansen, S. (2020). *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python*. Packt Publishing Ltd.
- Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84). Springer.
- Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D., & Tzovaras, D. (2021). Machine learning and deep learning in smart manufacturing: the smart grid paradigm. *Computer Science Review*, 40, 100341.
- Lange, O., & Perez, L. (2020). Traffic prediction with advanced Graph Neural Networks. Available at: <https://deepmind.com/blog/article/traffic-prediction-with-advanced-graph-neural-networks>
- López-Manuel, L., León-Mateos, F., & Sartal, A. (2020). Closing Loops, Easing Strains: Industry 4.0's Potential for Overcoming Challenges of Circularity in Manufacturing Environments. In *Circular Economy for the Management of Operations* (pp. 23-47). CRC Press.
- Monostori, L., Markus, A., Van Brussel, H., & Westkämpfer E. (1996). Machine Learning Approaches to Manufacturing. *CIRP Annals*, 45(2), 675-712.
- Osiński, B., & Budek, K. (2018). What is reinforcement learning? The complete guide. Available at: <https://deepsense.ai/what-is-reinforcement-learning-thecomplete-guide/>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12, 2825–2830
- Popel, M. et al. (2020). Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11.
- Ricquebourg, V., Menga, D., Durand, D., Marhic, B., Delauche, L., & Logé, C. (2007). The smart home concept: Our immediate future. In *2006 1st IEEE international conference on E-learning in industrial electronics* (pp. 23–28). ICIELIE.
- Rodrigues Jr J. F., Florea, L., De Oliveira, M. C., Diamond, D., & Oliveira Jr, O. N. (2019). A survey on big data and machine learning for chemistry. Available at: [arXiv:1904.10370](https://arxiv.org/abs/1904.10370).
- Rosov, S. (2017). Machine Learning, Artificial Intelligence, and Robo-Advisers: The Future of Finance? Available at: <https://blogs.cfainstitute.org/marketintegrity/2017/07/12/machine-learning-artificial-intelligence-and- robo-advisers-the-future-of-finance/>
- Scikit-learn (n.d.). 1.4. Support Vector Machines. Retrieved in October 2021 from <https://scikit-learn.org/stable/modules/svm.html>

- Smith, R. (2020). How Machine Learning Will Transform Engineering. Available at: <https://becominghuman.ai/how-machine-learning-will-transform-engineering-e239e5774408>
- Sören Dramsch, J. (2020). 70 years of machine learning in geoscience in review. *Advances in Geophysics*, 61, 1-55,
- Stokes, J. M., et al. (2020). A deep learning approach to antibiotic discovery. *Cell*, 180(4), 688-702.e13.
- Swaminathan, S. (2018). Logistic Regression — Detailed Overview. Available at: <https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>
- Tang, B., Pan, Z., Yin, K., & Khateeb, A. (2019). Recent advances of deep learning in bioinformatics and computational biology. *Frontiers in Genetics*, 10, 214.
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018.
- Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Science*, 28, 31–36
- Wuest, T., Weimer, D., Irgens, C., & Thoben, K.-D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1), 23–45.
- Yang, Y., & Yan, J. (2018). *Hands-On Data Science with Anaconda: Utilize the right mix of tools to create high-performance data science applications*. Packt Publishing Ltd.