1 Introduction

Machine learning is about learning, reasoning, and acting based on data. This is done by constructing computer programs that process the data, extract useful information, make predictions regarding unknown properties, and suggest actions to take or decisions to make. What turns data analysis into machine learning is that the process is automated and that the computer program is learnt from data. This means that generic computer programs are used, which are adapted to application-specific circumstances by automatically adjusting the settings of the program based on observed, so-called training data. It can therefore be said that machine learning is a way of programming by example. The beauty of machine learning is that it is quite arbitrary what the data represents, and we can design general methods that are useful for a wide range of practical applications in different domains. We illustrate this via a range of examples below.

The ‘generic computer program’ referred to above corresponds to a mathematical model of the data. That is, when we develop and describe different machine learning methods, we do this using the language of mathematics. The mathematical model describes a relationship between the quantities involved, or variables, that correspond to the observed data and the properties of interest (such as predictions, actions, etc.) Hence, the model is a compact representation of the data that, in a precise mathematical form, captures the key properties of the phenomenon we are studying. Which model to make use of is typically guided by the machine learning engineer’s insights generated when looking at the available data and the practitioner’s general understanding of the problem. When implementing the method in practice, this mathematical model is translated into code that can be executed on a computer. However, to understand what the computer program actually does, it is important also to understand the underlying mathematics.

As mentioned above, the model (or computer program) is learnt based on the available training data. This is accomplished by using a learning algorithm which is capable of automatically adjusting the settings, or parameters, of the model to agree with the data. In summary, the three cornerstones of machine learning are:

1. The data
2. The mathematical model
3. The learning algorithm.

In this introductory chapter, we will give a taste of the machine learning problem by illustrating these cornerstones with a few examples. They come from different application domains and have different properties, but nevertheless, they can all be addressed using similar techniques from machine learning. We also give some
1 Introduction

advice on how to proceed through the rest of the book and, at the end, provide references to good books on machine learning for the interested reader who wants to dig further into this topic.

1.1 Machine Learning Exemplified

Machine learning is a multifaceted subject. We gave a brief and high-level description of what it entails above, but this will become much more concrete as we proceed throughout this book and introduce specific methods and techniques for solving various machine learning problems. However, before digging into the details, we will try to give an intuitive answer to the question ‘What is machine learning?’ by discussing a few application examples of where it can (and has) been used. We start with an example related to medicine, more precisely cardiology.

Example 1.1 Automatically diagnosing heart abnormalities

The leading cause of death globally is conditions that affect the heart and blood vessels, collectively referred to as cardiovascular diseases. Heart problems often influence the electrical activity of the heart, which can be measured using electrodes attached to the body. The electrical signals are reported in an electrocardiogram (ECG). In Figure 1.1 we show examples of (parts of) the measured signals from three different hearts. The measurements stem from a healthy heart (top), a heart suffering from atrial fibrillation (middle), and a heart suffering from right bundle branch block (bottom). Atrial fibrillation makes the heart beat without rhythm, making it hard for the heart to pump blood in a normal way. Right bundle branch block corresponds to a delay or blockage in the electrical pathways of the heart.

![ECG examples](image)

By analysing the ECG signal, a cardiologist gains valuable information about the condition of the heart, which can be used to diagnose the patient and plan the treatment.
To improve the diagnostic accuracy, as well as to save time for the cardiologists, we can ask ourselves if this process can be automated to some extent. That is, can we construct a computer program which reads in the ECG signals, analyses the data, and returns a prediction regarding the normality or abnormality of the heart? Such models, capable of accurately interpreting an ECG examination in an automated fashion, will find applications globally, but the needs are most acute in low- and middle-income countries. An important reason for this is that the population in these countries often do not have easy and direct access to highly skilled cardiologists capable of accurately carrying out ECG diagnoses. Furthermore, cardiovascular diseases in these countries are linked to more than 75% of deaths.

The key challenge in building such a computer program is that it is far from obvious which computations are needed to turn the raw ECG signal into a prediction about the heart condition. Even if an experienced cardiologist were to try to explain to a software developer which patterns in the data to look for, translating the cardiologist’s experience into a reliable computer program would be extremely challenging.

To tackle this difficulty, the machine learning approach is to instead teach the computer program through examples. Specifically, instead of asking the cardiologist to specify a set of rules for how to classify an ECG signal as normal or abnormal, we simply ask the cardiologist (or a group of cardiologists) to label a large number of recorded ECG signals with labels corresponding to the underlying heart condition. This is a much easier (albeit possibly tedious) way for the cardiologists to communicate their experience and encode it in a way that is interpretable by a computer.

The task of the learning algorithm is then to automatically adapt the computer program so that its predictions agree with the cardiologists’ labels on the labelled training data. The hope is that, if it succeeds on the training data (where we already know the answer), then it should be possible to use the predictions made by the program on previously unseen data (where we do not know the answer) as well.

This is the approach taken by Ribeiro et al. (2020), who developed a machine learning model for ECG prediction. In their study, the training data consists of more than 2,300,000 ECG records from almost 1,700,000 different patients from the state of Minas Gerais in Brazil. More specifically, each ECG corresponds to 12 time series (one from each of the 12 electrodes that were used in conducting the exam) of a duration between 7 to 10 seconds each, sampled at frequencies ranging from 300 Hz to 600 Hz. These ECGs can be used to provide a full evaluation of the electrical activity of the heart, and it is indeed the most commonly used test in evaluating the heart. Importantly, each ECG in the dataset also comes with a label sorting it into different classes – no abnormalities, atrial fibrillation, right bundle branch block, etc. – according to the status of the heart. Based on this data, a machine learning model is trained to automatically classify a new ECG recording without requiring a human doctor to be involved. The model used is a deep neural network, more specifically a so-called residual network, which is commonly used for images. The researchers adapted this to work for the ECG signals of relevance for this study. In Chapter 6, we introduce deep learning models and their training algorithms.

Evaluating how a model like this will perform in practice is not straightforward. The approach taken in this study was to ask three different cardiologists with...
1 Introduction

Experience in electrocardiography to examine and classify 827 ECG recordings from distinct patients. This dataset was then evaluated by the algorithm, two 4th year cardiology residents, two 3rd year emergency residents, and two 5th year medical students. The average performance was then compared. The result was that the algorithm achieved better or the same result when compared to the human performance on classifying six types of abnormalities.

Before we move on, let us pause and reflect on the example introduced above. In fact, many concepts that are central to machine learning can be recognised in this example.

As we mentioned above, the first cornerstone of machine learning is the data. Taking a closer look at what the data actually is, we note that it comes in different forms. First, we have the training data which is used to train the model. Each training data point consists of both the ECG signal, which we refer to as the input, and its label corresponding to the type of heart condition seen in this signal, which we refer to as the output. To train the model, we need access to both the inputs and the outputs, where the latter had to be manually assigned by domain experts (or possibly some auxiliary examination). Training a model from labelled data points is therefore referred to as supervised learning. We think of the learning as being supervised by the domain expert, and the learning objective is to obtain a computer program that can mimic the labelling done by the expert. Second, we have the (unlabelled) ECG signals that will be fed to the program when it is used 'in production'. It is important to remember that the ultimate goal of the model is to obtain accurate predictions in this second phase. We say that the predictions made by the model must generalise beyond the training data. How to train models that are capable of generalising, and how to evaluate to what extent they do so, is a central theoretical question studied throughout this book (see in particular Chapter 4).

We illustrate the training of the ECG prediction model in Figure 1.2. The general structure of the training procedure is, however, the same (or at least very similar) for all supervised machine learning problems.

Another key concept that we encountered in the ECG example is the notion of a classification problem. Classification is a supervised machine learning task which amounts to predicting a certain class, or label, for each data point. Specifically, for classification problems, there are only a finite number of possible output values. In the ECG example, the classes correspond to the type of heart condition. For instance, the classes could be ‘normal’ or ‘abnormal’, in which case we refer to it as a binary classification problem (only two possible classes). More generally, we could design a model for classifying each signal as either ‘normal’, or assign it to one of a predetermined set of abnormalities. We then face a (more ambitious) multi-class classification problem.

Classification is, however, not the only application of supervised machine learning that we will encounter. Specifically, we will also study another type of problem referred to as regression problems. Regression differs from classification in that the
1.1 *Machine Learning Exemplified*

![Figure 1.2](image-url) Illustrating the supervised machine learning process with training to the left and then the use of the trained model to the right. **Left:** Values for the unknown parameters of the model are set by the learning algorithm such that the model best describes the available training data. **Right:** The learned model is used on new, previously unseen data, where we hope to obtain a correct classification. It is thus essential that the model is able to generalise to new data that is not present in the training data.

output (that is, the quantity that we want the model to predict) is a numerical value. We illustrate with an example from material science.

**Example 1.2 Formation energy of crystals**

Much of our technological development is driven by the discovery of new materials with unique properties. Indeed, technologies such as touch screens and batteries for electric vehicles have emerged due to advances in materials science. Traditionally, materials discovery was largely done through experiments, but this is both time consuming and costly, which limited the number of new materials that could be found. Over the past few decades, computational methods have therefore played an increasingly important role. The basic idea behind computational materials science is to screen a very large number of hypothetical materials, predict various properties of interest by computational methods, and then attempt to experimentally synthesise the most promising candidates.

Crystalline solids (or, simply, crystals) are a central type of inorganic material. In a crystal, the atoms are arranged in a highly ordered microscopic structure. Hence, to understand the properties of such a material, it is not enough to know the proportion of each element in the material, but we also need to know how these elements (or atoms) are arranged into a crystal. A basic property of interest when considering a hypothetical material is therefore the **formation energy** of the crystal. The formation energy can be thought of as the energy that nature needs to spend to form the crystal from the individual elements. Nature strives to find a minimum energy configuration. Hence, if a certain crystal structure is predicted to have a formation energy that is significantly larger than alternative crystals composed of the same elements, then it is unlikely that it can be synthesised in a stable way in practice.

A classical method (going back to the 1960s) that can be used for computing the formation energy is so-called **density functional theory (DFT)**. The DFT method, which is based on quantum mechanical modelling, paved the way for the first...
1 Introduction

breakthrough in computational materials science, enabling high throughput screening for materials discovery. That being said, the DFT method is computationally very expensive, and even with modern supercomputers, only a small fraction of all potentially interesting materials have been analysed.

To handle this limitation, there has been much recent interest in using machine learning for materials discovery, with the potential to result in a second computational revolution. By training a machine learning model to, for instance, predict the formation energy — but in a fraction of the computational time required by DFT — a much larger range of candidate materials can be investigated.

As a concrete example, Faber et al. (2016) used a machine learning method referred to as kernel ridge regression (see Chapter 8) to predict the formation energy of around 2 million so-called elpasolite crystals. The machine learning model is a computer program which takes a candidate crystal as input (essentially, a description of the positions and elemental types of the atoms in the crystal) and is asked to return a prediction of the formation energy. To train the model, 10 000 crystals were randomly selected, and their formation energies were computed using DFT. The model was then trained to predict formation energies to agree as closely as possible with the DFT output on the training set. Once trained, the model was used to predict the energy on the remaining ~99.5% of the potential elpasolites. Among these, 128 new crystal structures were found to have a favorable energy, thereby being potentially stable in nature.

Comparing the two examples discussed above, we can make a few interesting observations. As already pointed out, one difference is that the ECG model is asked to predict a certain class (say, normal or abnormal), whereas the materials discovery model is asked to predict a numerical value (the formation energy of a crystal). These are the two main types of prediction problems that we will study in this book, referred to as classification and regression, respectively. While conceptually similar, we often use slight variations of the underpinning mathematical models, depending on the problem type. It is therefore instructive to treat them separately.

Both types are supervised learning problems, though. That is, we train a predictive model to mimic the predictions made by a 'supervisor'. However, it is interesting to note that the supervision is not necessarily done by a human domain expert. Indeed, for the formation energy model, the training data was obtained by running automated (but costly) density functional theory computations. In other situations, we might obtain the output values naturally when collecting the training data. For instance, assume that you want to build a model for predicting the outcome of a soccer match based on data about the players in the two teams. This is a classification problem (the output is ‘win’, ‘lose’, or ‘tie’), but the training data does not have to be manually labelled, since we get the labels directly from historical matches. Similarly, if you want to build a regression model for predicting the price of an apartment based on its size, location, condition, etc., then the output (the price) is obtained directly from historical sales.

Finally, it is worth noting that, although the examples discussed above correspond to very different application domains, the problems are quite similar from a machine learning perspective. Indeed, the general procedure outlined in Figure 1.2 is also
1.1 Machine Learning Exemplified

applicable, with minor modifications, to the materials discovery problem. This generality and versatility of the machine learning methodology is one of its main strengths and beauties.

In this book, we will make use of statistics and probability theory to describe the models used for making predictions. Using probabilistic models allows us to systematically represent and cope with the uncertainty in the predictions. In the examples above, it is perhaps not obvious why this is needed. It could (perhaps) be argued that there is a ‘correct answer’ both in the ECG problem and the formation energy problem. Therefore, we might expect that the machine learning model should be able to provide a definite answer in its prediction. However, even in situations when there is a correct answer, machine learning models rely on various assumptions, and they are trained from data using computational learning algorithms. With probabilistic models, we are able to represent the uncertainty in the model’s predictions, whether it originates from the data, the modelling assumptions, or the computation. Furthermore, in many applications of machine learning, the output is uncertain in itself, and there is no such thing as a definite answer. To highlight the need for probabilistic predictions, let us consider an example from sports analytics.

Example 1.3 Probability of scoring a goal in soccer

Soccer is a sport where a great deal of data has been collected on how individual players act throughout a match, how teams collaborate, how they perform over time, etc. All this data is used to better understand the game and to help players reach their full potential.

Consider the problem of predicting whether or not a shot results in a goal. To this end, we will use a rather simple model, where the prediction is based only on the player’s position on the field when taking the shot. Specifically, the input is given by the distance from the goal and the angle between two lines drawn from the player’s position to the goal posts; see Figure 1.3. The output corresponds to whether or not the shot results in a goal, meaning that this is a binary classification problem.
1 Introduction

Clearly, knowing the player’s position is not enough to definitely say if the shot will be successful. Still, it is reasonable to assume that it provides some information about the chance of scoring a goal. Indeed, a shot close to the goal line with a large angle is intuitively more likely to result in a goal than one made from a position close to the sideline. To acknowledge this fact when constructing a machine learning model, we will not ask the model to predict the outcome of the shot but rather to predict the probability of a goal. This is accomplished by using a probabilistic model which is trained by maximising the total probability of the observed training data with respect to the probabilistic predictions. For instance, using a so-called logistic regression model (see Chapter 3) we obtain a predicted probability of scoring a goal from any position, illustrated using a heat map in the right panel in Figure 1.3.

The supervised learning problems mentioned above were categorised as either classification or regression problems, depending on the type of output. These problem categories are the most common and typical instances of supervised machine learning, and they will constitute the foundation for most methods discussed in this book. However, machine learning is in fact much more general and can be used to build complex predictive models that do not naturally fit into either the classification or the regression category. To whet the appetite for further exploration of the field of machine learning, we provide two such examples below. These examples go beyond the specific problem formulations that we explicitly study in this book, but they nevertheless build on the same core methodology.

In the first of these two examples, we illustrate a computer vision capability, namely how to classify each individual pixel of an image into a class describing the object that the pixel belongs to. This has important applications in, for example, autonomous driving and medical imaging. When compared to the earlier examples, this introduces an additional level of complexity, in that the model needs to be able to handle spatial dependencies across the image in its classifications.

Example 1.4 Pixel-wise class prediction

When it comes to machine vision, an important capability is to be able to associate each pixel in an image with a corresponding class; see Figure 1.4 for an illustration in an autonomous driving application. This is referred to as semantic segmentation. In autonomous driving, it is used to separate cars, road, pedestrians, etc. The output is then used as input to other algorithms, for instance for collision avoidance. When it comes to medical imaging, semantic segmentation is used, for instance, to tell apart different organs and tumors.

To train a semantic segmentation model, the training data consist of a large number of images (inputs). For each such image, there is a corresponding output image of the same size, where each pixel has been labelled by hand to belong to a certain class. The supervised machine learning problem then amounts to using this data to find a mapping that is capable of taking a new unseen image and produce a corresponding output in the form of a predicted class for each pixel. Essentially, this is a type of classification problem, but all pixels need to be classified simultaneously while respecting the spatial dependencies across the image to result in a coherent segmentation.
1.1 Machine Learning Exemplified

The bottom part of Figure 1.4 shows the prediction generated by such an algorithm, where the aim is to classify each pixel as either car (blue), traffic sign (yellow), pavement (purple), or tree (green). The best performing solutions for this task today rely on cleverly crafted deep neural networks (see Chapter 6).

In the final example, we raise the bar even higher, since here the model needs to be able to explain dependencies not only over space, but also over time, in a so-called spatio-temporal problem. These problems are finding more and more applications as we get access to more and more data. More precisely, we look into the problem of how to build probabilistic models capable of better estimating and forecasting air pollution across time and space in a city, in this case London.

Example 1.5 Estimating air pollution levels across London

Roughly 91% of the world’s population lives in places where the air quality levels are worse than those recommended by the world health organisation. Recent estimates indicate that 4.2 million people die each year from stroke, heart disease, lung cancer, and chronic respiratory diseases caused by ambient air pollution.

A natural first step in dealing with this problem is to develop technology to measure and aggregate information about the air pollution levels across time and space. Such information enables the development of machine learning models to better estimate and accurately forecast air pollution, which in turn permits suitable interventions. The work that we feature here sets out to do this for the city of London, where more than 9 000 people die early every year as a result of air pollution.

Air quality sensors are now – as opposed to the situation in the recent past – available at relatively low cost. This, combined with an increasing awareness of the problem, has caused interested companies, individuals, non-profit organisations, and community groups to contribute by setting up sensors and making the data...
1 Introduction

More specifically, the data in this example comes from a sensor network of ground sensors providing hourly readings of NO\textsubscript{2} and hourly satellite data at a spatial resolution of 7 km \times 7 km. The resulting supervised machine learning problem is to build a model that can deliver forecasts of the air pollution level across time and space. Since the output – pollution level – is a continuous variable, this is a type of regression problem. The particularly challenging aspect here is that the measurements are reported at different spatial resolutions and on varying timescales.

The technical challenge in this problem amounts to merging the information from many sensors of different kinds reporting their measurements on different spatial scales, sometimes referred to as a multi-sensor multi-resolution problem. Besides the problem under consideration here, problems of this kind find many different applications. The basis for the solution providing the estimates exemplified in Figure 1.5 is the Gaussian process (see Chapter 9).

Figure 1.5 illustrates the output from the Gaussian process model in terms of spatio-temporal estimation and forecasting of NO\textsubscript{2} levels in London. To the left, we have the situation on February 19, 2019 at 11:00 using observations from both ground sensors providing hourly readings of NO\textsubscript{2} and from satellite data. To the right, we have the situation on 19 February 2019 at 17:00 using only the satellite data.

The Gaussian process is a non-parametric and probabilistic model for nonlinear functions. Non-parametric means that it does not rely on any particular parametric functional form to be postulated. The fact that it is a probabilistic model means that it is capable of representing and manipulating uncertainty in a systematic way.

1.2 About This Book

The aim of this book is to convey the spirit of supervised machine learning, without requiring any previous experience in the field. We focus on the underlying mathematics as well as the practical aspects. This book is a textbook; it is not a reference work or a programming manual. It therefore contains only a careful (yet comprehensive) selection of supervised machine learning methods and no programming code. There are by now many well-written and well-documented