
CHAPTER 1

MODULE 1

The goal of this chapter is to explore the foundational concepts of machine learning, focusing on how algorithms can transform data into knowledge. We delve into the practical applications of supervised and unsupervised learning, equipping you with the skills to implement these techniques using Python tools for effective data analysis and prediction.

Observation 1.0.1 (Learning Objectives)

Learning Objectives:

- Analyze data patterns to make future predictions.
- Design systems for supervised and unsupervised learning.
- Implement machine learning algorithms using Python tools.

1.1 OVERVIEW

The goal basically is to learn machine learning using Python libraries such as Scikit-Learn and PyTorch.

1.2 GIVING COMPUTERS THE ABILITY TO LEARN FROM DATA

So, basically in this scenario we will do the following:

- Implement machine learning algorithms using Python tools.
- Design systems for supervised and unsupervised learning.
- Analyze data patterns to make future predictions.

1.3 INTRODUCTION

Definition 1.3.1 (Machine Learning)

Machine learning is the *application and science of algorithms that make sense of data*.

The goal of this section is to introduce some of the main concepts and different types of machine learning, together with a basic introduction to the relevant terminology. The goal is to establish the groundwork for successfully using machine learning techniques for practical problem solving.

1.3.1 THE THREE DIFFERENT TYPES OF MACHINE LEARNING

In this age, we have a large amount of structured and non-structured data.

Definition 1.3.2 (Structured and Non-structured Data)

Structured data refers to *information that is organized in a predefined format*, typically within a fixed schema, such as rows and columns in a database or spreadsheet.

In contrast, **non-structured data** *lacks a predefined format or structure and exists in its*

native, raw state. It is typically qualitative and encompasses a wide variety of formats such as text documents, emails, social media posts, images, videos, and audio files.

One of the main goals of machine learning is to extract knowledge from data in order to make predictions.

Observation 1.3.1 (Use of Machine Learning)

Machine Learning offers a more efficient alternative for capturing the knowledge in data to improve the performance of predictive models and make data-driven decisions.

Idea 1.3.1 (Note on the use of Machine Learning and some of its Applications)

Also, notable progress has been made in medical applications; for example, researchers demonstrated that deep learning models can detect skin cancer with near-human accuracy [link to article](#).

Another milestone was recently achieved by researchers at DeepMind, who used deep learning to predict 3D protein structures, outperforming physics-based approaches by a substantial margin [link to article](#).

There are three types of machine learning: **supervised learning**, **unsupervised learning**, and **reinforcement learning**. There are fundamental differences between these three types of machine learning and we will look at each of them in detail with some notes on its possible applications.

Three Types of Machine Learning	
Supervised learning	• Labeled data
	• Direct feedback
	• Predict outcome/future
Unsupervised learning	• No labels/targets
	• No feedback
	• Find hidden structure in data
Reinforcement learning	• Decision process
	• Reward system
	• Learn series of actions

Table 1.1: Three Types of Machine Learning.

1.3.2 SUPERVISED LEARNING

The main goal of **supervised learning** is to *learn a model from labeled training data that allows us to make predictions about unseen or future data.*

The term *supervised* refers to a set of training examples (data inputs), where the desired output signals (labels) are already known.

Definition 1.3.3 (Supervised Learning)

Supervised learning is the *process of modeling the relationship between data inputs and the labels.*

We can think of supervised learning as *label learning*.

Example 1.3.1 (Example of Supervised Learning)

Lets consider the example of email spam filtering. We can train a model using a supervised machine learning algorithm on a corpus of labeled emails, which are correctly marked as spam or non-spam.

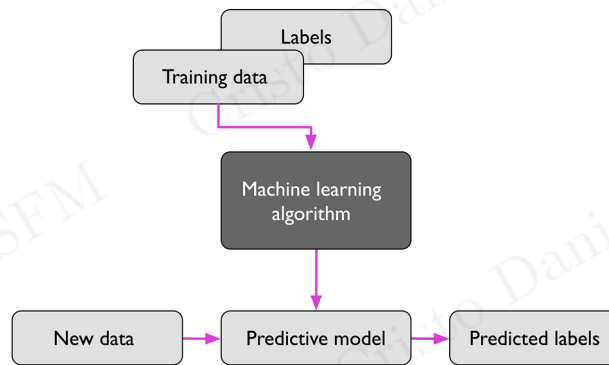


Figure 1.1: General Process of Supervised Learning

To predict whether a new email belongs to either of the two categories, a supervised learning task with discrete class labels, such as in the previous email spam filtering example.

This is called a **classification task**. Another subcategory of supervised learning is **regression**, where the outcome signal is a continuous value.

1.3.3 CLASSIFICATION FOR PREDICTING CLASS LABELS

Definition 1.3.4 (Classification)

Classification is a subcategory of supervised learning, where the goal is to predict the categorical class labels of new instances or data points based on past observations.

Those class labels are discrete, unordered values that can be understood as the group memberships of the data points. The previously mentioned example of email spam detection represents a typical example of a binary classification task, where the machine learning algorithm learns a set of rules to distinguish between two possible classes: spam and no-spam emails.

The next illustration shows the concept of binary classification task given 30 training examples; 15 training are labeled as **class A** and the other 15 labeled as **class B**.

Our dataset is two-dimensional, meaning that each example has two values associated with it: x_1 and x_2 . We can use a supervised machine learning algorithm to learn a rule (decision boundary represented as the dashed line) that can separate two classes and classify new data into each of those two categories given its x_1 and x_2 values.

Observation 1.3.2 (Nature of classification of class labels)

When a set of class labels does not have to be of a binary nature. The predictive model learned by a supervised learning algorithm can assign any class label that was presented in the training dataset to a new, unlabeled data point or instance.

Example 1.3.2

An example of a multiclass classification task is handwritten character recognition.

1.3.4 REGRESSION FOR PREDICTING CONTINUOUS OUTCOMES

Definition 1.3.5 (Regression Analysis)

In a **regression analysis**, we are given a number of predictor (*explanatory*) variables and a

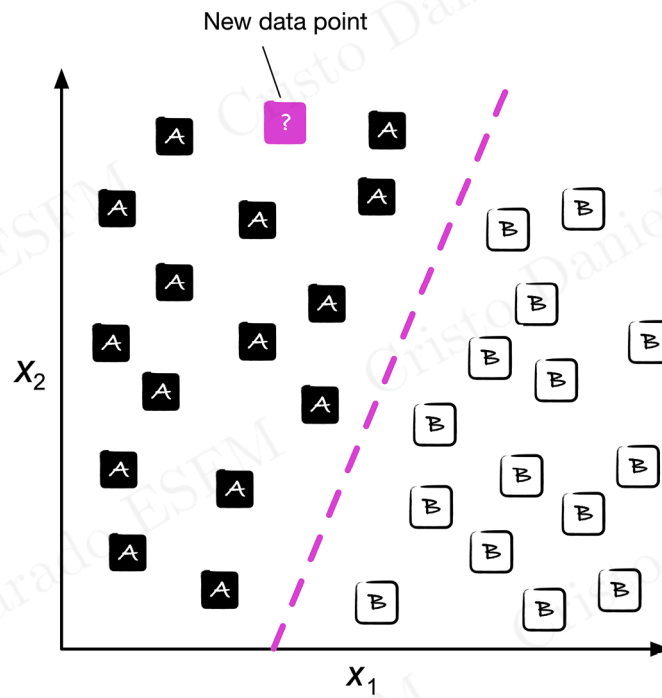


Figure 1.2: Classification in Machine Learning

continuous response variable (**outcome**) and we try to find a relationship between those variables that allows us to predict an outcome.

Observation 1.3.3 (Feature and Target Variables)

In the field of machine learning, the predictor variables are commonly called *features*, and the response variables are referred to as *target variables*.

Example 1.3.3

Let's assume we want to predict the mat SAT scores of students. If there is a relationship between time spent studying for the test and the final scores, we could use it as training data to learn a model that uses the study time to predict the test scores of future students.

Given a feature variable x and a target variable y , we fit a straight line to the data points that minimizes the error in predicting y from x .

We can now use the intercept and slope learned from this data to predict the target variable of new data.

1.4 SOLVING PROBLEMS WITH REINFORCEMENT LEARNING

Definition 1.4.1 (Reinforcement Learning)

Reinforcement Learning has the goal to develop a system (called **agent**) that improves its performance through interactions with the environment.

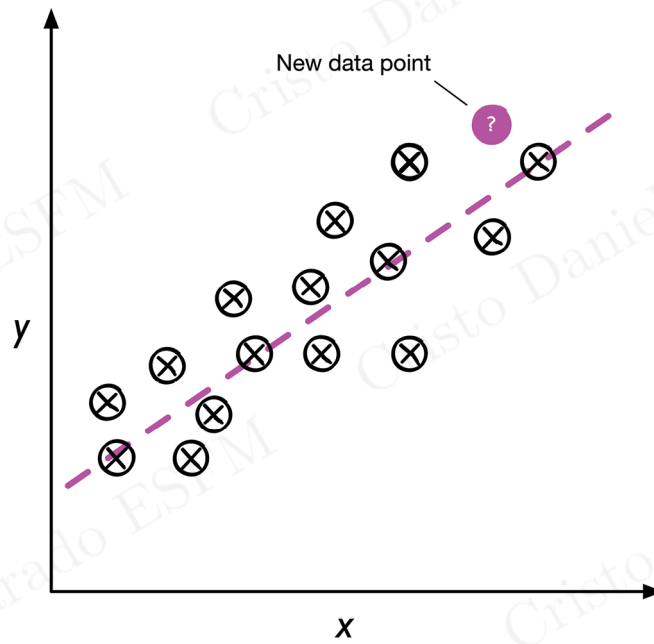


Figure 1.3: Linear Approximation

Observation 1.4.1

Typically, the information coming from a current state of the environment also includes a **reward signal**. With this in mind, reinforcement learning can be viewed as being related to supervised learning.

This feedback is not the correct ground-truth label or value, *but a measure of how well the action is evaluated by a reward function*.

Using a reward signal, a system can measure this reward via an exploratory trial-and-error approach or deliberative planning.

Example 1.4.1

One of the most popular examples of reinforcement learning is **chess program**. The agent decides upon a series of moves and the reward can be defined as win or lose at the end of the game.

In synthesis, the reinforcement learning can be described with the following diagram.

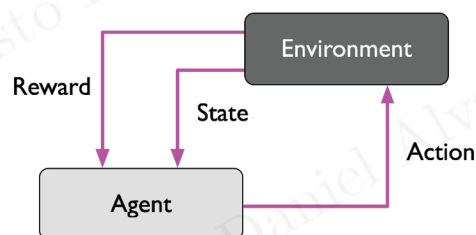


Figure 1.4: Reinforcement Learning Process

There are many different subtypes of reinforcement learning, but in a general scheme the agent tries to maximize the reward through a series of interactions with the environment.

Idea 1.4.1

In summary, reinforcement learning is concerned with learning to choose a series of actions that maximizes the total reward which could be earned immediately after taking an action or via *delayed* feedback.

1.5 UNSUPERVISED LEARNING

Unsupervised learning tries to find the hidden structures of our data. Using supervised learning or reinforcement learning, we have the right answer or a measure of a reward for particular actions, respectively. Unsupervised learning explores the structure of our data to extract meaningful information without the guidance of a known outcome variable.

1.5.1 FINDING SUBGROUPS WITH CLUSTERING

Definition 1.5.1 (Clustering)

Clustering is an exploratory data analysis or pattern discovery technique that organizes information into meaningful subgroups (called **clusters**) without any prior knowledge of group memberships.

Each cluster defines a group of objects that share a certain degree of similarity but are more dissimilar to objects in other clusters which is why clustering is called (sometimes) **unsupervised classification**.

The following scatter plot shows how clustering can organize unlabeled data into three distinct groups or clusters.

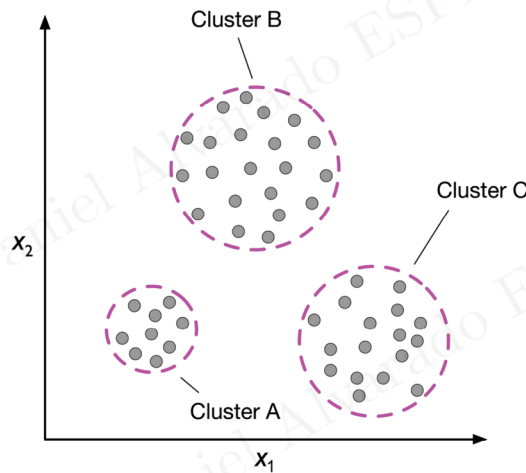


Figure 1.5: Caption

1.5.2 DIMENSIONALITY REDUCTION FOR DATA COMPRESSION

Another subfield of unsupervised learning is dimensionality reduction.

Definition 1.5.2 (Dimensionality Reduction)

Dimensionality reduction is the process to reduce the dimension of the data we are working it, preserving the original and meaningful data needed to interpret it.

Often, we work with data of high dimensionality that can present challenges for limited storage space and the computational performance of machine learning algorithms. The unsupervised dimensionality reduction is commonly used in feature preprocessing to remove noise from data, which can degrade the predictive performance of certain algorithms.

Dimensionality reduction compresses the data into a smaller dimensional subspace while retaining most of the relevant information.

Observation 1.5.1

Sometimes, dimensionality reduction is useful for visualizing data. The following figure is a good and useful example of it.

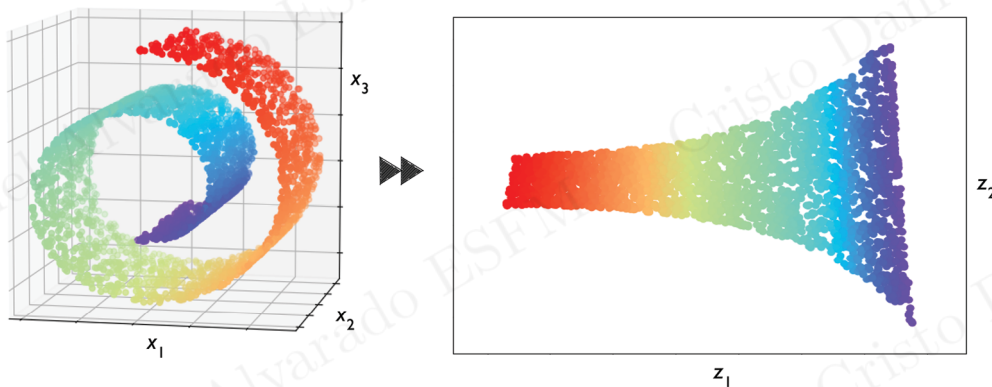


Figure 1.6: Example of Dimensionality Reduction in a Dataset.

1.6 TERMINOLOGY AND NOTATIONS

Definition 1.6.1 (Dataset)

A **dataset** is a *collection of related sets of information that is composed of separate elements* but can be manipulated as a unit by a computer.

One image that contains most of the information that we'll be using through this course is the following:

Example 1.6.1

The Iris dataset contains the measurements of 150 iris flowers from three different species (setosa, versicolor and virginica).

Each flower example represents one row in our dataset and the flower measurements in centimeters are stored as columns, called **features of the dataset**.

Definition 1.6.2 (Feature)

In a dataset, a **feature** is an *input, attribute, measurement or dimensions of something*.

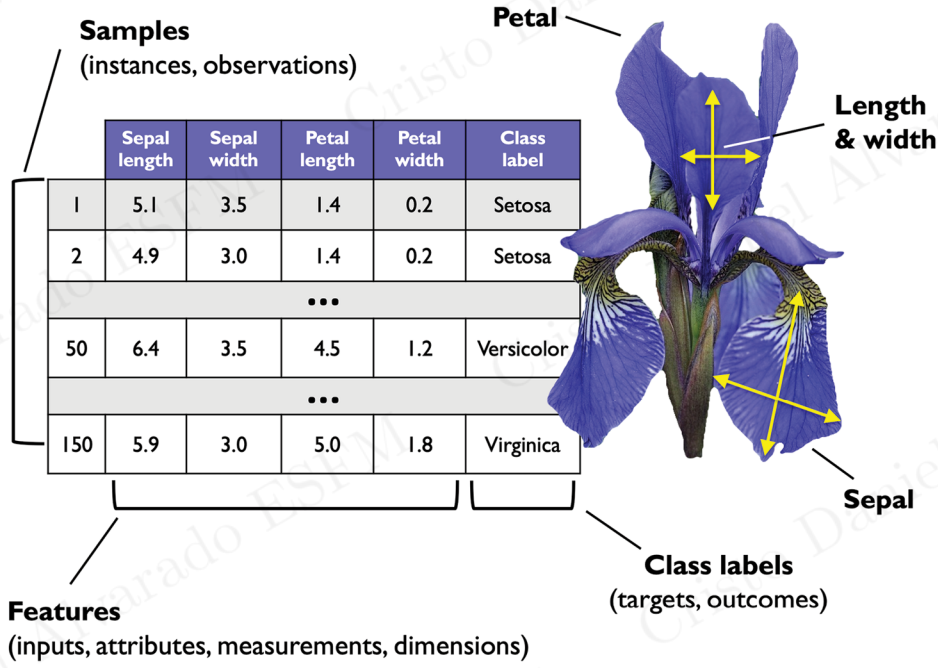


Figure 1.7: Samples, Features, and Class labels.

Idea 1.6.1 (Notation)

The iris dataset consists of 150 examples and four features, we will write them as a 150×4 matrix:

$$X = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

Vectors will be identified as the column matrix $\vec{x} \in \mathbb{R}^{n \times 1}$. To refer to the elements of the matrix we will use the notation $x_i^{(j)}$.

Observation 1.6.1 (Use of Notation)

In the latter idea, $x_1^{(150)}$ refers to the first dimension flower example 150, corresponding to the sepal length.

Each row in the matrix X represents one flower instance and can be written as a four-dimensional row vector:

$$x^{(i)} = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & x_3^{(i)} & x_4^{(i)} \end{bmatrix}$$

Each feature is a 150-dimensional column vector, denoted by x_j :

$$x_j = \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(150)} \end{bmatrix}$$

We represent the target variables (here, class labels) as 150-dimensional column vector:

$$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(150)} \end{bmatrix}$$

where $y^{(i)} \in \{\text{Setosa}, \text{Versicolor}, \text{Virginica}\}$.

1.6.1 MACHINE LEARNING TERMINOLOGY

Definition 1.6.3 (Terminology in Machine Learning)

Some of the following terms are largely used in machine learning:

- **Training example:** A row in a table representing the dataset and synonymous with an observation, record, instance, or sample (in most contexts, sample refers to a collection of training examples).
- **Training:** Model fitting, for parametric models similar to parameter estimation.
- **Feature (x):** A column in a data table or data (design) matrix. Synonymous with predictor, variable, input, attribute, or covariate.
- **Target (y):** Synonymous with outcome, output, response variable, dependent variable, (class) label, and ground truth.
- **Loss function:** Often used synonymously with a cost function. Sometimes the loss function is also called an error function. In some literature, the term 'loss' refers to the loss measured for a single data point, and the cost is a measurement that computes the loss (average or summed) over the entire dataset.

1.7 MACHINE LEARNING SYSTEMS

In this section we'll discuss the other important parts of a machine learning system accompanying the learning algorithm.

1.7.1 PREPROCESSING: GETTING DATA INTO SHAPE

Raw data rarely comes in the form and shape that is necessary for the optimal performance of a learning algorithm. Thus, the preprocessing of the data is one of the most crucial steps in any machine learning application.

Definition 1.7.1 (Preprocessing)

Preprocessing is the *process of converting raw data into data suitable for a machine learning algorithm*.

Example 1.7.1 (Optimal Features in the Iris Dataset)

Let's consider the iris dataset from the previous example. We can think of raw data as a series of flower images from which we want to extract **meaningful features**. Useful features could be:

- Centered around the color of the flowers.
- Height.
- Length.

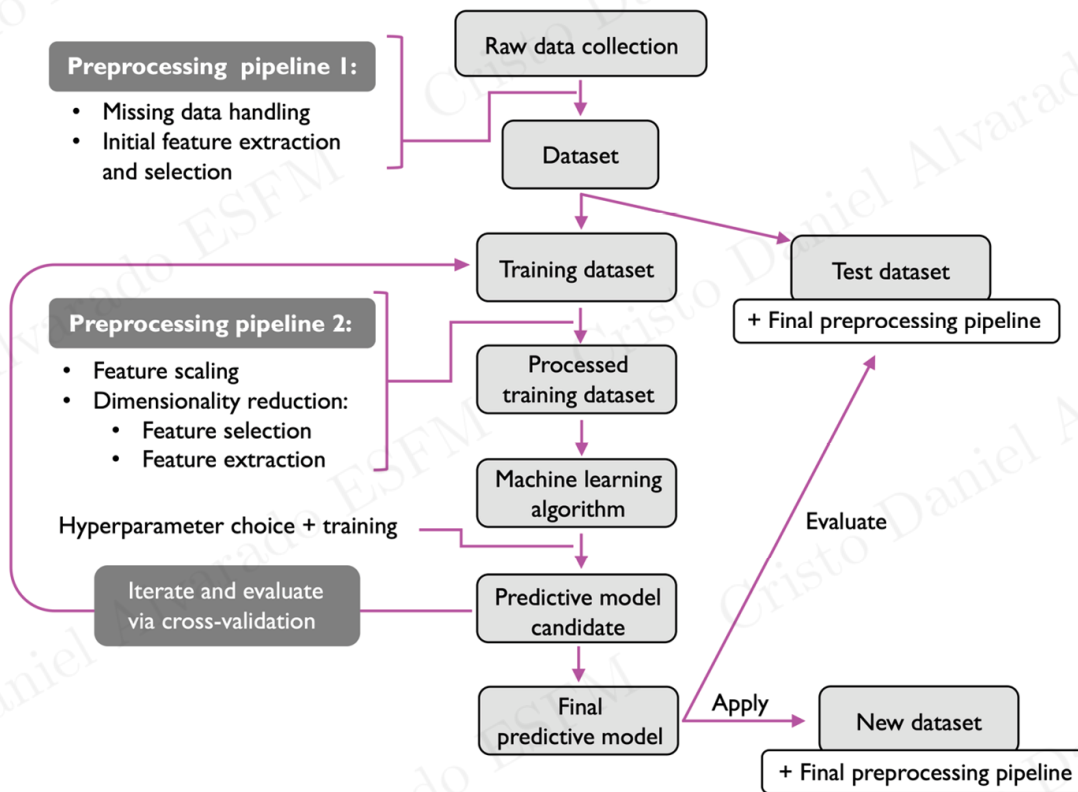


Figure 1.8: Machine Learning System.

- Width.

Observation 1.7.1 (Data Convesion)

Many machine learning models require that the selected features are on the same scale for optimal performance, which is achieved by transforming the features to the range $[0, 1]$.

Other way to achieve this is use standard normal distribution with zero mean and unit variance.

Observation 1.7.2 (Use of Dimensionaly Reduction)

When certain features are highly correlated and therefore, redundant, dimensionaly techniques may be useful for compressing the features onto a lower dimensional subspace.

The main advantage of dimensionality reduction is that less storage space is required, and therefore the learning algorithm can run much faster. This can also improve the performance of a model if the dataset contains a large number of irrelevant features (or noise).

Definition 1.7.2 (Noise)

In data science and statistics, **noise** refers to the *random, uncontrollable, and unexplained variations in the data. It is the part of the data that does not represent the underlying phenomenon you are trying to study or model.*

Observation 1.7.3 (How to Deal with Noise?)

You can rarely eliminate noise completely, but you can manage it.

(1) **Data Cleaning & Preprocessing:** This is the first line of defense.

- **Smoothing:** Applying algorithms to average out random fluctuations in time-series or signal data.
- **Binning:** Grouping numerical values into bins to smooth out small irregularities.
- **Correcting Typos:** Standardizing text entries.

(2) **Collecting More Data:** A larger dataset can sometimes "drown out" the noise, as the true signal becomes stronger with more examples.

(3) **Using Robust Algorithms:** Some machine learning models (like Random Forests) are inherently more resistant to noise than others (like Decision Trees).

(4) **Feature Selection:** Removing irrelevant or redundant features that mostly contain noise.

Idea 1.7.1

To determine whether our machine learning algorithm performs well not only on the training dataset but also generalizes well to new data, *we also want to randomly divide the dataset into separate training and test datasets*. In this way, we can measure whether our machine learning model is performing well.

1.7.2 TRAINING AND SELECTING A PREDICTIVE MODEL

We cannot get learning for free.

Observation 1.7.4

Many different machine learning models have been developed to solve different problem tasks.

The goal is that sometimes is easier to use a certain machine learning model for some tasks than use it for other ones.

Example 1.7.2

Each classification algorithm has its inherent biases, and no single classification model enjoys superiority if we do not make any assumptions about the task.

In summary, it's essential to compare at least a handful of different learning algorithms in order to train and select the best performing model.

Observation 1.7.5

Before deciding which models we shall use, we have to decide upon a metric to measure performance. One commonly used metric is classification accuracy, which is defined as the proportion of correctly classified instances.

Sometimes we may have a machine learning model which was trained using a dataset for model selection, but one question arises: what if we do not use this training dataset? To address this issue, we can use different techniques such as cross-validation.

Definition 1.7.3 (Cross-Validation)

Cross-validation *we divide further a dataset into training and validation subsets in order to estimate the generalization performance of the model.*

Observation 1.7.6 (Hyperparameters)

We also cannot expect that the default parameters of the different learning algorithms provided by software libraries are optimal for our specific problem task. Therefore, we will make frequent use of hyperparameter optimization techniques that help us fine-tune the performance of our model in later sections.

We can think of those hyperparameters as parameters that are not learned from the data but represent the knobs of a model that we can turn to improve its performance.

1.7.3 EVALUATING MODELS AND PREDICTING UNSEEN DATA INSTANCES

After selecting a model that has been fitted on the training dataset, we can use the test dataset to estimate how well it performs on unseen data to determine the **generalization error**.

Definition 1.7.4 (Generalization Error)

Generalization Error *is formally the expected error of the model on a new, random instance drawn from the same underlying data distribution.*

Observation 1.7.7 (Dataset Dependency)

It is important to note that the parameters for previously mentioned procedures, such as feature scaling and dimensionality reduction, **are solely obtained from the training dataset**, and the **same parameters are later reapplied to transform the test dataset as well as any new data instances**—the performance measured on the test data may be overly optimistic otherwise.