Chapter 5 Introduction to Machine learning

5.1 Introduction to artificial intelligence and machine learning

Artificial intelligence (AI) refers to intelligence demonstrated by machine in opposite to human intelligence. The goal of AI is to simulate human intelligence for reasoning and solving complex problems. AI technology can be applied to various area such as healthcare, entertainment, finance, business etc. One example of AI is AlphaGo which is a computer program that plays Go defeating the top human players in the world.

Machine learning (ML) is a subfield of AI which enables a machine to learn from data and to derive knowledge from it. The goal of ML is to make decision and prediction based on the given data. In recent years, AI and ML have shown an increasing impact in applications and research areas including data science in particular. ML provides a more efficient way for capturing the knowledge in data to improve the performance of predictive models and make data-driven decisions.

To summarize, ML can be divided into three types:

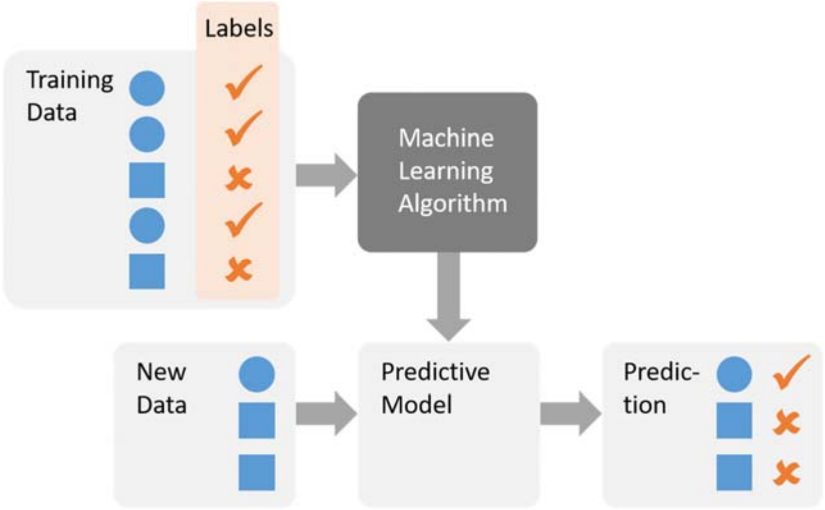
- **Supervised learning** - labelled data, direct feedback, predicted outcome
  e.g. classification for predicting class labels

  e.g. regression for predicting continuous outcomes
- **Reinforcement learning** - decision process, reward system, learn series of actions
  e.g. chess, computer games
- **Unsupervised learning** - no labels, no feedback, find hidden structure in data
  e.g. finding subgroups with clustering

supervised learning

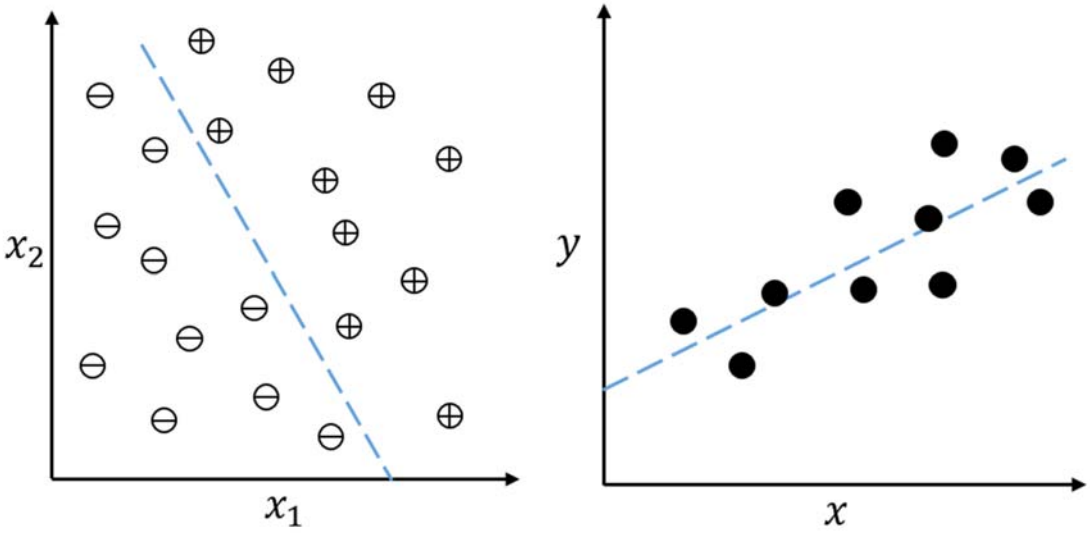
The main goal in supervised learning is to learn a model from labelled training data that allows us to make predictions about unseen or future data. Here, the term "supervised" refers to a set of training examples (data inputs) where the desired output signals (labels) are already known. The following figure summarizes a typical supervised learning workflow, where the labelled training data is passed to a machine learning algorithm for fitting a predictive model that can make predictions on new, unlabelled data inputs.

In this course, we will study into two major tasks of supervised learning. A supervised learning task with discrete class labels is called a **classification** task. Another subcategory of supervised learning is **regression**, where the outcome signal is a continuous value.



Classification is a subcategory of supervised learning where the goal is to predict the categorical class labels of new instances, based on past observations. Those class labels are discrete, unordered values that can be understood as the group memberships of the instances.

The figure on the left below illustrates the concept of a binary classification task given 20 training examples: 10 training examples are labelled as the negative class (minus signs) and 10 training examples are labelled as the positive class (plus signs). In this scenario, our dataset is two-dimensional, which means that each example has two values associated with it: x_1 and x_2 along the horizontal and vertical axes respectively. Now, we can use a supervised machine learning algorithm to learn a rule—the decision boundary represented as a dashed line—that can separate those two classes and classify new data into each of those two categories given its x_1 and x_2 values:



However, the 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, unlabelled instance.

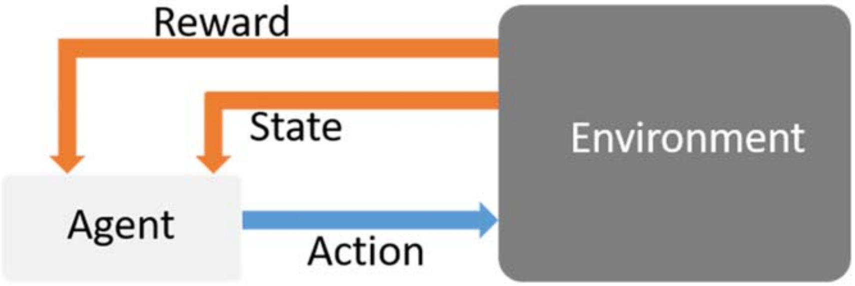
A second type of supervised learning is the prediction of continuous outcomes, which is also called regression analysis. In regression analysis, we are given a number of predictor variables (features) and a continuous response variable (outcome/target), and we try to find a relationship between those variables that allows us to predict an outcome.

The figure on the right above illustrates the concept of linear regression. Given a feature variable, x, and a target variable, y, we fit a straight line to this data that minimizes the distance between the data points and the fitted line. We can now use the intercept and slope learned from this data to predict the target variable of new data.

reinforcement learning

Another type of machine learning is reinforcement learning. In reinforcement learning, the goal is to develop a system (agent) that improves its performance based on interactions with the environment. Since the information about the current state of the environment typically also includes a so-called reward signal, we can think of reinforcement learning as a field related to supervised learning. However, in reinforcement learning, this feedback is not the correct ground truth label or value, but a measure of how well the action was measured by a reward function. Through its interaction with the environment, an agent can then use reinforcement learning to learn a series of actions that maximizes this reward via an exploratory trial-and-error approach or deliberative planning.

A popular example of reinforcement learning is a chess engine. Here, the agent decides upon a series of moves depending on the state of the board (the environment), and the reward can be defined as win or lose at the end of the game.

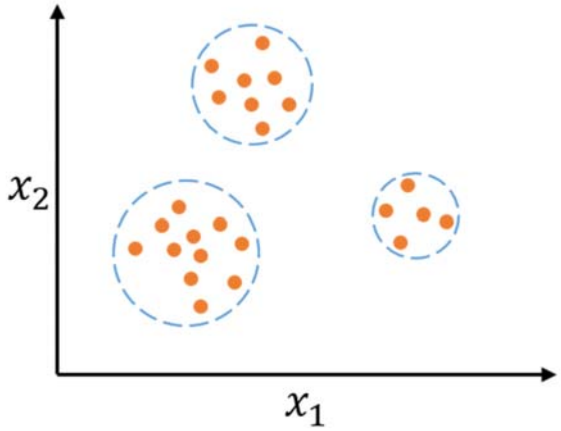


There are many different subtypes of reinforcement learning. However, a general scheme is that the agent in reinforcement learning tries to maximize the reward through a series of interactions with the environment. Each state can be associated with a positive or negative reward, and a reward can be defined as accomplishing an overall goal, such as winning or losing a game of chess. For instance, in chess, the outcome of each move can be thought of as a different state of the environment.

unsupervised learning

In supervised learning, we know the right answer beforehand when we train a model, and in reinforcement learning, we define a measure of reward for particular actions carried out by the agent. In unsupervised learning, however, we are dealing with unlabelled data or data of unknown structure. Using unsupervised learning techniques, we are able to explore the structure of our data to extract meaningful information without the guidance of a known outcome variable or reward function.

One example is finding subgroups with clustering. Clustering is an exploratory data analysis technique that allows us to organize a pile of information into meaningful subgroups (clusters) without having any prior knowledge of their group memberships. Each cluster that arises during the analysis 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 also sometimes called unsupervised classification. Clustering is a great technique for structuring information and deriving meaningful relationships from data. For example, it allows marketers to discover customer groups based on their interests, in order to develop distinct marketing programs. The following figure illustrates how clustering can be applied to organizing unlabelled data into three distinct groups based on the similarity of their features, x_1 and x_2:

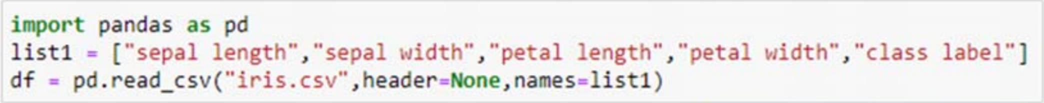


In this course, however, we will not go into unsupervised learning and reinforcement learning due to its complexity in both theory and techniques. Students only need to understand the basic concepts and the difference between the three types of machine learning.

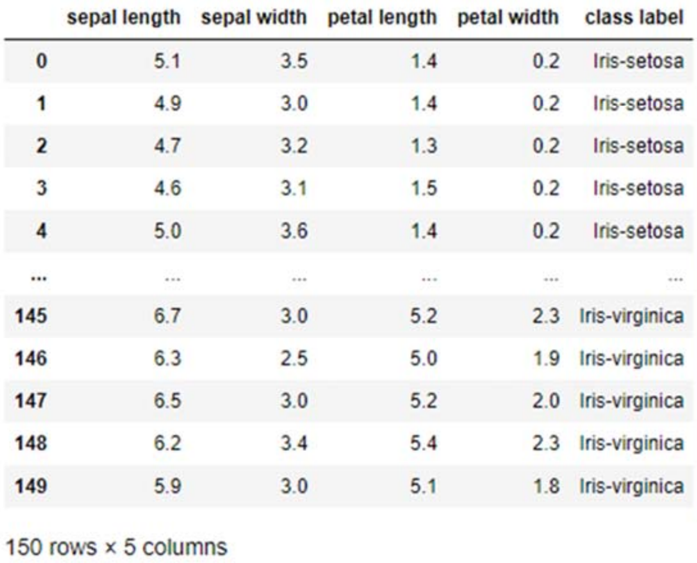
5.2 Basic terminology and notation

Before we study the algorithm of machine learning, we will look into the basic terminology including the common terms referring to different aspects of a dataset and also the mathematical notations. To make it more precise, we will use a very classical and popular dataset of Iris flower in many data science or machine learning textbook.

The dataset "iris.csv" contains the measurements of 150 Iris flowers from three different species—Setosa, Versicolor, and Virginica. Here, each flower example represents one row in our dataset, and the flower measurements in centimeters are stored as columns, which we also call the **features** of the dataset. These include sepal length, sepal width, petal length and petal width. The last column is the **class labels** which contain the species of each flower.

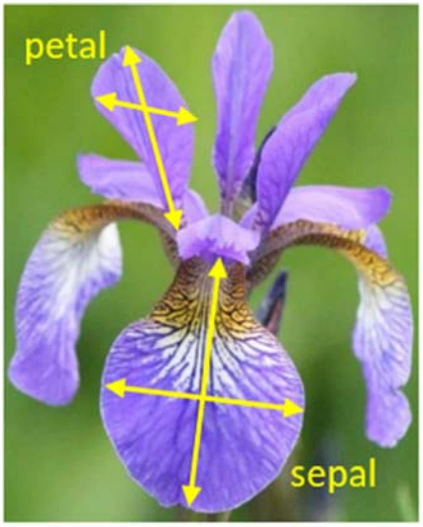


```
import pandas as pd
list1 = ["sepal length","sepal width","petal length","petal width","class label"]
df = pd.read_csv("iris.csv",header=None,names=list1)
```



	sepal length	sepal width	petal length	petal width	class label
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns



We can represent the features part of the dataset using a matrix. The number of rows refers to the number of samples. The number of columns refers to the number of attributes in the features. In the Iris example, we have 150 samples with 4 attributes (class label is not included). This gives us a 150×4 matrix:

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

where an entry $x_j^{(i)}$ represents the j-th attribute of the i-th sample.

In particular, all the attributes of the i-th sample is represented by the i-th row:

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

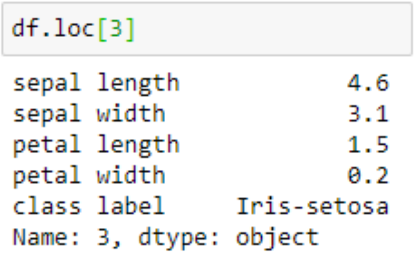
Also, the j-th attribute of all the samples is represented by the j-th column:

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

For our target variable (the class labels) will be stored in a single column:

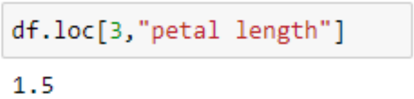
$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(150)} \end{bmatrix}$$

Recall the python data analytic techniques in a previous chapter. You can extract any entry, row or column from the DataFrame as shown in the example below.



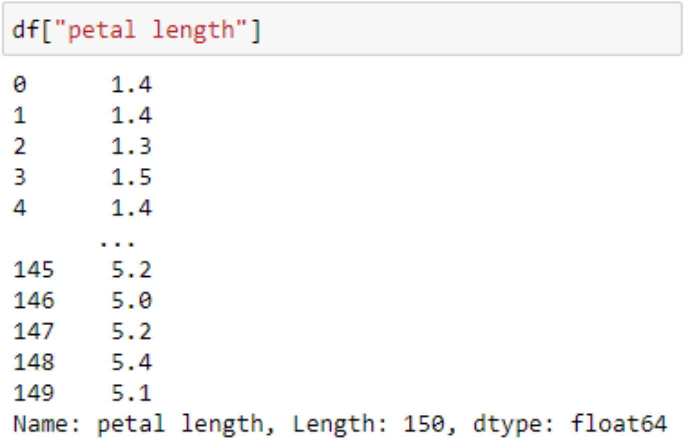
```
df.loc[3]
```

```
sepal length             4.6
sepal width              3.1
petal length             1.5
petal width              0.2
class label      Iris-setosa
Name: 3, dtype: object
```



```
df.loc[3,"petal length"]
```

```
1.5
```

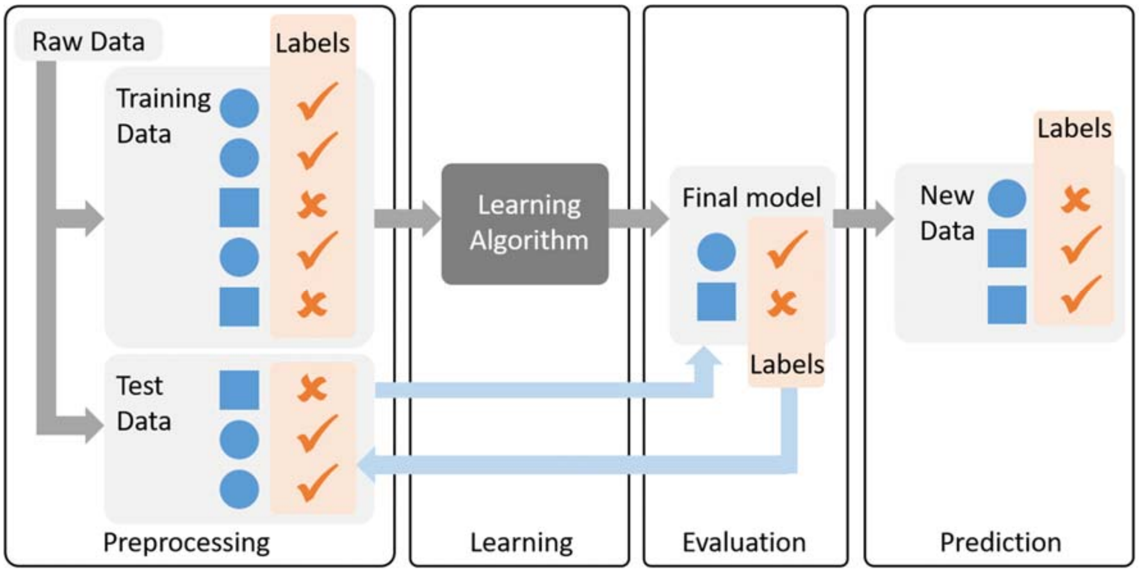


```
df["petal length"]
```

```
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: petal length, Length: 150, dtype: float64
```

5.3 Roadmap for building ML system

The Iris dataset is an example of labelled data. We will split it into a training set for training a machine learning model and a testing set for verifying the performance of our model. This will be introduced in the next chapter. Beforehand, we need to have a roadmap of such building a machine learning system, with a number of steps. The diagram below shows a typical workflow for using machine learning in predictive modelling which we are going to discuss in the remaining chapters. This can be concluded into four steps: preprocessing, learning, evaluation and prediction.



Preprocessing - getting data into shape

Raw data might come in different forms and shapes that is not suitable for optimal performance of a learning algorithm. Preprocessing refers to the step of extracting useful features from the raw data and to convert it into desired form. This usually include data cleaning, standardization, dimension reduction, noise reduction, etc. In many cases this step is crucial to make the algorithm more efficient and also improves predictive performance of the model.

To determine whether our machine learning algorithm not only performs well on the training dataset but also generalizes well to new data, we also want to randomly divide the dataset into a separate training and test dataset. We use the training dataset to train and optimize our machine learning model, while we keep the test dataset until the very end to evaluate the final model.

Learning - training and selecting a predictive model

There are many different machine learning algorithms developed to solve different types of problems. Each of these algorithm is based on some assumptions and therefore has its inherent biases. In practice, it is essential to compare the result of different algorithms in order to train and select the best performing model. For such comparison, we need to set up a metric to measure performance. One commonly used metric is classification accuracy.

In order to make sure that the performance of the model is not dependent to the selection of final test data, various techniques of cross validation is used. This refers to dividing a dataset into training and validation subsets in order to estimate the generalization performance of the model.

Evaluation - evaluating models

After we have selected a model that has been fitted on the training dataset, we can use the test dataset to estimate its performance and the generalization error. If we are satisfied with its performance, we can now accept this model.

Prediction - predicting unseen data instances

After all the previous steps, we come up with a model to predict new, future data. It is important to note that the parameters for the 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.