1 Learning Outcomes

- Explain how machine learning is different from traditional programming
- Explain the basic types of machine learning problems (e.g., regression, classification, clustering, and dimensionality reduction)
- List one example algorithm and/or model for each type of machine learning problem
- Explain what it means to fit a model
- Describe several use cases for machine learning
- Define the terms: features, samples, labels

2 What is machine learning?

Machine learning is a set of methods that allow computers to learn from data. It consists of the science and application of these methods, which transform data into knowledge and detect hidden patterns or structures directly from data. Many applications that we encounter in our everyday life are backed by machine learning: what Netflix suggests you to watch and what Amazon recommends you to buy (*recommender systems*), the auto-complete feature of an email app or search tab, chatbots (*natural language processing*), face detection and speech recognition, spam detection, and Google Maps' traffic prediction are all examples of machine learning applications.

The term *Machine Learning* was first coined in 1959 by the computer scientist Arthur Samuel, a pioneer in the field of artificial intelligence and computer gaming. In his 1959 paper [1], he hinted toward a universal explanation of machine learning, which was later popularly interpreted as the following:

> *"[Machine Learning is] the field of study that gives computers the ability to learn without being explicitly programmed."*

This definition highlights how machine learning is different than traditional programming. In traditional programming, a programmer manually derives the rules and writes the exact steps needed to perform a certain task to solve a given problem. In other words, after studying the problem on hand, a programmer writes a computer program that takes an input and transforms it into an output, in order to accomplish a certain task. On the other hand, in machine learning, a programmer might not know how to derive the rules or it might be complicated to come up with the rules for the given problem. Instead, the computer is given a dataset that consists of data inputs and their corresponding outputs, and the computer is then left to learn by itself the rules that maps the given data inputs to their corresponding outputs. We say that in In machine learning the computer self-learns a program that computes an output from a given input. The difference between traditional programming and machine learning is depicted in figures 1 and 2.

[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," in *IBM Journal of Research and Development*, vol. 44, no. 1.2, pp. 206-226, Jan. 2000, doi: 10.1147/rd.441.0206.

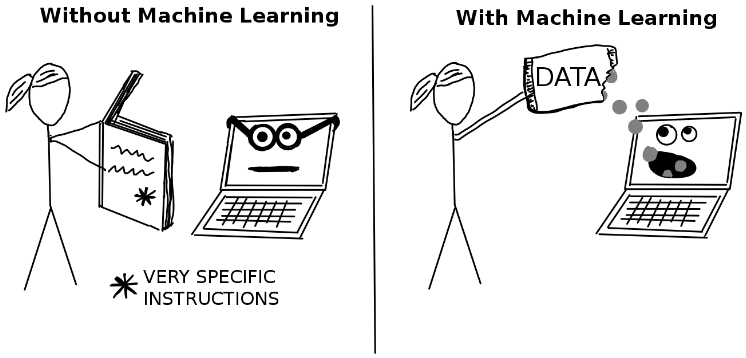

Figure 1: In traditional programming, instructions must be explicitly given to the computer. In machine learning, programming happens through providing data. Source: Interpretable Machine Learning

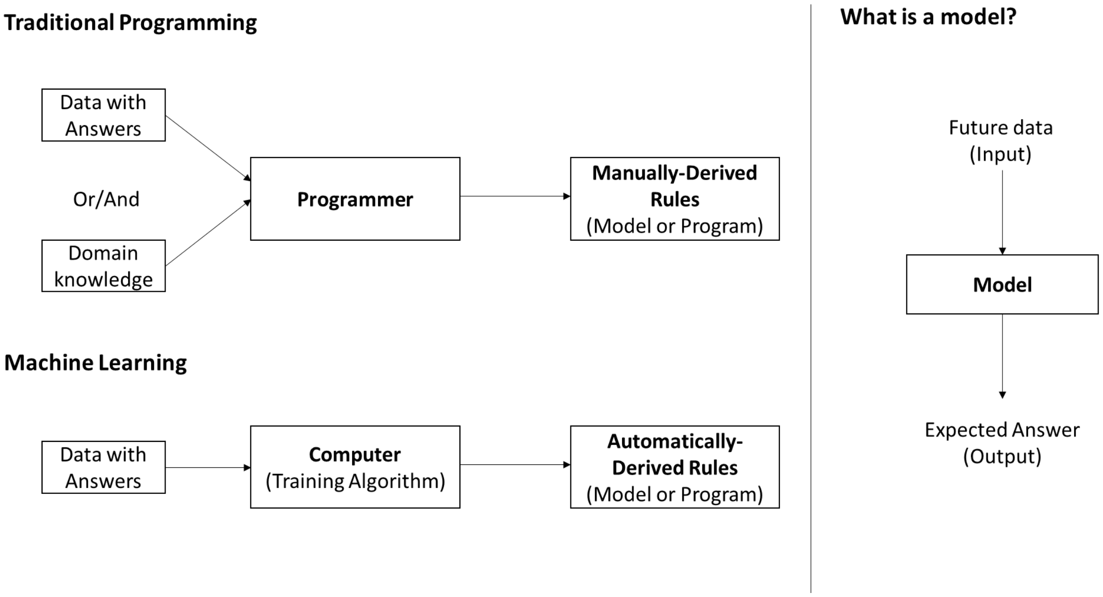

Figure 2: A model or a program can be seen as the rules that maps an input data to an output. In traditional programming, the model is manually derived by the programmer. In machine learning, the model is learned from the data.

Example 1 - Suppose you want to design a spam filter that classifies an email as spam or not-spam. In traditional programming, you need to spend some time trying to understand what a spam email looks like; for instance, you might try to examine the content of spam emails and look for any repeated words or catch any patterns. You then translate your observations into rules that define your spam filter program. It is easy to notice that this task is not trivial, can take some time and may not lead to accurate or consistent set of rules. In contrast, in machine learning, a computer is given a set of spam and not-spam emails and then uses machine learning techniques to automatically figure out frequent patterns that allow the detection of spam emails. In machine learning, the spam filter program is not explicitly written by hand; it is instead learned from data.

Example 2 - Suppose a bank wants to automate the process of evaluating credit card applications: approve or deny. The bank may not know any magical formula that can extract the necessary information from an application, in order to know whether to approve it or deny it. Instead of manually reviewing the applications, the bank can use historical data of credit applications and their corresponding outcomes (whether the applicant was a good or bad customer depending on their payment history) and relies on machine learning techniques to discover if there is a pattern between customers information and whether they are good or bad customers.

3 Types of machine learning problems

Machine learning problems can fit into different paradigms of learning depending on the desired task to perform. These paradigms can be divided into three different types of learning: supervised learning, unsupervised learning and reinforcement learning.

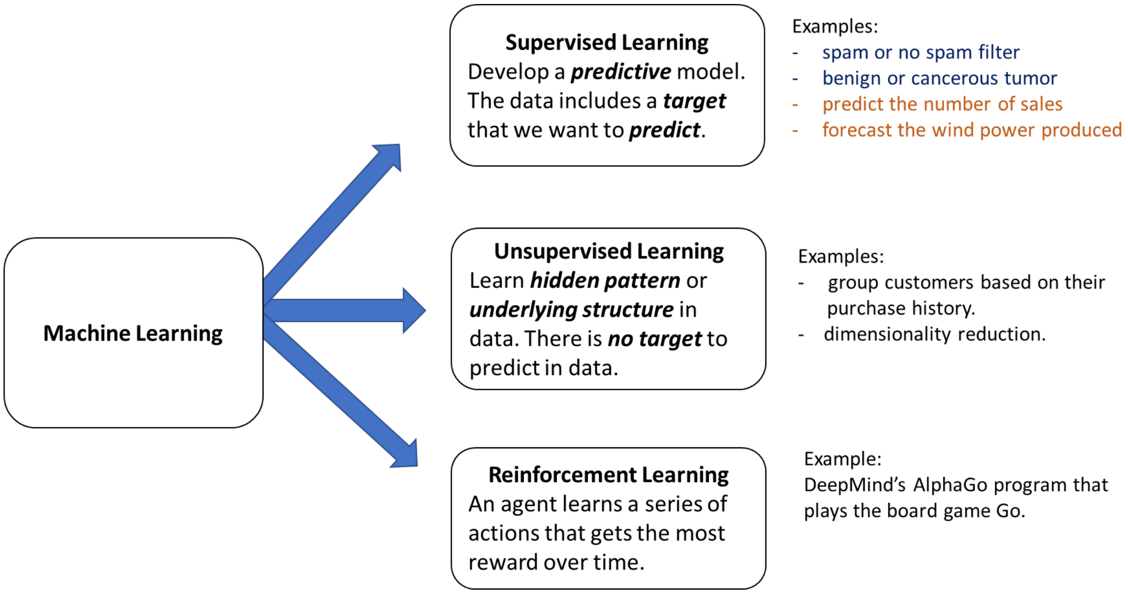
Machine Learning
Supervised Learning
Develop a ***predictive*** model. The data includes a ***target*** that we want to ***predict***.
Examples:
- spam or no spam filter
- benign or cancerous tumor
- predict the number of sales
- forecast the wind power produced
Unsupervised Learning
Learn ***hidden pattern*** or ***underlying structure*** in data. There is ***no target*** to predict in data.
Examples:
- group customers based on their purchase history.
- dimensionality reduction.
Reinforcement Learning
An agent learns a series of actions that gets the most reward over time.
Example:
DeepMind's AlphaGo program that plays the board game Go.

Figure 3: Types of machine learning problems

Supervised Learning: The main goal of supervised learning is to develop a program that makes **predictions** on future or unseen data: detect spam or not-spam email, detect benign or cancerous tumor, predict the number of sales, forecast the wind power produced. The historical data from which the machine learning program is intended to be learned must contain the target that needs to be predicted. For instance, in the example of spam filter, the computer is given not only each email content but also whether each email is spam or not-spam (the target). In the example of credit card approval, the bank has historical data of clients info and whether each client was a good or bad customer (was there any credit card default? How did they deal with it?). Using supervised machine learning techniques, the bank tries to learn from the given data a program that maps a client's info to whether it is a good or bad client, so that when future client applies for a credit card application, the bank uses the learned program to know whether to approve or decline the application. This type of learning is called supervised, because the presence of target in the data supervises or guides the learning process: we exactly know what we want to predict from the data.

Supervised learning can be further divided into two subcategories: regression and classification. In *regression*, the goal is to predict a continuous value (predict the number of sales, forecast the wind power produced).In *classification*, the goal is to predict a category (spam or not-spam email, benign or cancerous tumor). Binary classification refers to the case where there are two classes to predict, whereas multi-class classification refers to the case where there are three or more classes to predict.

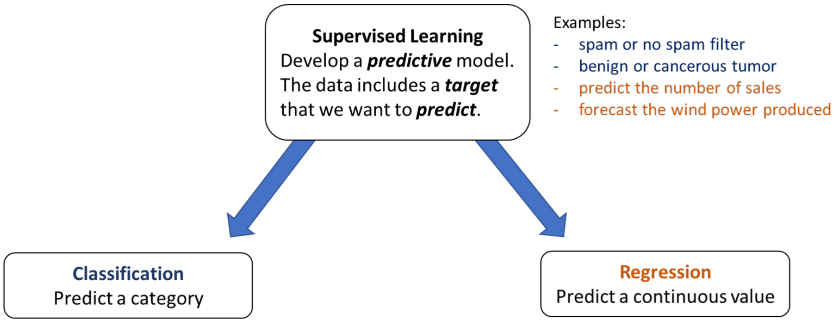
Supervised Learning
Develop a ***predictive*** model.
The data includes a ***target***
that we want to ***predict***.
Examples:
- spam or no spam filter
- benign or cancerous tumor
- predict the number of sales
- forecast the wind power produced
Classification
Predict a category
Regression
Predict a continuous value

Figure 4: Types of supervised machine learning problems.

Unsupervised Learning: In unsupervised learning, there's no target to predict in the data. The goal is to instead learn hidden patterns or underlying structure in data. One example of unsupervised learning is data clustering. For instance, suppose you have information about online shoppers and their purchase history. Your goal is to group the shoppers into groups based on the information given so that shoppers with similar taste end up in the same group. There is no information about how many groups exist in the data and to which group each shopper belongs. This is exactly what you wish to discover using clustering techniques. Another example of unsupervised learning is dimensionality reduction, which aims at transforming a high dimensional data (data with lots of measures) into lower dimensional data without loosing too much information. Simplifying data might be helpful for visualization purposes or when there is a limitation with space or computational resources.

Reinforcement Learning: In reinforcement learning, a software program (called agent) observes a certain environment, takes some actions and received rewards based on its actions. The goal is to learn a series of actions that maximizes its expectd long-term rewards. Reinforcement learning has interesting applications in games (AlphaGo program) and machine control.

4 Terminology

We have discussed in the previous sections the three types of learning - supervised learning, unsupervised learning and reinforcement learning. In this course, we will mainly focus on supervised learning. Let us now introduce the basic terminology that we will be using. Consider the following classification scenario:

Suppose you want to predict the presence of heart disease in patients. You have a set of labeled patients data. For each patient, you know the following: age, resting blood pressure, cholesterol level, fasting blood sugar and maximum heart rate achieved and you also know if they have heart disease. Using this historical data, you want to come up with a program that maps the patient's information to whether they have heart disease or not.

Feature: each measurement in the dataset that is part of each patient's info (age or cholesterol level or resting blood pressure, ...) is called a feature, which can be equivalently designated as **attribute, predictor, or independent variable.**

Label: it represents the target or outcome that needs to be predicted in supervised learning. In the above classification scenario, it is the "presence of heart disease". In regression scenario, the continuous target is also called a label. A label can be equivalently designated as **response, outcome, target or dependent variable**.

Sample: the set or collection of features that designates one patient is called a sample. It can be also called **observation, instance, record or example**.

Example: Consider now the following regression scenario:
Suppose you want to predict the sale price of houses in a given area. You have a set of labeled houses data. For each house, you know the following: year, number of bedrooms, area in square foot, number of bathrooms, zip code. What are the features, labels and samples of this regression scenario?

Example: The Iris dataset is a classic example in machine learning.
The dataset consists of measurements in centimeters of the sepal length and width, and the petal length and width, respectively, for 150 flowers from three species (setosa, versicolor, and verginica) of iris plants. What are the features, labels and samples of this classification scenario?

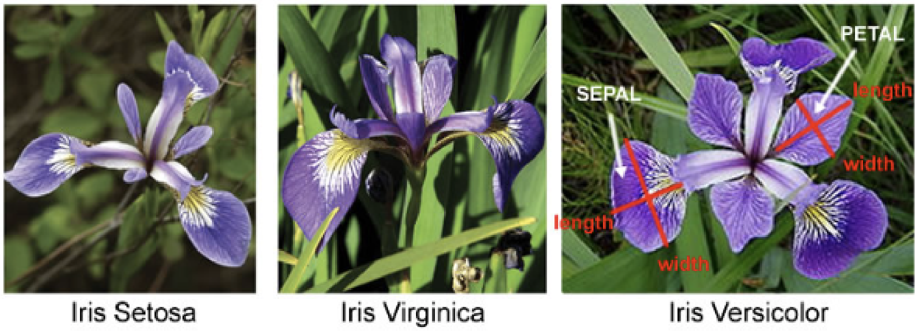

Figure 5: The three species of the Iris flower. Image Source

5 Roadmap of Machine learning System

Now that we have a general understanding of what is machine learning, its different paradigms and the corresponding terminology, let us now explain the different phases of building a machine learning system.

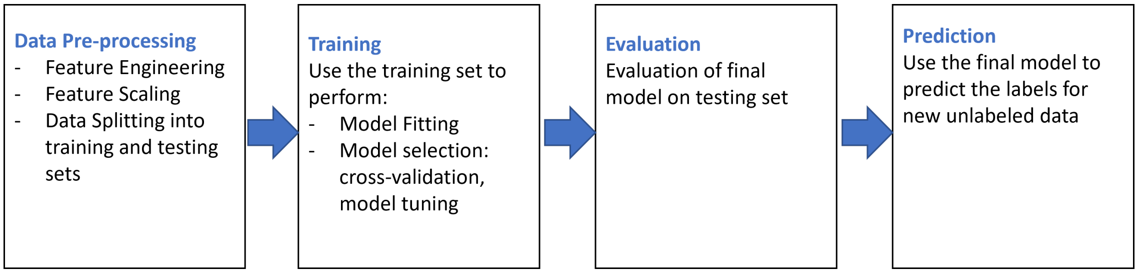
Data Pre-processing
- Feature Engineering
- Feature Scaling
- Data Splitting into training and testing sets
Training
Use the training set to perform:
- Model Fitting
- Model selection: cross-validation, model tuning
Evaluation
Evaluation of final model on testing set
Prediction
Use the final model to predict the labels for new unlabeled data

Figure 6: Phases involved in building a machine learning system.

Data Pre-Processing: This step makes sure that the data is in a form that is suitable for the subsequent steps. In supervised learning, the data is expected to come as a set of pairs of data samples and labels, where each data sample is a collection of features. Sometimes the data does not come in this ready-form and what is given instead is the raw data. In this case, some effort must be put in, in order to extract relevant features from the raw data (feature engineering). Another important pre-processing step (that might be required for certain machine learning systems) is to scale the features so that all features end-up having similar range of values. We will explain later in the course why and when feature scaling is important. Another essential pre-processing step is dividing the data into training and testing sets, which will be explained why in the sequel. The training set should be used in the training phase and the testing set should be used in the evaluation phase as shown in figure 6.

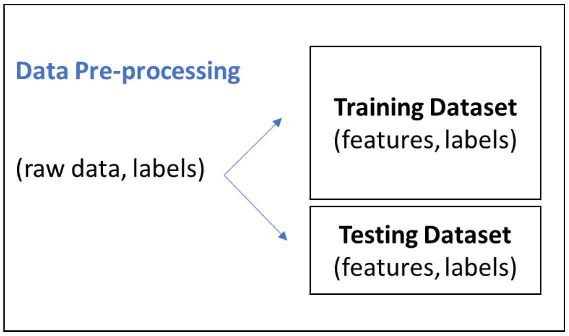

Figure 7: In the data-preprocessing steps, features are extracted from raw data. The pairs of features and labels are then split into training and testing sets.

Training: After we make sure that the data is in a format that is suitable for learning, the data is now ready for the training phase. The goal of the training phase is to learn, from the labeled data (pairs of data features and labels), the rules or the mathematical formula that maps an input feature to its output label. When trying to find the rules that are specific to the data-on-hand, we don't start from zero. We make assumptions of how the data features are related to the labels. These assumptions are captured by what is known as the machine learning models (they model the relationship between the data features and the labels). Many machine learning models were proposed where each makes different assumptions:

Example of regression models: Linear Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forests, Neural Networks.

Example of classification models: Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Random Forests, Support Vector Machines (SVMs), Naive Bayes, Neural Networks.

Depending on the problem, a model is picked and fit (or trained) on the training dataset. Fitting a model means finding the characteristics of the model; most of the machine learning models have certain characteristics (parameters) that need to be specified in order to be used. Since multiple possibilities exist for these characteristics, fitting the model on the training data means finding the best model's characteristics that best approximates the relationship between the data features and the labels, i.e., the training problem is set up as an optimization problem. The training algorithm mentioned in figure 8 is the algorithm that tries to find the best model's characteristics.

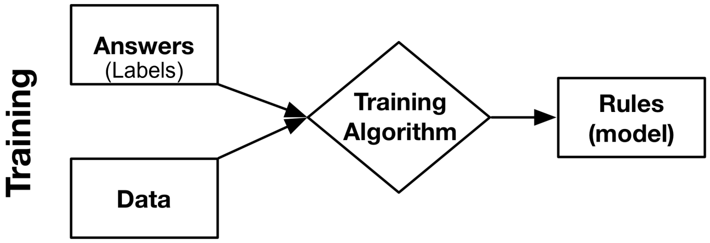

Figure 8: Model fitting

Let's illustrate the concept of model fitting with two examples. A linear regression model assumes that there is a linear relationship between the data features and the continuous target. Geometrically speaking, if the data consists of only one feature, a linear regression model assumes that the relationship between the feature and the target can be approximated with a line. A line is defined by its slope and y-intercept (model's characteristics); fitting a linear regression model means finding the best pair of slope and y-intercept that best approximates the relationship between the feature and the target. In logistic regression (binary classification model), if the data consists of two features, fitting a logistic regression model geometrically means finding the equation of the line that best separates the two classes of the data.

Each machine learning model has its strengths and weaknesses, and we might not know beforehand the most suitable model to choose for the given problem. In this case, we might train more than

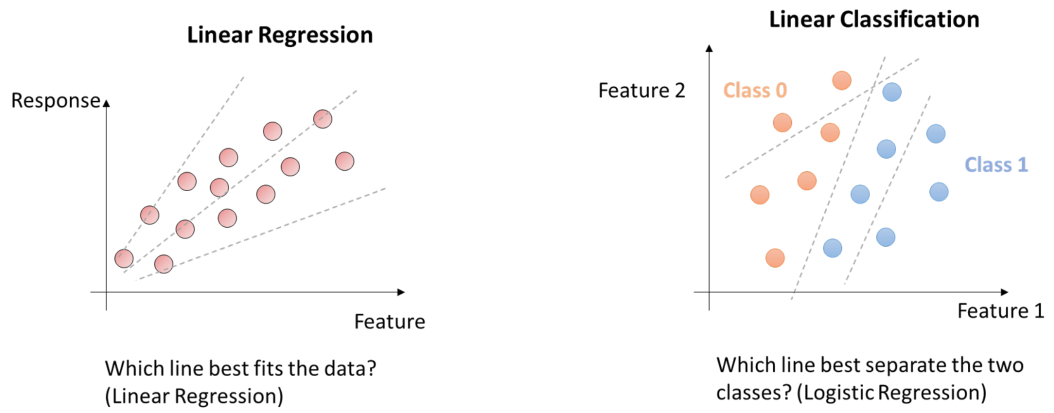

Figure 9: Fitting linear models

one model on the training data and then choose the best performing model as the final model. This process is known as ***model selection*** and it involves the use of methods like cross-validation and hyperparameter tuning.

Evaluation: Once a final model is chosen, the model is evaluated on the testing set. This is done by applying the final model on each sample in the testing set, in order to compute the predicted label for the given sample. The predicted labels are then compared against the actual labels in the testing set. The goal of the evaluation phase is to understand how the model will generalize to unseen data, i.e. how well the model would perform in real-life on future data once it is deployed. To quantify the amount of error made by the final model, we use metrics that measure how close the predictions are to the actual labels. The choice of the evaluation metric depends on: the type of machine learning problem (e.g., classification or regression) and whether some types of errors are more important than others.

Prediction on new unlabeled data: If we are satisfied with the final's model performance, we can use the model on new unlabeled data to predict their corresponding label. The final model needs to be monitored, maintained and updated (re-trained on updated datasets or re-trained due to observed drift in data).

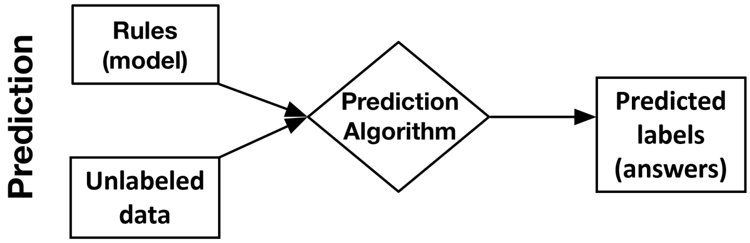

Figure 10: Model Prediction

6 Model errors

When training machine learning models, we might observe that some models might perform worse than some other models. This can be because of many reasons:

- Option 1: Models are not predictive at all

  This could happen if the data used is not informative due to errors in measurements, lots of missing entries, or presence of irrelevant features (garbage in, garbage out). This could also happen if the data does not contain enough features or the trained models are too simple.

- Option 2: Models "overfit" the training set

  Models learn rules that enable perfect predictions on the training set. Overfitting happens when these rules aren't applicable to any other data points. This might happen when we don't have enough samples or when use a too complex model for the data or when the available dataset is not representative enough of the population (biases exist in data).

7 Course outline

In these notes, we familiarized ourselves with the big picture of machine learning. In this course, we will cover the following main topics:

- the details of some basic machine learning models: their assumptions, advatanges and limitations;
- the best practices in evaluating machine learning models and the evaluation metrics that can be used;
- the use of optimization in training machine learning models.

8 Further Readings

- Chapter 1 of the book "Hands-On Machine Learning with Scikit-Learn and TensorFlow"
- Chapter 1 of the book "Python machine learning" by Sebastian Raschka
- Chapter 2.1 (2.1.1, 2.1.2, 2.1.4, 2.1.5) of the book "Introduction to statistical learning"