Practical work 05 – 18.10.22
Supervised learning – System Design and Debugging

Summary for the organisation :

- Submit the solutions of the practical work before Monday 12h00 next week in Moodle.
- **Rule 1**. Submit an archive (***.zip**!) with your Python notebooks (one per exercise), including datasets and all necessary files.
- **Rule 2**. The archive file name must contain the number of the practical work, followed by the family names of the team members by alphabetical order, for example `02_dupont_muller_smith.zip`. Put also the name of the team members in the body of the notebook (in first cell). Only one submission per team.
- **Rule 3**. We give a **fail** for submissions that do not compile (missing files are a common source of errors...). So, make sure that your whole notebooks give the expected solutions by clearing all cells and running them all before submitting.

Exercice 1 Data preparation - UBS use case

The bank UBS is offering to its client the possibility to invest money in funds[1]. See https://fundgate.ubs.com/. There are thousands of investment funds available. Clients, according to their profile, will be more or less inclined to invest in a given fund, according to the fund characteristics. For example, a younger client with no child is potentially more interested into funds composed with stocks, showing higher risks but also higher potential returns. A family father will be more inclined to invest into low-risk funds. UBS want to build a system as illustrated on Figure 1, taking as input a set of values characterizing the fund and a set of values defining the client profile.

An investment fund can be characterized by the following elements :

- The name of the fund.
- The current value of 1 share in the fund, expressed in CHF.

1. An investment fund is composed of financial values such as stocks or bonds and sometimes other instruments. For example, a fund composed mostly of stocks has more return potential but is more risky in case of stock market recession.

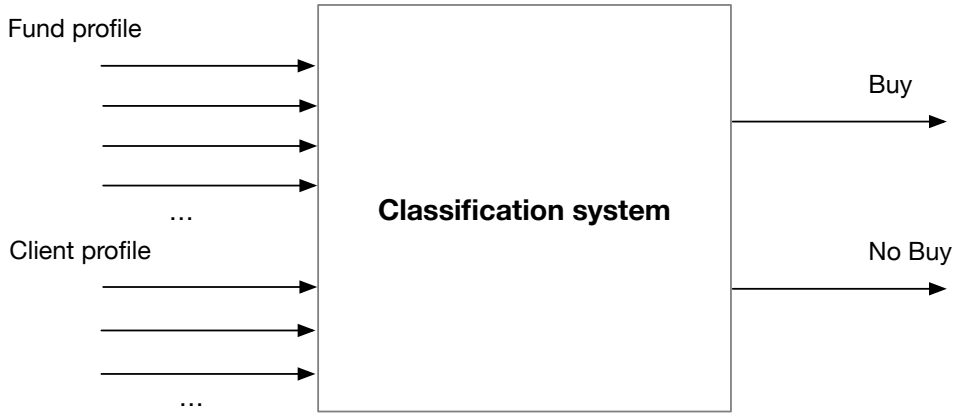

FIGURE 1 – Classification system to predict if a fund is likely to be bought by a client.

— The proportion of stock and bonds composing the fund (2 values in percentage).
— A vector of float values with the 5 last yearly returns over years from 2015 to 2019 (5 values expressed in percentage).
— A level of risk expressed with A, B, C, D, E with A representing the highest risk and E representing the lowest risk level.
— A sectorial information such as *technology*, *pharmaceutical*, *financial*. There are 24 different sectors available in UBS funds.

A client profile contains the following information :

— First name and last name of the client.
— The mother tongue of the client (mostly de, fr, it and en but other languages are present).
— The age of the client.
— The number of children of the client.
— The current wealth of the client that could be used to buy funds, expressed in CHF (total of cash available in the different accounts, not yet invested in funds).
— The postal code of the address of the client.
— A level of acceptance to risk expressed with A, B, C, D, E with A representing the highest level of acceptance of risk and E representing the lowest acceptance of risk.

Answer the following questions :

a) For each available information in the fund and client profile, explain how you would prepare the data : encoding, normalization, outlier treatment, etc.

b) How could you collect targets (output of the system) to train the system ? How would you prepare the different sets ?

Be as comprehensive as possible. Imagine that you give your analysis to your trainee : he must be able to implement the system from it.

Exercice 2 Debugging drugs dataset

The dataset contains a set of patients, all of whom suffered from the same illness. During their course of treatment, each patient responded to one of 5 medications : Drug A, Drug B, Drug c, Drug x or y. *(Source : kaggle. com/ ammaraahmad/ top-10-machine-learning-datasets)*

This complex dataset proposes a set of challenges that you'll try to overcome. A basic ML pipeline is already in place. You have to optimize the performance of the model by applying good practices, debugging pre-processing errors, etc.

Rules :

- Do not use other modules than those already imported (or do it only if your own code is not working. In this case, keep your own code in the notebook, commented, so that we know what you tried).
- Explain **all** of your choices. For every task, choose the most appropriate option for this problem and describe your choice.
- You can modify any parts of the code or replace the model by one already used in previous PWs.

Work to do :

- Apply a type of normalization.
- Encode categorical data.
- Use all columns in the dataset (or choose the most meaningful features).
- Choose a more appropriate metric.
- Optimize hyper-parameters.
- Test the model performance correctly using a separated test set.
- Apply **two** of those techniques :
  - Keep relative class frequencies in the train/test sets (check train_test_split docs)
  - Show which feature(s) are the most correlated to the target.
  - Use cross-validation.

Exercice 3 Linear regression optimisation

We would like you to find the order O of the polynomial function that models the best the data illustrated in Figure 2.

$$h_\theta(\mathbf{x}) = \theta_0 + \theta_1 x + +\theta_2 x^2 + \ldots + \theta_D x^O \tag{1}$$

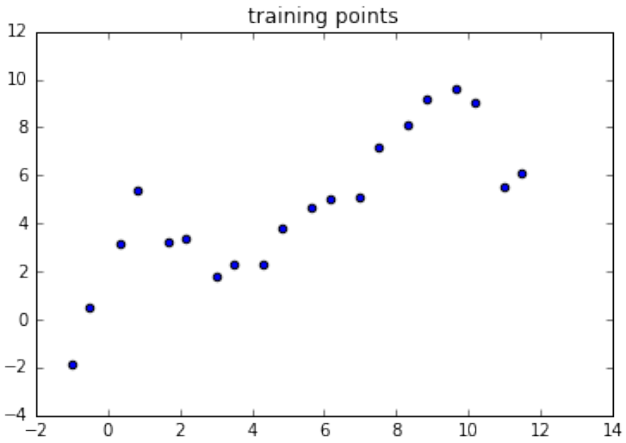

FIGURE 2 – Training data for linear regression optimization

The data set is split into a training set and a test set. You will find the data in the files `overfitting_train.csv` and `overfitting_test.csv`. The data has been split for you into a training set and a test set.

a) Read the data in separate variables for the training and test sets.

b) Define a cost function $J(\theta)$ that will allow you to compute the cost on the training and test sets.

$$J_{train}(\theta) = \frac{1}{2N_{train}} \sum_{n=1}^{N_{train}} (h_\theta(\mathbf{x}_n^{train}) - y_n^{train})^2 \tag{2}$$

$$J_{cv}(\theta) = \frac{1}{2N_{cv}} \sum_{n=1}^{N_{cv}} (h_\theta(\mathbf{x}_n^{cv}) - y_n^{cv})^2 \tag{3}$$

c) Perform the training for increasing orders $O = 1, \ldots, 10$. You can use any training method developed in the previous practical work PW4 (using the *normal* equations is probably the easiest choice (see http://mlwiki.org/index.php/Normal_Equation)).

d) Plot the trained hypothesis. You should have something similar to the next figure.

e) Plot the evolution of the costs $J_{train}(\theta)$ and $J_{cv}(\theta)$ as a function of the order O.

f) What is your best model according to these costs ? Comment your answer.

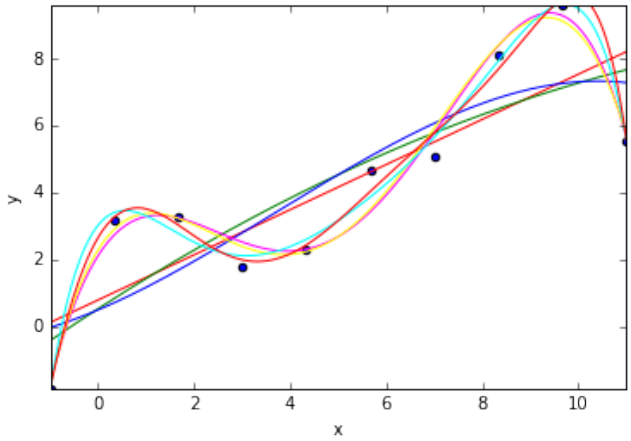

FIGURE 3 – Trained models for $D = 1, \ldots, 7$.

g) Would you still choose the model with the lowest cost for production ?

h) Explain how under/over-fitting is involved here.