Practical work 02 – 23rd of September 2025
KNN for Supervised Classification and Regression

Summary for the organisation :

- The practical works (PW) are optional. If you submit a PW, we provide **feedbacks** and you get potential **bonuses** on the final grade as explained in the first week. Submissions should be done before the date specified in Moodle.
- **Rule 1**. Submit 1 archive (***.zip**) with your Python notebooks. Do not include datasets unless specific instructions, but do include all necessary files to reproduce your experiments.
- **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 don't give bonuses 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 `Numpy` tutorial

This exercise is to get you more familiar with `numpy`. Read the content of the ipython notebook `numpy-tutorial-stud.ipynb` that you will find on Moodle. Pay a special attention to the *broadcasting* section that allows to gain significant speedup when processing large numpy arrays. Regarding this, it is usually more efficient to use *broadcasting* instead of for loops in your code. At the end of the tutorial, you have to complete some manipulations of images stored by arrays.

Exercice 2 Classification system with KNN - To Loan or Not To Loan

The objective of this exercise is to build a classification system to predict whether a loan is approved based on the client status. A dataset of applicants is provided. For each sample $\mathbf{x}_n$, you have several features such as income $x_{n,1}$ and loan amount $x_{n,2}$. The approval decision y_n is also provided for each sample. See the notebook for a description of these features.

a. Getting started

Use the supplied notebook as a basis for the following tasks. Some codes like data loading and data preprocessing are given to help you. Do not use the functions of the sklearn package except those present in the notebook.

Remark : The provided data preprocessing do the following tasks :

- Encode categorical values using an ordinal encoding.
- Apply scaling to normalize the features.
- Apply some type conversion.
- Split the data in training (80%) and test (20%) sets.

b. Dummy classifier

a) Build a dummy classifier that takes decisions randomly.

b) Implement a function to evaluate the performance of a classification by computing the accuracy ($N_{correct}/N$).

c) Compute the performance of the dummy classifier using the provided test set.

c. K-Nearest Neighbors classifier

a) Build a K-Nearest Neighbors (KNN) classifier on the data using an Euclidian distance computation and a simple majority voting criterion, e.g. decide C_0 when there is a majority of points in class 0 in the k nearest neighbors. Code the algorithm by yourself !

b) Compute the performance of the system as a function of $k = 1 \ldots 7$. What value of k gives you the best performances ? Comment your result.

c) Run the KNN algorithm with $k = 3$ using only the features `TotalIncome` and `CreditHistory`. Report its performance on the test set.

d) Re-run the KNN algorithm by using following features : `TotalIncome`, `CreditHistory` and `Married`. Report its performance on the test set.

e) Do it again by using all the features available. Compare the performance obtained when using different number of features. What do you notice ? What can you tell about the relationship between the performance, the number of samples and the number of features when using this algorithm ?

f) How is your system taking decisions when you have an equal number of votes for both classes with values of $k = 2, 4, 6$?

Exercice 3 Classification system with KNN - MNIST dataset

It is now time to move to larger datasets and more intensive tasks. We will use the MNIST database that contains images of handwritten digits. This page offers a description of the dataset : http://yann.lecun.com/exdb/mnist/. It has a training set of 60,000 examples, and a test set of 10,000 examples. It is actually a subset of a larger set available from NIST [1]. In MNIST, the digits have been size-normalized and centered in a fixed-size image, as depicted in Fig 1.

a) Download the dataset `mnist.zip` from Moodle and expand the archive.
b) Download the notebook file `knn-mnist-stud.ipynb` from Moodle.
c) Follow the steps explained in the notebook.

You need to hand in the modified Python notebook with inline answers to questions and code completed wherever there is a `TODO` indication.

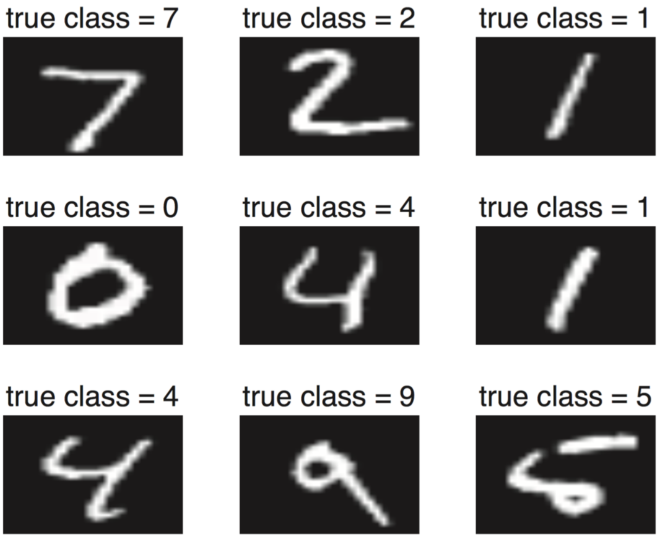

FIGURE 1 – MNIST dataset examples.

[1] National Institute for Standards and Technology - USA

Exercice 4 Optional – Classification with KNN - Fashion MNIST and CIFAR10

a) Redo the experiments of exercise 3 using the Fashion MNIST dataset from Zalando : https://github.com/zalandoresearch/fashion-mnist. Report your best performances and compare it to the performances obtained on digit MNIST.

b) Redo the experiments of exercise 3 using the CIFAR10 dataset : https://www.cs.toronto.edu/~kriz/cifar.html. Report your best performances and compare it to the performances obtained on Fashion MNIST and digit MNIST.

c) Optional difficult : using pixel based L1 distance is a bit of a naive approach. Attempt to perform more advanced feature extraction on the images such as projection histogram or Linear Binary pattern to increase your performance on Fashion MNIST.

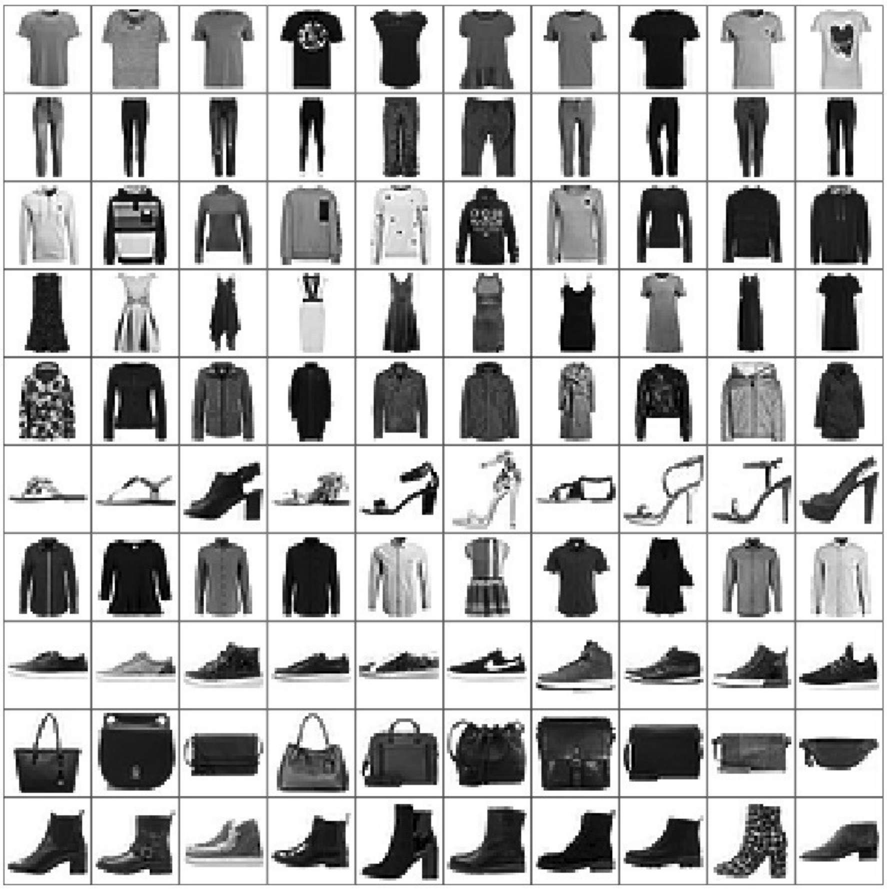

FIGURE 2 – Fashion MNIST dataset examples.

Please provide the answer to this question in a separate Python notebook.

Exercice 5 Regression with KNN

The objective of the exercise is to experiment with a KNN approach used in regression mode.

a) Read again the regressor KNN in the theory of this week's class.
b) Download the `lifesat.csv` file from moodle. The data represents the level of life satisfaction as a function of the GDP per capita (2020 data set, taken from `https://ourworldindata.org/grapher/gdp-vs-happiness`).
c) Start from the `regression-knn-stud` and complete the code to perform a regression with a KNN approach.
d) Make sure you can reproduce the figures presented in the class with, e.g. K=3 or K=7.

Questions to address in the report :

— What is the predicted life satisfaction for Cyprus assuming that the GDP per capita is 38,341 USD ?
— What is the predicted life satisfaction for Switzerland assuming that the GDP per capita is 69,669 USD ?
— What becomes the prediction when K is approaching N (the number of points in the training set). Do the experiment with $K = 20$ and $K = 27$, report your observation.

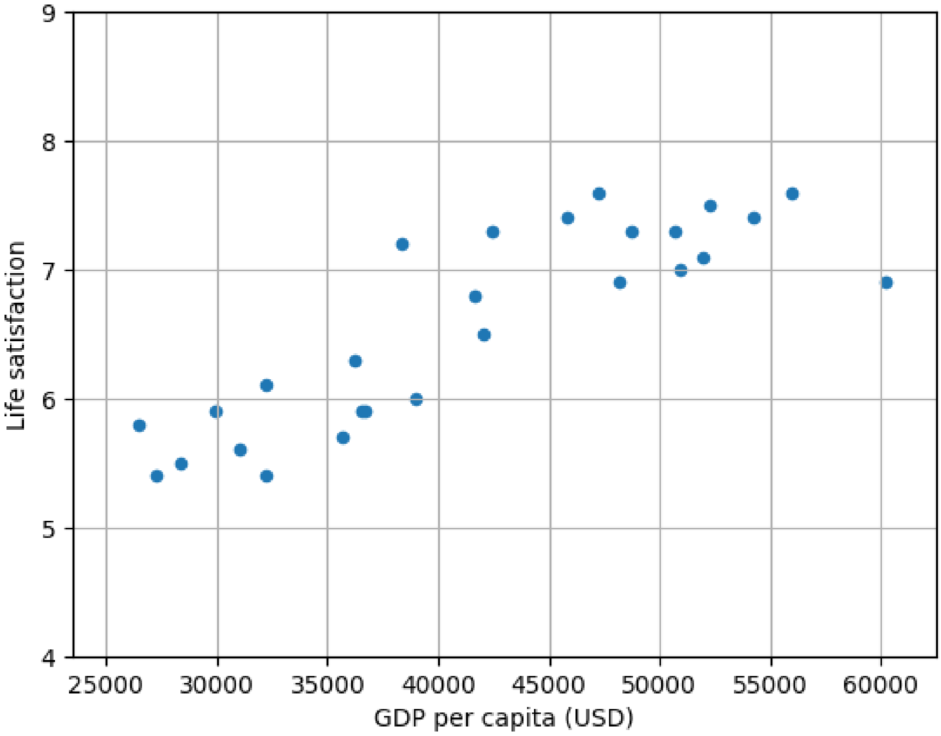

FIGURE 3 – Life satisfaction index as a function of the money a country makes (GDP).

Please provide the answer to this question in a separate Python notebook.

Exercice 6 Optional – Review questions

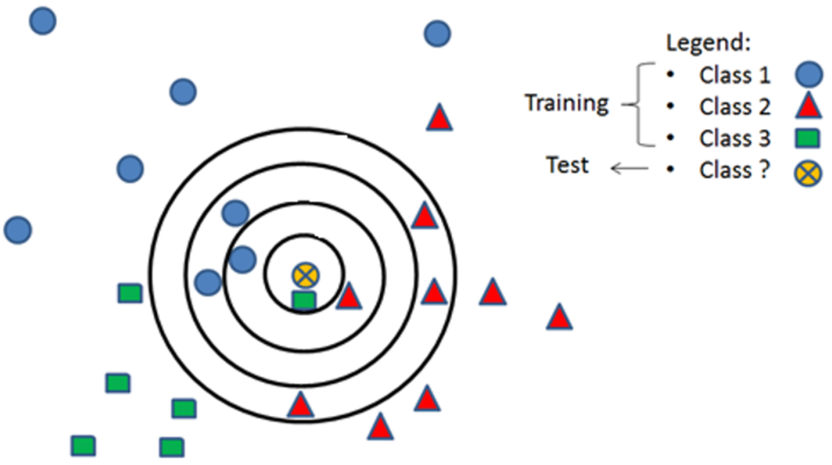

Figure 4 – K-NN Classification

a) The figure above illustrates the K-nearest neighbors (K-NN) algorithm using Euclidean distance, represented by circles. Based on a majority voting and shortest distance (when tie), which class would the “Test” instance be assigned to with $K = 2$, $K = 5$, $K = 7$, $K = 8$?

b) Explain in your own words the differences between instance-based learning and model-based learning.

c) Is K-NN an instance based learning or a model-based learning ?

d) When are larger K beneficial ?

e) Why are too large K detrimental ?

f) When used in classification mode, what can we do when the first categories have equal number of votes with a K-NN ?

g) Are K-NN algorithms good candidates to build a 1’000 classes image classification system ? Explain your answer.

h) Is K-NN impacted by the "curse of conditionality" ? Explain your answer.

i) **Difficult** What is the expected error rate computed on the training set with $K = 1$?

j) **Difficult** What is the expected error rate computed on the training set with $K = 2$ and a shortest distance based tie resolution ?