ML I supervised learning : 2024 - 2025 project

NICOLAS LE HIR
nicolas.le-hir@epitech.eu

TABLE DES MATIÈRES

INTRODUCTION

You can push your work to the epitech repo dedicated to the project, inside the course team. Please do not send me a fork of the course repo, with some added files. Instead, make a repo containing only your project files.

For questions related to group inscriptions to the project, please contact the administration directly via a ticket.

All processing should be made with python3.

Some form of report must accompany your code, in order to explain and comment it. General explanations and **conclusions** on your global approach are expected, instead of low level explanations on elementary functions. Please write some sumary of your conclusions at the **beginning** of the report (or notebook) of each exercise (this actually facilitates reading a lot). For instance :

— (For exercice 3) : "we managed to obtain a test accuracry of 0.88 with the (input model A name) model, whereas with (input model B name) we could not obtain more than 0.75."

— Example of explanations to include in your report : discussion over the choice of models, hyperparameters, and conclusion on which model(s) worked best, or over the choice of preprocessing methods on the datasets.

— Example of explanations **not** to include in your report : presentation of elementary python function, or library functions from numpy, matplotlib, or the libraries themselves. You should **not** present these.

The preferred format is using one notebook per exercise, with markdown comments explaining you approach. If you use python scripts, you may also write a pdf report. Short docstring at the top of files and functions will be appreciated, if

relevant. There is no length constraint on the report, you do not need to write more than necessary. The goal of writing a report is to help me give you useful feedback.

General, important guidelines :
- when you mention the score of an estimator, **always** explicitely mention whether it is a train, test, validation, or cross-validation score (see the "scoring" chapter in the class)
- never forget to label the axes of plots
- never forget to mention the dimensions (unités) of the quantities

The 5 parts of the project are independent.

1 PART 1 : ARTIFICIAL DATASET GENERATION

The goal of this exercise is to work with statistical notions such as mean, standard deviation, and correlation.

Write a file or a notebook that generates a numerical dataset with 300 datapoints (i.e. lines) and at least 6 columns and saves it to a csv file named **artificial_dataset.csv**.

The columns must satisfy the following requirements :
- they must all have a different mean
- they must all have a different standard deviation (English for "écart type")
- at least one column should contain integers.
- at least one column should contain floats.
- one column must have a mean close to 2.5.
- some columns must be positively correlated (a pair of column must have a correlation > 0.2).
- some columns must be negatively correlated (a pair of column must have a correlation < -0.4).
- some columns must have a correlation close to 0.

2 PART 2 : DEFINITION OF A METRIC

A dataset representing a population is stored in **dataset.csv** inside the **project/ex_2_metric/** folder. The features are :
- age in years
- height in centimeters
- job
- city
- favorite music style

Define two different **metrics** in this dataset, which means define two methods to compute **dissimilarities** between the samples, by taking into account all their features (columns of the dataset). The objective is that :
- the two samples that are the closest in the dataset are different according to metric 1 and to metric 2.

— the two samples that are the most far appart in the dataset are different according to metric 1 and to metric 2.

Some features are numerical and others are categorical, hence you can not use a standard euclidean metric, and you need to define custom metrics, like we did in the **code/metrics/hybrid_data/** exercise during the course. Compute the mean dissimilarity and the standard deviation of the dissimilarity distribution that you obtain for each metric. The units of measurement (unités, like Kg, cm) should be taken into account while computing the metrics. Compute explicitely the most similar and most dissimilar samples for each metric and discuss the result by commenting on the balance of the features in each metric.

3 PART 3 : PREDICTION OF THE WINNER OF A NBA GAME (CLASSIFICATION)

We would like to predict the winner of a Basketball game, as a function of the data gathered at half-time.

The dataset is stored in **project/ex_3_classification_NBA/** :

— The train and test inputs representing the features are stored in **X_train.npy** and **X_test.npy** respectively.

— If the home team wins, the label is 1, −1 otherwise. The train and test labels are stored in **y_train.npy** and **y_test.npy** respectively.

Your objective is to obtain a mean accuracy superior than 0.84 on the test set.
https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score

Remark : Pay attention to the fact that **the test must not be used for training. The test set should be used only once, even for scoring**. If you compute the score several times, with different models on the test set, it means that you use it more than once, even if you do not call a scikit model.train() method on the test set (see the class for more explanations) ! Note that this strict unique usage of the test set is not always common practice in companies, but try to apply it for this exercise.

You are free to choose the classification methods, but you must compare at least 2 models. You can do more than 2 but this is not mandatory for this exercise. Discuss the choice of the optimization procedures, solvers, hyperparameters, cross-validation, etc. It is sufficient that 1 of your models reaches the objective score.

Several methods might work, including some methods that we have not explicitely studied in the class, do not hesitate to try them.

Indication : a solution, with the correct hyperparameters, exists in scikit among the following scikit classes :

— linear_model.LogisticRegression

— svm.SVC

— neighbors.KNeighborsClassifier

— neural_network.MLPClassifier

— ensemble.AdaBoostClassifier.

4 PART 4 : PREDICTION OF THE AMOUNT OF ELECTRICITY PRODUCED (REGRESSION)

We would like to predict the amount of electricity produced by a windfarm, as a function of the information gathered in a number of physical sensors (e.g. speed of the wind, temperature, ...).

The dataset is stored in **project/ex_4 _regression_windfarm/**, similarly to 3 and the instructions are the same (compare at least two models and discuss the various settings). Your objective is to obtain a R2 score superior to 0.85 on the test set, for at least 1 of your models.

The same remark about the test set, presented in exercice 3 also applies here.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html

Several methods might work, including some methods that we have not explicitely studied in the class. Do not hesitate to try such methods.

Indication : a solution, with the correct hyperparameters, exists in scikit among the following scikit classes :

— linear_model.Ridge

— linear_model.Lasso

— neural_network.MLPRegressor.

— svm.SVR, ensemble.AdaBoostRegressor.

5 PART 5 : APPLICATION OF SUPERVISED LEARNING

Pick a dataset of your choice and perform a classification or a regression on it, in order to study or solve a problem (of your choice as well).

Mandatory (very important) : before any processing, you must fulfil these two steps :

— present the dataset shortly in your own words (please do not copy a description from another resource) and link to the url where you downloaded it from. For instance, it is very important to **present the features of this dataset that are not obvious** for someone (me) that is not necessary familiar with the dataset.

— **explain very explicitely what problem you are trying to solve, and in particular what quantity you are trying to predict, as a function of which features**. If relevant, discuss why solving this problem would be interesting or have a value for an industry.

You are encouraged to compare several estimators / optimization procedures, from different points of view (scoring, computation time, etc). **General guideline :** this course is dedicated to discovering and exploring some of the many principles of machine learning, rather than being a production-oriented course. Hence, you are encouraged to explore original and personal approaches. It is not a huge deal is the final scores are not outstanding, as long as you took the chance to explore a custom approach and learned a new possible method.

Suggestion of steps :

— provide general analysis of the dataset, that studies its statistical properties, outliers, correlation matrices, or any other interesting analysis. You may produce visualizations.

— if relevant or necessary, preprocess the data, and to justify this preprocessing. You could compare the estimator(s) obtained with and without preprocessing.

— discuss the relevant optimization details : cross validation, hyperparameters, etc
— (**mandatory**) provide an **evaluation** or multiple evaluations of the obtained estimator(s), thanks to scorings of your choice.
— (**mandatory**) discuss the results obtained. Have we solved a problem with this processing ?

Some resources to find datasets : Link 1, Link 2, Link 4. If necessary, you can even tweak a dataset in order to artificially make it possible to apply analysis ans visualization techniques that you like, or downsample it. This is not a production project and you are encouraged to experiment !

6 ORGANISATION

Number of students per group : 3.

Deadline for submitting the project :
— 1st session (October 10, 11th 2023) : November 10th 2024.
— 2nd session (November 7, 8th 2024) : December 8th 2024.

The project should be shared through a github repo with contributions from all students. Please briefly indicate how work was divided between students (each student must have contributions to the repository).

If you used only libraries that were used during the course, you do not need to add a requirements.txt file. Else, you can include a **requirements.txt** file in order to facilitate installations for my tests, but please specify whether or not you did use libraries that were not used during the course.

`https://pip.pypa.io/en/stable/user_guide/#requirements-files`

You can reach me be email if you have questions.