iscte SINTRA
TECNOLOGIAS DIGITAIS
ECONOMIA E SOCIEDADE

Guidelines for Final Project

Supervised Machine Learning

Machine Learning Full Project

1 Introduction

In this final project, you are expected to build upon the foundations laid in Tutorials #1 and #2, extending your skills into the more advanced territory of machine learning. This project is not just an exercise in cleaning and analyzing a dataset but a comprehensive application of predictive modeling techniques using frameworks like Scikit-Learn. Your task involves implementing a range of machine learning algorithms, including neural networks, and rigorously evaluating their performance.

A key aspect of this project is the integration of hyperparameter tuning to optimize your models. This process is essential for enhancing the accuracy and efficiency of the predictive models you develop. Your work should reflect the standards of a major study, demanding conciseness and objectivity in presenting your findings. It's crucial to summarize the results from both the Exploratory Data Analysis (EDA) and the Machine Learning (ML) phases. This summary should offer clear insights without the need for scrolling through extensive notebook entries. The goal is to present your results in a way that highlights the key findings and decisions, offering a coherent narrative from data preprocessing to the final model evaluation. Your final submission should demonstrate not only your technical proficiency but also your ability to communicate the results in a clear and impactful manner.

2 Project Structure

2.1 Problem Definition

It is true that defining the problem, what we want to achieve, and/or what conclusion we want to draw can be difficult at times. If the context of *Exploratory Data Analysis* and the application of algorithms from *Machine Learning*, or any other *buzzword* is the solution, then what is the problem? As the saying goes, don't put the cart before the horse. Problems should be defined before requirements, and in turn, requirements before solutions, solutions before design, and design before technology. Sometimes we are too quick to switch to a new technology, tool or algorithm before we have concretely determined the problem we are trying to solve.

2.2 Data Acquisition

Although this may appear to be a simple step for those who have prior experience with data (or who work with them on a daily basis), it can be a significant challenge for those who want to investigate and solve a specific problem but cannot locate the main requirement for a project of this nature: the data. Fortunately, nowadays, due to the ease of access to information through the Internet and the need to study data, there is a set of *Dataset Finders*, aka repository indexers or simply repositories of *datasets*, that provide a vast set of *datasets* that are the *Holy Grail* for those who are starting their career as a machine learning engineer or data scientist. They have a vast set of *datasets*, *notebooks Jupyter* complete with examples of Exploratory Data Analysis (EDA) and its visualization, application of *Machine Learning* algorithms, among other examples. The list below, ordered by personal preference, presents several examples of such repositories:

1. **Kaggle Datasets** (`https://www.kaggle.com/datasets`) - This is one of the most popular data repositories. Each *dataset* is a small community where you can have a discussion about it, find public code or create your own projects. It has a large amount of real *datasets* of all shapes and sizes and in many different formats. You can also see *Kernels* associated with each *datasets*, where many community users provide *notebooks* of their own analyses. It is also possible to find *notebooks* that implement *Machine Learning* algorithms, and that solve classification / regression problems. It is also possible to participate in *Machine Learning* challenges on Kaggle, with the goal of achieving good results on the problems described above.

2. **Awesome Public Datasets - GitHub**

   (`https://github.com/awesomedata/awesome-public-datasets`) - This is also an excellent repository that organises the *datasets* by topics, such as Biology, Economics, Education, etc. Most of the *datasets* are public, but you should always check the user licences before using any *dataset*.

3. **UCI Machine Learning Repository** (https://archive.ics.uci.edu/ml) - Another example of a repository, from the University of California, where it separates the *datasets* according to the *Machine Learning* problem it is intended to solve (classification / regression) It classifies the datasets by the type of machine learning problem. You can find datasets for univariate and multivariate time series datasets, classification, regression or recommendation systems. Some of the datasets in the UCI are already cleaned and ready to be used.

4. **Amazon Datasets** (https://registry.opendata.aws/) - This repository contains many *datasets* from different areas, such as: (public transport, ecological resources, satellite images, among others).

5. **Google's Datasets Search Engine** (https://toolbox.google.com/datasetsearch) - In late 2018, also Google launched a repository where they can search various *datasets* from different areas.

6. **Data.world** (https://toolbox.google.com/datasetsearch) - Data.world is a platform that stands out as a comprehensive open dataset indexer, designed to facilitate the discovery, sharing, and management of data across various domains. It offers a user-friendly interface where researchers, data scientists, and analysts can find and contribute datasets on a wide range of topics.

7. **OpenML** (https://www.openml.org/) - OpenML is an online machine learning platform for sharing and organizing data, machine learning algorithms and experiments. It is designed to create a frictionless, networked ecosystem, that you can readily integrate into your existing processes/code/environments, allowing people all over the world to collaborate and build directly on each other's latest ideas, data and results, irrespective of the tools and infrastructure they happen to use.

2.3 Data Wrangling

This process consists of cleaning, manipulating and transforming data considered as *raw*, into data that can be read / processed by, e.g. algorithms *Machine Learning*. It is a necessary process when we want to 'filter' our data to serve as a basis for the problem to be solved, eliminating possible incorrect and/or null data, incorrect and/or invalid formats, selecting only interesting columns and discarding those that are not interesting (*Feature Selection / Feature Extraction*), detecting possible *outliers*, among others. The result of this process allows us to obtain clean visualizations of the data, good *Machine Learning* models and also the opposite: it could be the reason why our *Machine Learning* model does not have the expected results.

2.4 Exploratory Data Analysis (EDA)

This process is heavily dependent on the preceding process (2.3), as it is in this process that we perform the descriptive, statistical, and graphical analysis of our data. This process allows us to draw conclusions about our data using tabular visualization tools like **Pandas** and graphical visualization tools like **Matplotlib** and/or **Seaborn**. This is a crucial step because it allows us to identify potential problems with the data, patterns, classifications, correlations, and comparisons in our *dataset*. Furthermore, data categorization (i.e. qualitative vs quantitative) is critical for understanding and selecting the *Machine Learning* algorithms that we want to implement.

2.5 Predictive Modeling using Machine Learning algorithms

This procedure is the cherry on top. If we have a lot of information about the data, we can now move on to the step that allows us to present the final solution to our problem, which can be either classifying emotions based on text/speech or predicting future results of a given company's share price. It is critical to remember that algorithms are not magic wands. We will have to figure out how to choose the best tool for the job. As an example, imagine asking someone to hand you a Philips screwdriver and receiving either a screwdriver or a hammer. At best, it demonstrates a complete lack of comprehension. At worst, it makes the project impossible to complete. The same is true when it comes to selecting the best model. The incorrect model can result in poor performance at best and incorrect conclusions at worst.

2.6 Dashboarding with Streamlit (optional)

After the predictive modeling using machine learning algorithms, an essential step is to present the results in an easily interpretable and interactive manner. This is where Streamlit comes into play. Streamlit is an open-source Python library that simplifies the creation of web apps for machine learning and data science. It enables machine learning developers and data scientists to turn data scripts into shareable web applications with minimal effort.

In the context of our processes, Streamlit can be used to create dynamic dashboards that showcase the results of the Exploratory Data Analysis (EDA) and the outputs of the Machine Learning models. These dashboards can include interactive charts, maps, and data tables, providing a more engaging way to visualize the data insights and model predictions. Streamlit's simplicity and flexibility make it possible to quickly iterate and refine the presentation of the data, which is crucial for effective communication with stakeholders.

By incorporating Streamlit into the workflow, the end-to-end process from data wrangling to predictive modeling is not just about generating insights or predictions but also about effectively communicating these results to a non-technical audience. This step completes the data science lifecycle, ensuring that the work done in the earlier stages is accessible and actionable. This component is **entirely** optional!

3 Requirements

- Each group must **propose** a dataset, which may require similar **data wrangling** tasks as performed in Tutorial #1.
- Implement **predictive modeling** on the cleaned dataset to address either a classification and / or regression problem (or both), as performed in Tutorial #2.
- Use at least **6 machine learning** algorithms (being one of them **Artificial Neural Networks**).
- Use of at least **2** methods for **hyperparameter tuning**.
- Your analysis, implementation, and performance evaluation must be thoroughly documented in a single Jupyter Notebook file (`.ipynb`).
- (Optional) If you use **Streamlit** for dashboarding, you can use multiple `.py` files.
- All development code, problem description, along with a discussion of the results and obtained conlusions should be also included in this notebook.

4 Group Composition

- Final project will be conducted in groups of **2-3**.
- Evaluations will be carried out on an **individual** basis.

5 Delivery

Your final project should be completed in a group of two to three people and should take the form of a project report in which you clearly define the processes/steps as well as the conclusion of their development. There are at least **2/3 mandatory deliverables**:

- ***Dataset*** - CSV, SQL, TXT file, etc.
- ***Notebook*** - `.ipynb` file containing your entire project.
- **Streamlit** (optional) - If you make a dashboard with Streamlit, you also need to uploaded the needed `.py` files for dashboarding.

They should create a compressed file (.ZIP, .RAR, .TAR.GZ, ...) that should have all the deliverables mentioned above and others that they may find relevant for the project. The name of the compressed file should have the following format: **AAS-XX_FinalProject.YYY**, where ***XX*** is your Moodle group number and ***YYY*** is the file extension.

The final compressed file size must not exceed **1 GB**. If it exceeds, please upload the file via platforms such as **WeTransfer**, **Microsoft OneDrive** or **Google Drive**, create a simple text file and insert its shareable link. Then, just upload that text file containing the link.

The final project submission is made via Moodle, in the folder **Final Project**, and in the *exercise* with the designation '**[PROJ] Submission**'. Students can submit as many times as they want, but only the last submission is considered for evaluation. The deadline for submission is **23h59, UTC Lisbon / London timezone**, on the **5th of January 2025**.

6 Oral Discussion

- The oral discussion will be divided in **2 parts** on the **same day**, one for each of the **2 methodologies**: **Public** and **Private** discussions.
- The **Public** methodology consists in an open-to-public oral discussion, where each group will present to an audience, which can be members from other groups or students from another courses.
- The **Private** methodology, as the name states, consists in a private oral discussion, closed, in which only the members of the group and the professor will be present in the room.
- The **Private** oral discussions will take place on **January 6th**, from **09h30** to **12h30**, on **Room TBA**, and the **Public** ones from **14h00** to **17h30**, on **Room TBA**.
- These time frames can (and will) be **adjusted** according to the number of Private / Public discussions.
- Your group needs to choose a time slot in this link, and indicate the group number and the preferrable discussion methodology (public / private).

- To minimize downtime between sessions, it is recommended to opt for the **earliest** available time slot.
- Each discussion will last for **30 minutes**, **20 minutes** for you presentation, which need to focus on your implementation, key findings and main results; and **10 minutes** for individual discussion.
- Please arrive **5-10 minutes** early for your scheduled time slot.
- Only one group member needs to **bring a laptop** with the notebook prepared for demonstration and ready to connect to the room's video projector (HDMI interface). **Macbook users** that don't have HDMI interface must **bring their own HDMI adapter**.
- Grades will be provided after a thorough review of the project and not immediately post-discussion.

7 Evaluation Criteria

CA1: Quality and correctness of **data wrangling** using Pandas.

CA2: Effective use of **data visualization** tools like **Matplotlib** / **Seaborn** to present your findings.

CA3: Implementation and understanding of all the implemented **machine learning** algorithms (being one of them **Artificial Neural Networks**).

CA4: Effective use of the methods for **hyperparameter tuning**.

CA5: Correctness of the **performance evaluation** made on the implemented models.

CA6: Ability to interpret and discuss the final results of this study.

CA7: Clarity and organization of the Jupyter Notebook.

CA8: Technical proficiency and clarity in responding to the professor's queries.

8 Additional Information

- Maximum grade: 20 points.
- Weight: 40%.