DSTI School of Engineering

Applied MSc in Data Analytics
Applied MSc in Data Science & Artificial Intelligence
Applied MSc in Data Engineering & Artificial Intelligence

Project: Mental Illness Prediction

Instructor: Pauline Salis, PhD

Project Objectives:

The aim of the course project is to ensure students are comfortable enough to develop an end-to-end machine learning pipeline to answer a given problem or use case.

The project is a group project. Since this is a mixed course (DA/DS/DE), students are encouraged to form mixed groups to benefit from the competences of their teammates when working on different components of the pipeline. An example of an ideal group is a group of 3 members consisting of: 1 Data Analyst, 1 Data Scientist and 1 Data Engineer. Group diversity is **highly encouraged**.

Project Summary:

The dataset includes 76 columns and 196,102 rows of patient data. The data is raw, imperfect, and has not been cleaned or pre-processed.

Information about the dataset attributes can be found in the **NYSOMH_PCS2019_DataDictionary** file. Using the provided dataset, you are asked to train a model to predict the presence of mental illness. In particular, you are asked to predict the value in the **Mental Illness** column. You will apply the steps of a machine learning pipeline as seen in class to build and deploy a small web application that takes a patient's data as input on an interface and returns the prediction for the presence or absence of mental illness based on the patient's characteristics. The details of the web application implementation (framework, style, etc.) are left to you.

Each student in the group is expected to submit the following deliverables. **Failure to do so is considered an incomplete submission and will result in a grade of 0**:

- A Jupyter Notebook or Python script that includes exploratory analysis of the data, feature engineering and selection, model training, comparison and evaluation.
- The complete code of the web application containing the best chosen Machine Learning model as well as an interface that accepts standard inputs (this could be something as simple as a text box handling manual input values or a drop zone accepting CSV files containing rows of input) and returns predictions (in the format of your choosing, e.g., in a text box, table, downloadable CSV file, etc.). You can refer to Flask or Stremlit to get started with web application development in Python.
- A **5-10 page** report detailing your analysis and process structured in the following manner:
    - Introduction: Short description of the dataset and the problem at hand
    - Methodology: Section containing the steps of the pipeline, most notably:
        - Exploratory Data Analysis
        - Feature Engineering and Selection
        - Model Selection, Comparison and Evaluation
    - Results: Analysis of the model performances and justification of the best model choice as well as interpretation of the prediction results
    - Deployment: Short description of the web application
    - Conclusion: Concluding thoughts with insights of improving the project

    A link to a GitHub repository containing the project should also be included in the report.
- A short video of the web application running with a sample input provided and a sample prediction returned on the web interface.
- A GitHub repository containing the project (GitHub link to be provided in the report), including most notably a **README** file describing the project and a **requirements.txt** file to reproduce the project.

You may use additional resources as you see fit (provided you can justify how they can serve your solution). You can even consult similar solutions from the Internet. **However, this comes with a big responsibility: any submission that is over-plagiarized or does not reflect personal work will not be accepted. This also includes, but is not limited to, the use of Generative AI in project submissions**.

Project Evaluation:

The project will be evaluated using the following rubric. It contains the required items for a complete submission as well as bonus elements. The grading system is over 5 and the final grade will be transformed to a grade over 100.

- Jupyter Notebook (or Python script) containing entire machine learning pipeline **[1 point]**
- Complete web application code **[1 point]**
- Report (in PDF format) **[1 point]**
- Web application short demo video **[1 point]**
- GitHub repository **[1 point]**
- **BONUS:** Best group model performance in class **[1/2 point]**