Applied Machine Learning for Life Sciences Project

The course project is an opportunity for you to gain experience in completing an end-to-end tabular data science research project in the healthcare domain based on what we studied in class.

This project will account for 100% of the course grade.

Dataset from the UC Irvine ML Repository:

To download: https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

Dataset descriptions and ICD codes:

This research uses the dataset obtained from the Center for Machine Learning and Intelligent Systems at the University of California, Irvine covering 10 years (1998-2008) of Diabetes patient data gathered from 130 US hospitals having 70,000 distinct patients.

Every record was labelled as to whether the patient was readmitted within 30 days, readmitted after 30 days, or not readmitted at all.

The features and their description: https://www.hindawi.com/journals/bmri/2014/781670/tab1/

The values for the primary diagnosis: https://www.hindawi.com/journals/bmri/2014/781670/tab2/

Distributions of variable values and readmissions:
https://www.hindawi.com/journals/bmri/2014/781670/tab3/

Teams: You'll work in groups of 2 to 3 students, and you can team up with other students. If you can't find group mates, then notify the instructor of the course and she will find you teammates.

Deadline for project submission (report and code): 18:00, 14/03/24

Project should be submitted to: The course instructor's email ort.dayan@gmail.com

Project Instructions:

Project Workflow and Report: The report should be written in English and according to the structure provided in the below Project Workflow and Report Structure section of this document.

Code:

- Write your code using .py extension files (e.g., using Pycharm or VSCode IDEs). Only the EDA section should be submitted in a Jupyter Notebook file format.
- You are encouraged to use Git for version control of your code and to work on the same code files with your teammates.
- You are encouraged to perform code reviews for your teammates.
- You are encouraged to use GPU for faster hyperparameter tunning.
- If you have any concerns working with one of your project teammates, please contact the course instructor about this.
- Please include a link to a GitHub repository or zip file with the code for your final project.

Violations of the Honor Code: You may consult any resources with implementations of ideas and code that you may want to incorporate into your project, so long as you clearly cite your sources in your writeup and code comments. However, under no circumstances may you look at another group's code or incorporate their code into your project.

Evaluation Breakdown:

- Validity of your methodology: The final grade will be mostly based on the validity of your methodology including EDA, preprocessing, the algorithms you chose to fine tune, how you fine tune, error analysis and the reasoning you provide for your entire methodology rather than how good is the performance of your best performing model. In addition, replicating the methodology and results in a paper using the same dataset is a great way to start but you are expected to try to improve on their methodology and provide reasoning for your attempts.
- Originality in your methodology (rather than implementing existing ones in the literature) and novel insight in your explanations including but not limited to how you performed EDA, prepared the data, or chose your target for prediction.
- Validity of your explanations provided in the report (sections 2-8) for the entire methodology you have implemented through the project.
- If the authors convey novel insight about the task.
- Code implementation: how well you have followed the coding methodologies and guidelines

taught in class with emphasis on organizing your code in functions, using Scikit-Learn Pipelines, fine tuning your best performing models using Scikit-Learn GridSearchCV class and Optuna framework as well as using Python packages for synthesizing samples.

- Report completeness and clarity with respect to visualizations and explaining the work that has been done and the reasoning behind it.

Project Workflow and Report Structure:

Section 1 - Introduction:

- Overview (based on literature) of the diabetes medical condition with relation to the dataset features.
- Literature review of ML implementations using the same dataset with a discussion on their strengths and weaknesses.

Section 2 - Description of the dataset

Section 3 -

- Your choice of objective (target/s and features) and what are the current solutions in the literature for predicting your choice of target/s?
- Is your objective a supervised or unsupervised ML problem?
- What is your selection for performance measures?
- List your assumptions and explain if you managed to verify them.

Section 4 - How you split the dataset into train and test sets Remember the importance of performing a stratified split and preventing train validation/test leakage e.g., by grouping data of same patient records.

Section 5 - EDA: Exploratory Data Analysis (EDA) using a Jupyter Notebook including your conclusions based on it.

Section 6 - Data Preparation:

- As first step check that the dataset complies with the Tidy Data Requirements and if needed perform data preprocessing [1].
- Perform feature engineering (feature transformations/selection/extraction) to comply with the ML algorithms requirements and to reduce curse of dimensionality and create features that are stronger predictors.

- Please note that in terms of evaluation there will be an emphasis on techniques to account for the imbalance in the dataset including balancing the dataset by undersampling the majority class and oversampling the minority class including by synthesizing new samples using SMOTE and GANs including CopulaGAN [2] from the SDV library and a collection of GANs from the ydata-synthetic library [3].
- This part should be implemented using Scikit-learn Pipeline (including the previous data preparation steps).

Section 7 - Train appropriate models:

- List appropriate models and state the reasoning for choosing them and how you trained and evaluated them using their default hyperparameters including Logistic Regression/Softmax Regression, SVMs, Random Forest (including using out of bags evaluation and the Imbalanced-Learn library BalancedRandomForestClassifier class, XGBoost, LGBM and CatBoost.
- This part should be implemented using Scikit-learn Pipeline (including the data preparation steps from section 6 of the report).

Section 8 - Fine Tuning: How you fine-tuned, performed model calibration, and evaluated the performance of your short list of best models.

- Please note that in terms of evaluation there will be an emphasis on how well you fine tune the following 4 algorithms: LGBM, CATBoost, and XGBoost as well as Random Forest (including using out of bags evaluation and the Imbalanced-Learn library as mentioned in section 7 of the report). In case, you found a better performing algorithm in the previous section, fine tune it instead of one of LGBM/CatBoost/XGBoost.
- Fine tune also the models hyperparameters that can account for the imbalance. E.g., For LGBM using the `is_unbalanced` and `scale_pos_weight` hyperparameters.
- Please follow the guidelines including code implementations provided in class and as described in the lecture notes and Python files shared on how to fine tune your models including using both the GridSearchCV class and the Optuna framework.
- Account for randomness by evaluating across 15 seeds.
- In this part the estimator for GridSearchCV or Optuna should be the pipeline from section 6 of the report, you can follow this tutorial on how to implement this [4].

Section 9 - Discussion:

- Present your solution with clear visualizations and statements.
- Include what you have learned, what worked and what did not, what assumptions were made, what your system limitations are and how does it compare to existing solutions.

Section 10 - Team Members Individual Contribution: List which parts of the project including report and code were completed by which group mate. This is to make sure team members are contribute equally to the work on the project.

Section 11 - References: List all references used for completing your project.

I look forward to reading about your project!