CMPT459 Spring 2021
Data Mining
Martin Ester
TAs: Madana Krishnan V K and Arash Khoeini

Milestone 2 of the Course Project

Deadline: March 22nd

Total marks: 100

Introduction

In the first milestone, you have completed the data pre-processing steps. You should now have a *train* and *test* dataset that is cleaned to an extent, obtained by merging the cases and locations datasets. You will be using only the *train* dataset in this milestone.

In the second milestone, you will be building various classification models and use metrics to evaluate the performance of the models. Each group member has to build one classification model. Thus each group will have 2 or 3 models built, depending on the group size. Remind yourselves that the problem statement for this project is to predict the outcome of a case.

Tasks

2.1 Splitting dataset (**5 marks**)
Split the *train* dataset further into *train* dataset and *validation* dataset. Train to validation ratio should be 80:20. To get deterministic results every time, you can set the random state to a constant value.

2.2 Build models (**40 marks**)
As mentioned above, each individual in a group has to build a classification model. The models can be of any type, and you can use any existing Python libraries (ex: Scikit-Learn) to build them. One out of the two/three models MUST be a variant of the boosting tree (ex: XGBoost, AdaBoost, LightGBM etc.). The other one/two models could include SVMs, KNN, Decision Trees, Random Forests, MLPs, Naive Bayes or any other classifiers that you think could be appropriate for the problem statement. Explain the reason why you use a particular model over the other.

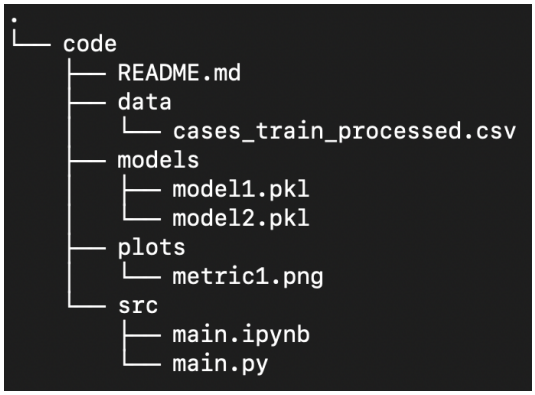
Save each trained model to your disk as a .pkl file (the models would look like xgb_classifier.pkl, rf_classifier.pkl), and include the models in submission.

2.3 Evaluation (**35 marks**)
Load the saved models from task 2.2. Once loaded, evaluate the models on both the *train* set and the *validation* set. Choose appropriate metrics to do the evaluation. Report the scores of your metrics for the *train* and the *validation* set and interpret them. Which of the metric(s) are most important for this classification problem? Provide an in-depth, quantitative and qualitative discussion about the observations you make.

2.4 Overfitting (**20 marks**)
It is not uncommon for the classification models to overfit. Do you observe overfitting in the models that you trained? How do you check for overfitting? Explain steps taken by using plots and/or metrics evaluated. You can vary the values of at least one hyperparameter, train models for different values of that hyperparameter and then compare the performance.

Submission (Code + Report)

A). Code
Submit a 'code.zip' file with the following contents. It should contain the code that you have written for the above tasks. The code could contain how train-test split was done, steps performed to fit the classification models, saving the models as .pkl files, loading the .pkl files to perform evaluation, different evaluation metrics used. Make sure to submit the saved models in the submission. You can include any plots that you might generate.
Please note that in order for the TAs to run the code for milestone 2, you would need to provide the train *dataset* that you generated from milestone 1. Include the *dataset* within a folder named 'data' as shown in the figure. The structure can look like this.



- *data/* folder contains *cases_train_processed.csv* obtained from milestone 1
- *models/* folder contains saved trained models as .pkl files
- *plots/* folder contains any plots that you wish to generate
- *src/* folder contains code and scripts
    - *src/main.py*
    - *src/main.ipynb*
- *README.md* for any special instructions for code execution (ex: if main.py or main.ipynb should be executed, if third party APIs are used, high running time, links to dataset or saved models)

You can **either** submit a .py OR a .ipynb file. Running '*src/main.ipynb --> Cell --> Run All*' **or** '*python src/main.py*' should reproduce all the reported results with the same directory structure as described above. You can use helper functions to support your codes. Ensure to use relative paths to access files and folders within the code.

B). Report
Briefly explain your work for task 2.1. Explain in more detail your work for tasks 2.2, 2.3 and 2.4. Reports should **NOT** be more than 2 pages. Submit a 'report.pdf' file.

NOTE:

- For milestone 2, you can vary the values for at least one of the hyperparameters to check for overfitting. In milestone 3, you will be tuning all the combinations of hyperparameters to find the best model.
- Include the code to save and load the models in the submission. Ensure to submit the two/three saved models (.pkl files).
- Please include steps for execution in the README.md file.
- code.zip should be less than 30MB. Keeping cases_train_processed.csv and model<n>.pkl files could bloat the size. If file size exceeds 30MB, you can follow one or both of these steps:
    - Upload cases_train_processed.csv to Github and read directly from Github.
    - Upload model<n>.pkl files to Google drive and submit a downloadable link.