Project I

- Work individually.
- Submit a code and a detailed project report. The names of the files should be YOURNAME.py (or YOURNAME.ipynb) and YOURNAME.docx (e.g. OrsanOzener.py, OrsanOzener.docx)
- Any type of plagiarism will not be tolerated and will lead to disciplinary actions.
- **Due Date: 18th of May 2020, 13:00. Please submit your report through LMS. Only ONE submission per person.**

1 Introduction

In this project assignment, you are supposed to work on a classification/regression problem (could be binary/multi-nominal classification or regression) of your choice. In that aspect, you are supposed to find a public data suitable for the project (please make sure that the data you have selected have sufficient number of observations and features and the missing values etc are dealt properly). On the given data you are supposed to use standard methods such a decision trees, random forests, boosting methods, xgboost, lightgbm and svm (maybe not all of them, but please do not use just one). Recall that we discuss the feature importance concept at length during class and how different models' feature importance values could be quite different form one another. Based on the feature importance values of the models, how would you make your model leaner (work with fewer number of features) to achieve higher performance (both solution quality and computational time, though latter is not as important as the former for the sake of this project). Alternative to the feature importance values, you are supposed to use recursive feature elimination and compare the solutions to the previous ones. Even though we did not discussed this subject in class, the python implementation is a quite straight forward (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html). Nevertheless if you have any question about this module, please let me know.

To sum up, you will compare three types of solutions (i) solutions using all available features, (ii) solutions using a subset of features based on the feature importance values acquired by the method used, (iii) solutions using a subset of features based on the recursive feature elimination process. Naturally, the number of features remaining in the second and third will be based on your implementation but please do this a rigorous manner with justification. For instance, say you have 100 features, you might keep any number up to 100

based on the feature importance values, however I would rather have a justification such as "with the normalized feature importance values, I have decided to keep features up until 70% of the feature importance values". The final performance criteria should be appropriate based on the problem of your choice (e.g. MAD, MSE, Accuracy, AUC, LogLoss).

If you have any questions, please send an email to: orsan.ozener@ozyegin.edu.tr