**Statistical/Machine Learning:**

**Preliminary Project Report**

Gabriel de Medeiros

Evan Callaghan

November 28, 2021

**Files**

This project is consisted of tree .py files, one .ipynb file, and four .csv files.

The files in the following table should be ran in the given order:

|  |  |  |
| --- | --- | --- |
| File Name & Format | Function | Description |
| FraudTrain.csv | Raw Data | Original data from Kaggle. Data is already split into training. |
| FraudTest.csv | Raw Data | Original data from Kaggle. Data is already split into testing. |
| NewFeatures.ipynb | Feature Creation | Adds custom features to the raw data set |
| CleanTrainData.csv | Clean Data | This train data set was generated into the NewFeatures.ipynb file |
| CleanTestData.csv | Clean Data | This test data set was generated into the NewFeatures.ipynb file |
| HyperParameters.py | Create Hyper Parameters | This file creates data frames with different combinations of hyper parameters and cut off values. |
| Get\_Variable\_Importance.py | Gets Variable Importance | Runs Decision Trees and Random Forest models to get the most important variables from the data set. |
| EvaluateModels.py | Create Models | This code generate models for predictions and uses hyper parameters from the HyperParameters.py file. |
| CallModels.py | Run models | This file, when ran, loads the CleanData train and test, gets the important variables from the file Get\_Variable\_Importance.py, and returns the results of the models. |

**Running the code**

The first file to be ran is NewFeatures.ipynb using the filesFraudTrain.csv and FraudTest.csv. After running this code, the next step is to load the files that are returned from the procedure into CallModels.py. If the files are all in the same folder, the next step is to run the callModels.py file in a console since it is designed to call the functions from other .py files. The results for analysis are returned in .csv files. There is no estimated time to run the CallModels.py file, but it should take more than a week using the entire data set. Using the current set up that splits the data into 10% of the size of the original data set and stratifies the number of 1s and 0s, it should still take more than 4 days, plus AdaBoost using SVM that the group did not manage to run because of the time it took to run.