HW06 REPORT

In this homework, I developed a machine learning algorithm in Python using scikit-learn library for a real-life regression problem from finance industry. My machine learning algorithm predicted the number of cash withdrawals from 47 different ATMs of a bank using the information given about each ATM and the withdrawal date.

After importing pandas library, as a first step, I read the both test and training data using read.csv() function. I assigned them to train\_X and and test\_X variables. I retrieved TRX\_COUNT column and assigned as train\_Y variable, after that, I drop that column from train\_X variable. The reason is to use train\_X and train\_Y variables in cross validation. Before doing cross validation, I made one hot encoding on train\_X data. One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. Hence, IDENTITY column consists of 9-digit values and correspond 47 different ATMs, I made one hot encoding on IDENTITY column. After that, I inserted a new feature called WEEKEND to the train\_X to capture change in cash withdrawals on weekends and weekdays. To make cross validation, I split train\_X and train\_Y as a test and train data. I give 0.1 to test size parameter (small part of the actual train data) to train my model on more data points.

As a model, I tried MLPRegressor, DecisionTreeRegressor and RandomForestRegressor from scikit-learn library. After lots of tries, I determined on RandomForestRegressor. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. It provides higher accuracy. If there are more trees, it does not allow overfitting and it can handle a large dataset with higher dimesionality. I gave n\_estimators=50, since 50 trees is enough for our dataset.

After that, I fit my train data to my model, and I predicted y\_predict values for test data from cross validation. I calculated root mean squared error and mean absolute error, and found 15.8433 and 8,8142 respectively.

As a final step, I made predictions on our original test data and wrote the results in a csv file.