ISA IM: Project 1
IEEE – CIS Fraud Detection Report
Author: Shashwata Sourav Roy Samya

➢ ABSTRACT

IEEE – CIS Fraud Detection is a competition on Kaggle which is a binary classification problem where I have to generate the likelihood of a fraudulent transaction. Merging, dropping columns containing high percentage of null values and filling the remaining null values with suitable statistical approach were taken into consideration. I also looked tried to exploit the seaborn library while preparing both my training and test dataset for modelling. XGBoost model was used for training which resulted in obtaining an accuracy of 0.930087 on my test dataset.

➢ METHODOLOGY

Train and test datasets each came in two parts, one of identity and another of transaction. Both of these datasets were left joined with respect to their TransactionID feature. This resulted in obtaining a full train and a full test dataset.

Initially unique columns were found between the two datasets and renamed accordingly to keep them identical. Then I went on calculating the percentage of null values in every column of both the train and test datasets. Upon examining my results, any columns containing a null value percentage of over 20% were dropped from both of my respective datasets – 20% was chosen as a threshold because of my experience from previous works and also taking into note of other peoples' different works that I have come across on the internet. It is generally considered that 20%-30% null values of the full dataset is tolerable. In my dataset, total values were very close to 0.6 million and 20% of it being null values would mean that the amount of it would be just over 0.1 million. This resulted with columns of 183 and 182 for train and test datasets respectively.

Upon removing the unwanted columns, I looked into the datatypes of the remaining columns and came across 4 of them as object types – rest were numeric. Performing a percentage calculation of how many of these object columns contained isFraud value as 1, I found that except for one variable in my Product CD column, every variable of all these columns contained around 5%. I saw 5% as being significantly less percentage and dropped 3 of the object type columns except for Product CD as it contained one variable having considerably higher than 5%.

Next, I went on filling the remaining null values in our newly shaped train and test datasets with mode values from their respective columns. Mode was chosen over mean on the basis that any extreme values in the columns may deter me from a suitable value and also the fact that my datasets were very large.

For EDA, I looked into correlation between the features and plotted them on a heatmap. A pattern could be observed in it. I further looked into seaborn library's distplots of some of the features and also some pairplots with isFraud and other features to get a better idea of the dataset that I had.

Since I only had one feature with 4 variables as an object datatype, the get_dummies method was used to encode those 4 variables with my train dataset. Furthermore, the column TransactionID was later dropped from both of my datasets in preparation for my modelling. The isFraud column from my train dataset was set as my target separately whilst also dropping it from my train dataset.

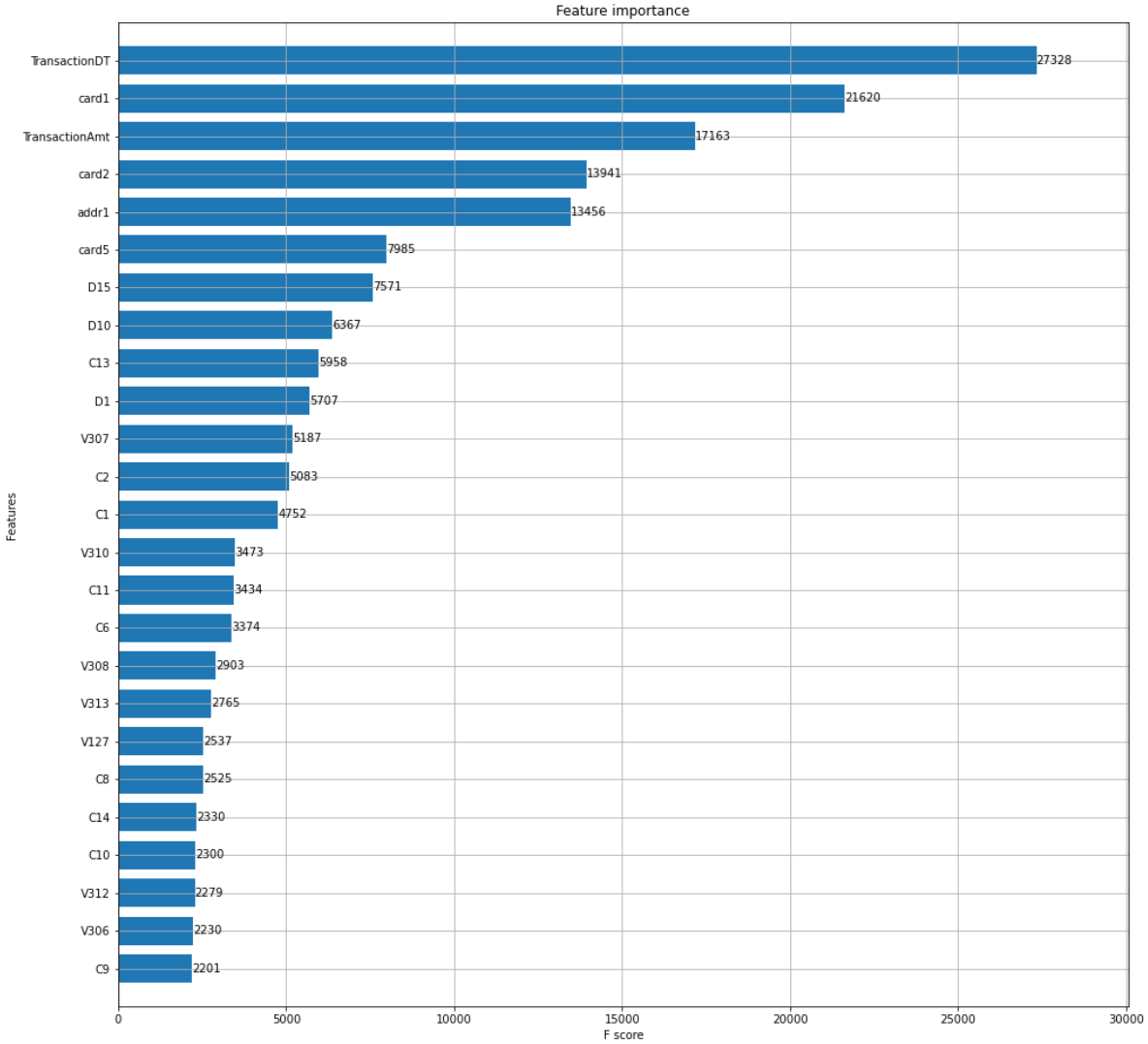
➢ RESULT ANALYSIS

I have used XGBoost model as I found it to be robust while dealing with large number of observations and when maximum of our data consists of numeric features. XGBoost uses gradient boosting which generates a prediction based on several 'weak learners' such as decision trees.

Hyperparameter tuning was done based on references from other code notebooks and different articles on the internet. GridSearchCV or RandomSearchCV were not used due to the limitations of my machine. Thus, I ended up my running my model at different learning rates of 0.02, 0.04, 0.05 separately. From my trial runs I found a learning rate of 0.04 to be the most convenient choice. Hence keeping that fixed I trialed with max depth of 17, 19, 21, 25, 50 to get the best combination. I later found a max depth of 21 to be the best partner for my learning rate. All other parameters that were set away from their default values were referenced from another notebook that followed a similar approach to mine.

This resulted me with a model evaluation score of 0.930087 from Kaggle.

I also looked into the most important features in my model from which I could see that the top 3 important features were TransactionDT, card1 and TransactionAmt. Below I have added the bar graph of our 20 most important features.



➢ CONCLUSION

The accuracy of our model can be even more improved by a polished feature selection process such as ‘Recursive Feature Selection’ along with feature engineering. We can also hope to further investigate and perform extensive feature engineering on 2 of our most important features, TransactionDT and TransactionAmt. Other classifications models may also be used to compare our current accuracy on this dataset.