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 are taken into considering while preparing both our training and test dataset for modelling. XGBoost model was used for training which then resulted in us obtaining an accuracy of 0.923940 on our test dataset.

➢ METHODOLOGY

Train and test datasets each came in two parts, one for identity and one for transaction. Both of these datasets were left joined with respect to their TransactionID feature. This resulted us 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 our results, any columns containing a null value percentage of over 5 - in respect to the missing data dataframe the value was 0.05 – were dropped from both our respective datasets. This resulted us with columns of 112 and 111 for train and test datasets respectively.

Upon removing unwanted columns, I went on filling the remaining null values in our newly formed 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 us from a suitable value.

The get_dummies method was used to further enhance our train and test datasets. Furthermore, the column TransactionID was dropped from both of our datasets. The isFraud column from our train dataset was set as our target separately while also dropping that column from our 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 most if not all 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. A learning rate of 0.05, gamma of 0.0468, max depth of 50 were used along

with a host of other parameters which resulted was in model evaluation of 0.923940 after predictions using our test dataset.

➢ CONCLUSION

The accuracy of our model can even more be improved by a polished feature selection process such as ‘Recursive Feature Selection’ along with feature engineering. Other classifications models may also be used to compare our accuracy on this dataset.