Project Report: It's a Fraud

Darshak Jivrajani IMT2020119 Darshak.Jivrajani@iiitb.ac.in
Arhant Arora IMT2020503 Arhant.Arora@iiitb.ac.in

Overview

Given data about transactions, train a model which tells if a given transaction is fraudulent or not.

The Train Dataset given to us has 434 columns and rows. To train a model accurately, we clearly need to perform heavy preprocessing and EDA on the dataset.
On observation, we also find that the dataset is highly biased with only 3.5% of the entries as fraud.

1. Preprocessing and EDA:

Removing Null Values:

Since the dataset has only 3.5% of fraud entries, any row with more than 96.5% of NULL Values was removed.

Dealing with Vxx, Cx and Dx columns:

We grouped the V columns on the basis of NULL value %, got 9 different groups and looked at the correlation matrix of each group. This helped us remove a large number of columns. For the V Columns left, we again looked at the correlation matrix to remove any other similarities among columns. We then performed the same with the C and D columns. This helped us decrease number of columns from 434 to around 102.

Dealing with Skewness:

We looked at the skewness of all the columns. We calculated the square root for the columns with skewness greater than 5.1 and square for the columns with skewness less than -4. After this we calculated the skewness again and this time we removed the columns who's skewness was outside of the range -4 to 5.1.

Filling NULL Values:

For categorical columns, we filled the with the median of the data values we had. For non-categorical columns, same process as mentioned above except we filled the values with mean if we still had any empty columns, we'll fill it as per our data.

Outlier Removal:

We checked the outlier for all columns ad for values between 2% and 98%, we kept it and removed the rest. We are finally ready to train our models.

2. Models and Final Scores:

Here is a summary of the models used:

Model	**Hyperparameter**	**Value**	**Model Score**
KNN	metric algorithm leaf_size n_neighbours weights	manhattan ball_tree 10 11 distance	0.82645
Logistic Regression	C max_iter penalty solver	0.1 100000 l2 lbfgs	0.756
Naive Bayes	var_smoothing	1	0.686
Bagging (With Decision Tree)	base_estimator__max_depth max_samples	5 0.5	0.85897
ADA Boost (With Decision Tree)	base_estimator__max_depth base_estimator_min_samples_leaf	10 10	0.8475
XG Boost	colsample_bytree gamma learning_rate max_depth reg_alpha objective n_estimators njobs	0.75 0.65 0.1 20 0.4 binary:logistic 8000 -1	0.928
Neural Networks	Layers loss_function epochs batch_size optimizer	2(relu, sigmoid) binary_crossentropy 20 100 adam	0.69716

Table 1: Table showing Hyperparameter values for different models and their final scores.

3. Observation And Conclusion:

We can observe that the best model for our data is XGBoost. This can be because our data is a high bias one and XGBoost aims to reduce bias. Also, XGBoost works well on a heterogeneous data (our data becomes highly uncorrelated after the initial preprocessing.) Thus we can conclude XGBoost is the best model for a highly-biased heterogeneous binary classification data like the one we had.