CREDIT CARD FRAUD DETECTION

Problem Statement

Although digital transactions in India registered a 51% growth in 2018-19, their safety remains a concern. Fraudulent activities have increased severalfold, with around 52,304 cases of credit/debit card fraud reported in FY'19 alone. Due to this steep increase in banking frauds, it is the need of the hour to detect these fraudulent transactions in time in order to help consumers as well as banks, who are losing their credit worth each day. Machine learning can play a vital role in detecting fraudulent transactions. With the help of machine learning models, we can predict the fraudulent credit card transactions.

Understanding and Defining the Problem

Credit card fraud costs consumers and the financial company billions of dollars annually, and the fraudsters continuously try to find new rules and tactics to commit illegal actions. Among different ways of frauds, Skimming is the most common one, which is the way of duplicating of information located on the magnetic strip of the card. Apart from this, the other ways are:

- Clone transactions
- Account theft and suspicious transactions
- False application fraud
- Account takeover
- Manipulation/alteration of genuine cards
- Creation of counterfeit cards
- Stolen/lost credit cards
- Fraudulent telemarketing

Data Understanding

The data set includes credit card transactions made by European cardholders over a period of two days in September 2013. Out of a total of 2,84,807 transactions, 492 were fraudulent. This data set is highly unbalanced, with the positive class (frauds) accounting for 0.172% of the total transactions.
Apart from '**time**' and '**amount**', all the other features (V1, V2, V3, up to V28) are the principal components obtained using PCA.
The feature '**time**' contains the seconds elapsed between the first transaction in the data set and the subsequent transactions. The feature '**amount**' is the transaction amount.
The feature '**class**' represents class labelling, and it takes the value 1 in cases of fraud and 0 in others.

Approach

1. **Data Analysis**: We analyze the data by performing the following
   - Loading the data
   - Analyzing the null values
   - Analyzing the missing the values
   - Missing value treatment
2. **Exploratory Data Analytics (EDA)**: We will perform Univariate and Bivariate analyses of the data, followed by feature transformations, if necessary. We will check if there is any skewness in the data and try to mitigate it, as it might cause problems during the model-building phase. Remove outlier if necessary

3. **Train/Test Split**: Break the dataset into train-test split and for validation, we would use the k-fold cross-validation method.
4. **Class Imbalance**: we will use ADAptive SYNthetic (ADASYN) to balance the classes. It also lowers the bias introduced by the class imbalance.
5. **Model-Building/Hyperparameter Tuning**: This is the final step at which we can try different models beginning with simple model such as Logistic regression, k nearest neighbor followed by random forest, SVM. We will apply boosting strategy like XG boost, adaboost over models to convert weak learners to strong learners with regularization techniques L1, L2. We will fine-tune their hyperparameters until we get the desired level of performance.
6. **Model Evaluation Criteria**: The models are evaluated by the False-Positive Rate (FPR), recall, precision, and Area under the Curve (AUC).
   Let us understand these four metrics in a bit more detail with regards to the given problem.
   True Positives (TP): The model has predicted the transaction to be fraudulent and in real life the transaction is fraudulent.
   True Negatives (TN): The model has predicted a transaction to be a non-fraudulent one and in real life the transaction is non-fraudulent.
   False Positives (FP): The model has predicted the transactions to be fraudulent whereas in real life the given transaction is not fraudulent. These are also known as Type 1 errors.
   False Negatives (FN): The model has predicted the transactions to be non-fraudulent whereas in real life the transactions are fraudulent. These are also known as Type 2 errors.

   Ideally, for a perfect model, we would want the values of TPs and TNs to be very high and our FPs and FNs to be very low. Also, for this problem it's an absolute necessary to keep the False Negative values as low as possible because those are the actual frauds.