Credit Card Fraud Detection

Problem Statement

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash.

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

We have to build a classification model to predict whether a transaction is fraudulent or not.

Approach

1. Data Understanding, Data preparation and EDA

First look at the data provided suggests that it is highly imbalanced. The positive class (frauds) account for only 0.172% of all transactions.

Class is the target variable which we have to predict where 0 is the normal transaction and 1 is fraudulent transaction.

Features V1, V2, V3….V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset.

Class Imbalances:

The normal Oversampling method won't be used here as it does not add any new information to

the dataset and Undersampling will also not be used as it leads to the loss of information.

Next, we will try the below two class imbalance handling techniques:

• SMOTE is a process where you can generate new data points, which lie vectorially between

two data points that belong to the minority class.

• ADASYN is similar to SMOTE, with a minor change i.e. the number of synthetic samples that it will add will have a density distribution. The aim here is to create synthetic data for minority examples that are harder to learn, rather than the easier ones.

2. **Model Selection and Model Building:**

We will start building the model with the train-test split. (At least 100 class 1 rows should be there in the test split), use the stratified split here. (80-20 ratio can be used)
We need to find which ML model works good with the imbalance data and have better results on the test data.
• Logistic regression works best when the data is linearly separable and needs to be interpretable.

3. **Model Evaluation:**

We will use sklearn.metrics.roc_auc_score for this as AOC and ROC metric in sklearn is used as the metric for highly imbalanced data-set, rest all fails.
ROC have better false negative than the false positives.
ROC-Curve = Plot between TPR and FPR
The threshold with highest value for TPR-FPR on the train set is usually the best cut-off.
We should not use the confusion matrix as the performance metrics as well as they have internally defined hard threshold of 0.5.
We also can't completely rely on the precision, recall and F1-score for now as they also have their strings attached of some threshold value.
ROC curve takes into cognizance of all the possible threshold values.
The ROC curve is used to understand the strength of the model by evaluating the performance of the model at all the classification thresholds.
Because the ROC curve is measured at all thresholds, the best threshold would be one at which the TPR is high and FPR is low, i.e., misclassifications are low.

4. **Cost-Benefit Analysis:**

Depending on the use case, we have to account for what we need: high precision or high recall.
For banks with smaller average transaction value, we would want high precision because we only want to label relevant transactions as fraudulent. For every transaction that is flagged as fraudulent, you can add the human element to verify whether the transaction was done

by calling the customer. However, when precision is low, such tasks are a burden because the human element has to be increased.

For banks having a larger transaction value, if the recall is low, i.e., it is unable to detect transactions that are labelled as non-fraudulent. So consider the losses if the missed transaction was a high-value fraudulent one, for e.g., a transaction of $10,000?

So here, to save banks from high-value fraudulent transactions, we have to focus on a high recall in order to detect actual fraudulent transactions.

We need to determine how much profit or dollar/rupee value we are saving with our best selected model.