Summary

Abstract

In the banking industry, credit card fraud detection using machine learning is not just a trend but a necessity for them to put proactive monitoring and fraud prevention mechanisms in place. Machine learning is helping these institutions to reduce time-consuming manual reviews, costly chargebacks and fees, and denials of legitimate transactions.

Objective

The objective of this project is **to predict fraudulent credit card transactions with the help of machine learning models.**

Observation

1. The data set includes credit card transactions made by European cardholders over a period of two days in September 2013.
2. **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.**
3. The data set has also been modified with Principal Component Analysis (PCA) to maintain confidentiality. Apart from ‘time’ and ‘amount’, all the other features **(V1, V2, V3, up to V28)** are the principal components obtained using PCA.
4. 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.

Project Pipeline

The project pipeline can be briefly summarized in the following four steps:

1. **Data Understanding**: We need to load the data and understand the features present in it. This would help we choose the features that we will need for our final model.
2. **Exploratory data analytics (EDA)**: We will be performing univariate and bivariate analyses of the data, followed by feature transformations, if necessary. However, we can check if there is any skewness in the data and try to mitigate it, as it might cause problems during the model-building phase.
3. **Train/Test Split**: Now we will be doing the train/test split, which we can perform in order to check the performance of our models with unseen data. Here, for validation, we can use the k-fold cross-validation method.
4. **Model-Building/Hyperparameter Tuning**: In this step we can try different models and fine-tune their hyperparameters until we get the desired level of performance on the given dataset. We will try and see if we get a better model by the various sampling techniques.
5. **Model Evaluation**: Evaluate the models using appropriate evaluation metrics. Note that since the data is imbalanced it is more important to identify which are fraudulent transactions accurately than the non-fraudulent.