Capstone Project
Credit Card Fraud Detection
Summary

Problem Statement:

To predict fraudulent credit card transactions with the help of machine learning models.

For many banks, retaining high profitable customers is the number one business goal. Banking fraud, however, poses a significant threat to this goal for different banks. In terms of substantial financial losses, trust and credibility, this is a concerning issue to both banks and customers alike.

Credit card fraud is any dishonest act and behaviour to obtain information without the proper authorization from the account holder for financial gain. 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.

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 just 0.172% of the total transactions. 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. 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.

Analysis Process:

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

1. Data Understanding: Here, we need to load the data and understand the features present in it. This would help to choose the features that we will need for our final model.
2. Exploratory data analytics (EDA): Here, we need to perform univariate and bivariate analyses of the data, followed by feature transformations, if necessary. For the current data set, because Gaussian variables are used, we do not need to perform Z-scaling. 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: Here, we are familiar with the train/test split, which we can perform in order to check the performance of your models with unseen data. Here, for validation, we can use the k-fold cross-validation method. We need to choose an appropriate k value so that the minority class is correctly represented in the test folds.

4. Model-Building/Hyperparameter Tuning: This is the final step at which we can try different models and fine-tune their hyperparameters until we get the desired level of performance.