# **Project Proposal**

# **for**

# **Credit Card Fraud Detection Machine Learning Models**

## **Business Problem Overview**

## With the rise of digital payment channels, the number of fraudulent transactions is going very high day by day. It has been estimated by [Nilson report](https://nilsonreport.com/upload/content_promo/The_Nilson_Report_Issue_1164.pdf" \t "_blank) that by 2020 the banking frauds would account to $30 billion worldwide. It is difficult to manually predict fraudulent transactions which is also error prone

## **Problem Statement**

## Build machine learning models to predict fraudulent credit card transactions

## **Input Dataset**

## 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.  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

## **Machine Learning Model Development**

## **Data Understanding**: Check the no. of columns, no. of rows, data types, null values, etc. This dataset does not have any null values. However, the data set is highly imbalanced, with the positive class (frauds) accounting for 0.172% of the total transactions. So, the following techniques can be used for treatmentof **class imbalances**:

## **Over- sampling**

## **Under-sampling**

## **SMOTE**

## **AdaSyn**

* **Exploratory Data Analysis**: Check the distribution of classes over the dataset as well as with respect to attributes: time and amount. Drop unnecessary columns (if any)
* **Train/Test Split**: Split the dataset into 2 parts: training data and testing data in a ratio of 80:20. Check the skewness by plotting histogram of variables and use PowerTransformer if the data is skewed
* **Model Building and Hyperparameter Tuning**: Use the following machine learning algorithms to train the models:
  + **Logistic Regression**
  + **Support Vector Machine**
  + **Decision Tree**
  + **KNN**
  + **Random Forest**
  + **XGBoost**

The hyperparameters for the models will be tuned using **GridSearchCV**. GridSearchCV helps to loop through predefined hyperparameters and fits the model on the training set. So, in the end, the best parameters from the listed hyperparameters can be selected. Repeat the following steps to find the best model:

* perform **cross validation**
* perform **hyperparameter tuning**
* evaluate the model result based on the evaluation metric
* find the optimal value of the hyperparameters
* **Model Evaluation Metric**: Since we have class imbalance in the dataset, we will use the area under the receiver operating characteristic curve (**AUC - ROC**) for comparison of models. A ROC curve is a diagnostic plot for summarizing the behavior of a model by calculating the **false positive rate** and **true positive rate** for a set of predictions by the model under different thresholds

Choose the model with the highest value of AUC ROC score