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

**Shivapriya Pillalamarri - 00905151**

## ****Problem Statement****

Credit card fraud is a critical concern for financial institutions and their customers. Early detection of fraudulent transactions is essential to mitigate financial loss and enhance security measures. This project focuses on developing a supervised learning model to predict fraudulent transactions using historical transaction data.

## ****Data Source****

### Dataset Details

* **Dataset Name:** Credit Card Fraud Detection Dataset
* **Dataset Size:**
  + **Rows:** 284,807 transactions
  + **Features:** 30 anonymized features and 1 target variable
* **Target Variable:**
  + **1:** Fraudulent transaction
  + **0:** Legitimate transaction

### Feature Set

The dataset includes:

* **Transaction Amount**
* **Time of Transaction**
* **Principal components** derived from PCA (Features V1–V28)

## ****Evaluation Metrics****

To handle the class imbalance and assess model performance, the following evaluation metrics will be used:

1. **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):**  
   Measures the model's ability to distinguish between fraudulent and legitimate transactions.
2. **Precision, Recall, and F1-Score:**  
   Key metrics to evaluate the performance on the minority class (fraudulent transactions).
3. **Confusion Matrix:**  
   Provides detailed insights into prediction performance by displaying true positives, false positives, true negatives, and false negatives.
4. **Stratified k-Fold Cross-Validation:**  
   Ensures robust validation across the imbalanced dataset.

## ****Tools and Technologies****

The project is developed in **Python**, implemented in a **Jupyter Notebook** environment.

### Libraries Used:

* **Scikit-learn:** Machine learning modeling and evaluation.
* **Pandas/NumPy:** Data manipulation and analysis.
* **Matplotlib:** Visualization of trends and insights.
* **Statsmodels:** Statistical analysis.

## ****Expected Outcomes****

**Machine Learning Model:**  
A supervised learning model capable of accurately detecting fraudulent transactions with high **AUC-ROC (> 0.85)**.

**Feature Importance Insights:**  
Detailed analysis of the most critical features influencing the detection of fraudulent transactions.

**Reusable Framework:**  
A structured framework to analyze financial transaction datasets and identify anomalies effectively.

## ****Conclusion****

This project addresses the pressing issue of credit card fraud by leveraging advanced machine learning techniques. The model and framework developed will provide financial institutions with a robust tool to enhance security and safeguard customers from fraudulent activities.

Code

import pandas as pd

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

import plotly.graph\_objs as go

import plotly.figure\_factory as ff

from plotly import tools

from plotly.offline import download\_plotlyjs, init\_notebook\_mode, plot, iplot

init\_notebook\_mode(connected=True)

import gc

from datetime import datetime

from sklearn.model\_selection import train\_test\_split

from sklearn.model\_selection import KFold

from sklearn.metrics import roc\_auc\_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import AdaBoostClassifier

from catboost import CatBoostClassifier

from sklearn import svm

import lightgbm as lgb

from lightgbm import LGBMClassifier

import xgboost as xgb

pd.set\_option('display.max\_columns', 100)

RFC\_METRIC = 'gini'  #metric used for RandomForrestClassifier

NUM\_ESTIMATORS = 100 #number of estimators used for RandomForrestClassifier

NO\_JOBS = 4 #number of parallel jobs used for RandomForrestClassifier

#TRAIN/VALIDATION/TEST SPLIT

#VALIDATION

VALID\_SIZE = 0.20 # simple validation using train\_test\_split

TEST\_SIZE = 0.20 # test size using\_train\_test\_split

#CROSS-VALIDATION

NUMBER\_KFOLDS = 5 #number of KFolds for cross-validation

RANDOM\_STATE = 2018

MAX\_ROUNDS = 1000 #lgb iterations

EARLY\_STOP = 50 #lgb early stop

OPT\_ROUNDS = 1000  #To be adjusted based on best validation rounds

VERBOSE\_EVAL = 50 #Print out metric result

IS\_LOCAL = False

import os

if(IS\_LOCAL):

    PATH="/Users/shyamkumar/Downloads/creditcard.csv"

else:

    PATH="/Users/shyamkumar/Downloads"

print(os.listdir(PATH))