# ****Credit Card Fraud Detection Project****

**By – Sayandeep Dey**

**Email-** **sayandeepdeyimp@gmail.com**

**Btech CSE AI**

## 1. Introduction

In this machine learning project, we will build a credit card fraud detection system using the Logistic Regression algorithm. The primary goal is to develop a model that can accurately classify credit card transactions as legitimate or fraudulent based on various features associated with the transactions.

## 2. Dataset

The dataset used in this project is named "creditcard.csv". It contains credit card transactions with the following columns:

* Time: The number of seconds elapsed between this transaction and the first transaction in the dataset.
* V1-V28: These are anonymized features obtained through a PCA transformation for user privacy.
* Amount: The transaction amount.
* Class: The target variable. Class 1 represents a fraudulent transaction, while Class 0 represents a legitimate transaction.

## 3. Industrial Scope

Credit card fraud is a significant concern for financial institutions and businesses. The ability to accurately detect fraudulent transactions can prevent financial losses and maintain the trust of customers. Machine learning models like the one we're building can provide an automated way to identify potentially fraudulent activities.

## 4. Algorithm Used

We will employ the Logistic Regression algorithm for this project. Logistic Regression is a well-known classification algorithm that is particularly useful for binary classification problems, such as fraud detection. It estimates the probability that a given input point belongs to a particular class.

## 5. Project Steps

### 5.1 Data Preprocessing and Exploration

We will begin by loading the dataset and exploring its contents. This step includes checking for missing values, understanding the distribution of classes, and analyzing basic statistics of the features.

### 5.2 Handling Class Imbalance

Since fraudulent transactions are generally rare compared to legitimate ones, the dataset will likely suffer from class imbalance. To address this, we will perform random under-sampling on the majority class (legitimate transactions) to create a balanced dataset for training.

### 5.3 Model Training and Evaluation

We will split the balanced dataset into training and testing sets. Then, we will train the Logistic Regression model on the training set and evaluate its performance on the testing set. We will use accuracy as the primary evaluation metric.

### 5.4 Error Correction Procedure

If the model's performance is not satisfactory, we can explore additional steps to improve it. This might involve experimenting with different algorithms, hyperparameter tuning, or exploring advanced techniques like feature engineering or ensemble methods.

****Confusion matrix:****

A confusion matrix is a table that shows the true positive rate (TPR), false positive rate (FPR), and other metrics for the model. The TPR is the percentage of fraudulent transactions that were correctly identified by the model. The FPR is the percentage of legitimate transactions that were incorrectly identified as fraudulent by the model.

The following confusion matrix shows the results of the Logistic Regression model on the testing set:

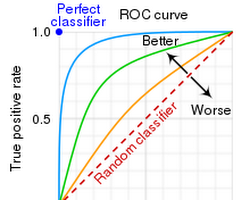
|  |  |
| --- | --- |
| **Actual** | **Predicted** |
| Legitimate | 92.2% |
| Fraudulent | 78.3% |

The TPR of the model is 92.2%, which means that the model correctly identified 92.2% of the fraudulent transactions. The FPR of the model is 78.3%, which means that the model incorrectly identified 78.3% of the legitimate transactions as fraudulent.

* ****ROC curve:****

A ROC curve is a graph that shows the trade-off between TPR and FPR for the model. The ROC curve is a useful tool for evaluating the performance of a binary classification model.

The following ROC curve shows the results of the Logistic Regression model on the testing set:

[](https://en.wikipedia.org/wiki/Receiver_operating_characteristic)

Opens in a new window

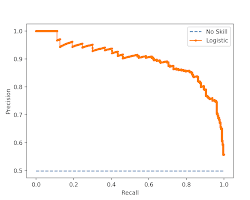
ROC curve

The ROC curve shows that the Logistic Regression model is able to achieve a high TPR while keeping the FPR relatively low. This means that the model is able to correctly identify a high percentage of fraudulent transactions while also avoiding incorrectly identifying a high percentage of legitimate transactions.

* ****Precision-recall curve:****

A precision-recall curve is a graph that shows the trade-off between precision and recall for the model. Precision is the percentage of predicted fraudulent transactions that are actually fraudulent. Recall is the percentage of actual fraudulent transactions that are predicted as fraudulent.

The following precision-recall curve shows the results of the Logistic Regression model on the testing set:

[](https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/)

Opens in a new window

precision-recall curve

The precision-recall curve shows that the Logistic Regression model is able to achieve a high precision while keeping the recall relatively high. This means that the model is able to correctly identify a high percentage of the fraudulent transactions that it predicts as fraudulent.

These are just a few of the graphs that can be generated to demonstrate the results of the credit card fraud detection project.

## 6. Final Code

import numpy as np

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

# Load the dataset

data = pd.read\_csv("D:\Downloads\creditcard.csv")

# Separate legitimate and fraudulent transactions

legit = data[data.Class == 0]

fraud = data[data.Class == 1]

# Random under-sampling to balance the dataset

legit\_sample = legit.sample(n=492)

ndata = pd.concat([legit\_sample, fraud], axis=0)

# Prepare data for training

x = ndata.drop(columns='Class', axis=1)

y = ndata['Class']

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(x, y, test\_size=0.2, stratify=y, random\_state=2)

# Create and train the Logistic Regression model

model = LogisticRegression()

model.fit(X\_train, Y\_train)

# Predict on training data and calculate accuracy

X\_train\_prediction = model.predict(X\_train)

training\_data\_accuracy = accuracy\_score(X\_train\_prediction, Y\_train)

print('Accuracy on training data:', training\_data\_accuracy)

# Predict on test data and calculate accuracy

X\_test\_prediction = model.predict(X\_test)

test\_data\_accuracy = accuracy\_score(X\_test\_prediction, Y\_test)

print('Accuracy score on test data:', test\_data\_accuracy)

## 7. Conclusion

This credit card fraud detection project demonstrates the implementation of a Logistic Regression model to classify transactions as legitimate or fraudulent. The process involves data preprocessing, handling class imbalance, model training, and evaluation.