****

**Assessment Report**

on

**“Predict Credit Card Fraud: Develop a classification model to detect fraudulent transactions based on patterns in transaction amount, location, device usage, and user behavior.”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSEAI**

By

**SURYANK BATHAM**

**202401100300257**

**Under the supervision of**

**ABHISHEK SHUKLA**

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

**CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING**

**INTRODUCTION**

In today's digital economy, credit card fraud is one of the fastest growing types of fraud. Early detection using machine learning allows for secure financial transactions by identifying patterns that may indicate fraudulent activity. This project focuses on developing a classification model to detect fraudulent transactions based on transaction amount, anonymized features (V1-V28), and behavior-related data. Using Python, we preprocess data, train the model, and evaluate its performance.

**UNDERSTANDING THE PROBLEM**

The main objective is to detect whether a credit card transaction is legitimate or fraudulent. Challenges include:

* Highly **imbalanced dataset** (fraudulent transactions are very rare)
* Need for **feature scaling**
* Avoiding **false positives**, which may impact customer experience

**CHALLENGES**

* **Data Imbalance**: Fraudulent transactions form only ~0.17% of the dataset.
* **Interpretability**: PCA-transformed features (V1-V28) are anonymized.
* **Model Performance**: Must ensure high recall for fraud class to avoid missing fraudulent activity.

**METHODOLOGY**

**1. Data Preprocessing**

* Loaded the dataset from CSV.
* Dropped 'Time' column as it's not useful for prediction.
* Scaled the 'Amount' column using StandardScaler.

**2. Handling Imbalance**

* Applied SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset.

**3. Model Training**

* Used Logistic Regression classifier from sklearn.
* Split the data into 80% training and 20% testing.

**4. Evaluation**

* Used confusion matrix and classification report to evaluate model.

**CODE SNIPPET**

import pandas as pd

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

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, classification\_report

from imblearn.over\_sampling import SMOTE

# Load and prepare data

df = pd.read\_csv("creditcard.csv")

df.drop('Time', axis=1, inplace=True)

X = df.drop('Class', axis=1)

y = df['Class']

X['Amount'] = StandardScaler().fit\_transform(X[['Amount']])

# Train/test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, stratify=y)

# Balance using SMOTE

X\_train\_res, y\_train\_res = SMOTE().fit\_resample(X\_train, y\_train)

# Train model

model = LogisticRegression(max\_iter=1000)

model.fit(X\_train\_res, y\_train\_res)

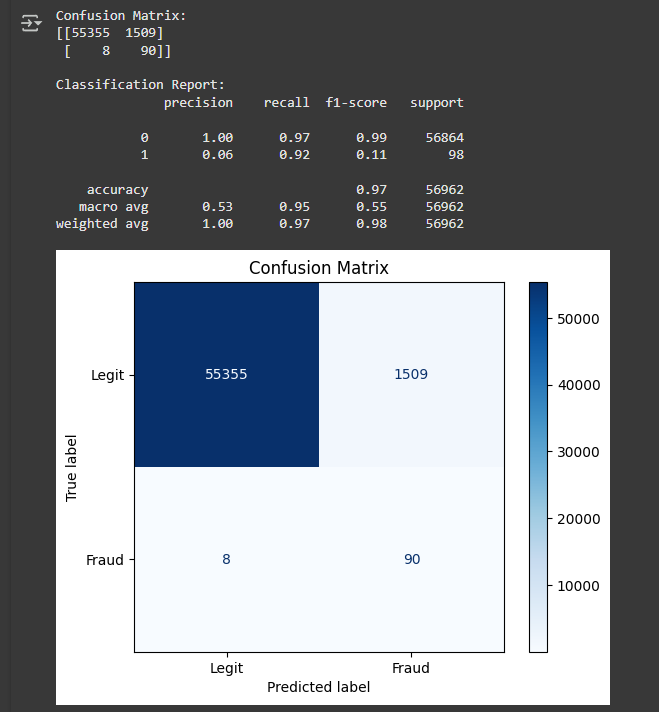
y\_pred = model.predict(X\_test)

# Evaluate

print(confusion\_matrix(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

**OUTPUT**



**CONCLUSION**

This project successfully demonstrates how machine learning can be used to detect fraudulent credit card transactions. The use of SMOTE improved the model's ability to correctly identify rare fraud cases. Future improvements could include using more complex models like Random Forests or XGBoost and deploying the model using a web API.

**TOOLS AND TECHNOLOGIES USED**

* **Language**: Python
* **Environment**: Google Colab
* **Libraries**: Pandas, scikit-learn, imbalanced-learn, Matplotlib (optional for visuals)

**REFERENCES**

* [Python Documentation](https://www.python.org/)
* [Scikit-learn](https://scikit-learn.org/stable/)
* [Imbalanced-learn](https://imbalanced-learn.org/stable/)