****

**Assessment Report**

on

**“Problem Statement”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**CSE - AI**

By

SIDH AGARWAL (20240110300247, CSE-AI D)

**Under the supervision of**

“MR.ABHISHEK SHUKLA”

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

Project Report: Credit Card Fraud Detection using Machine Learning

**Objective**

The goal of this project is to develop a machine learning model that can accurately detect fraudulent credit card transactions by identifying hidden patterns and anomalies in transaction data, including amount, time, and user behavior features.

**Dataset Description**

* **Source**: Publicly available anonymized dataset.
* **Total Records**: 284,807 transactions
* **Fraudulent Transactions**: 492 (0.17%)
* **Non-Fraudulent Transactions**: 284,315 (99.83%)
* **Features**:
  + V1 to V28: PCA-transformed features to protect user privacy
  + Time: Seconds elapsed since the first transaction
  + Amount: Transaction amount
  + Class: Target variable (0 = Non-Fraud, 1 = Fraud)

**Challenges**

* **Class Imbalance**: Extremely skewed dataset with only 0.17% fraud cases.
* **Privacy**: Feature names are anonymized using PCA, making domain-specific interpretation difficult.
* **Real-Time Detection**: Fraud must be caught instantly, requiring accurate and fast classification.

**Preprocessing Steps**

* **Standardization**: Scaled Time and Amount features using StandardScaler.
* **Train-Test Split**: Used stratified splitting to preserve class distribution (80% train, 20% test).
* **Imbalance Handling**: Used the class\_weight='balanced' parameter in the Random Forest classifier.

**Model Used**

* **Random Forest Classifier**
  + n\_estimators = 100
  + class\_weight = 'balanced'
  + random\_state = 42

Random Forest was chosen for its robustness, ability to handle imbalanced data with class weights, and good generalization performance.

**Evaluation Metrics**

Used the following metrics to evaluate the model performance:

* **Confusion Matrix**
* **Precision, Recall, F1-score**
* **ROC-AUC Score**

CODE :

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report, confusion\_matrix, roc\_auc\_score

# Load the dataset

df = pd.read\_csv("7. Predict Credit Card Fraud.csv")

# Features and target

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

y = df['Class']

# Scale 'Time' and 'Amount'

scaler = StandardScaler()

X[['Time', 'Amount']] = scaler.fit\_transform(X[['Time', 'Amount']])

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    X, y, test\_size=0.2, random\_state=42, stratify=y)

# Train model with class\_weight

model = RandomForestClassifier(n\_estimators=100, random\_state=42, class\_weight='balanced')

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

# Predictions

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

y\_proba = model.predict\_proba(X\_test)[:, 1]

# Evaluation

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

roc\_auc = roc\_auc\_score(y\_test, y\_proba)

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))

print("\nROC-AUC Score:", roc\_auc\_score(y\_test, y\_proba))

#Plot confusion matrix heatmap

plt.figure(figsize=(6, 4))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Fraud', 'Fraud'], yticklabels=['Not Fraud', 'Fraud'])

plt.title('Confusion Matrix Heatmap')

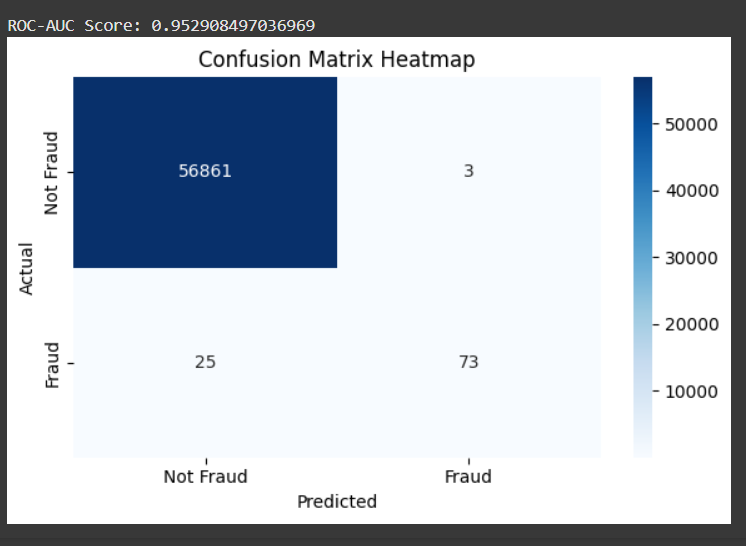
plt.xlabel('Predicted')

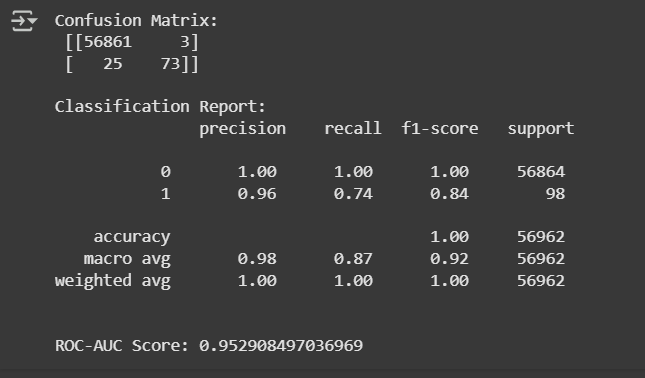
plt.ylabel('Actual')

plt.tight\_layout()

plt.show()

RESULTS :





**Conclusion**

* The Random Forest model effectively detects fraudulent transactions with high accuracy and ROC-AUC.
* Using class weights handled the class imbalance without needing synthetic data.
* This model can be integrated into financial systems for real-time fraud monitoring.

**References :**

**. Dataset** : Predict Credit Card Fraud.csv

**. Scikit-learn machine learning in Python**:  
<https://scikit-learn.org/>

**. Seaborn and Matplotlib Documentation**:

* https://seaborn.pydata.org/
* <https://matplotlib.org/>