A

Assessment Report

on

“Predicting Credit Card Fraud”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Introduction To AI

By

Preeti Tiwari (202401100400147)

Under the supervision of

“Mr. Abhishek Shukla”

KIET Group of Institutions, Ghaziabad

Affiliated to

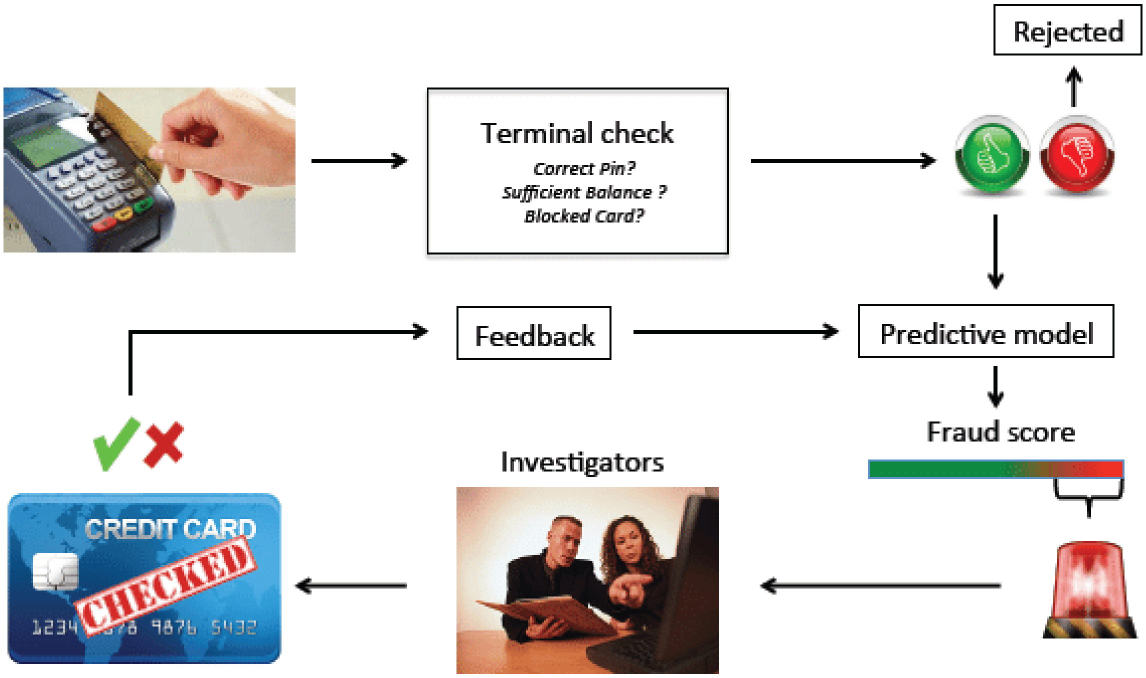
Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

Introduction

Detecting credit card fraud is a critical challenge in the financial sector, where billions are lost every year due to unauthorized transactions. The goal of this project is to build a machine learning classification model that accurately identifies fraudulent transactions by analysing patterns in user behaviour, transaction amount, location, and device usage. Since fraudulent activity is rare and often disguised among normal transactions, this problem involves handling highly imbalanced data and learning subtle anomalies. An effective solution helps banks and payment systems prevent financial loss, improve trust, and enhance security for customers by flagging suspicious transactions in real time.



Methodology

To detect fraudulent credit card transactions, we followed a structured machine learning pipeline:

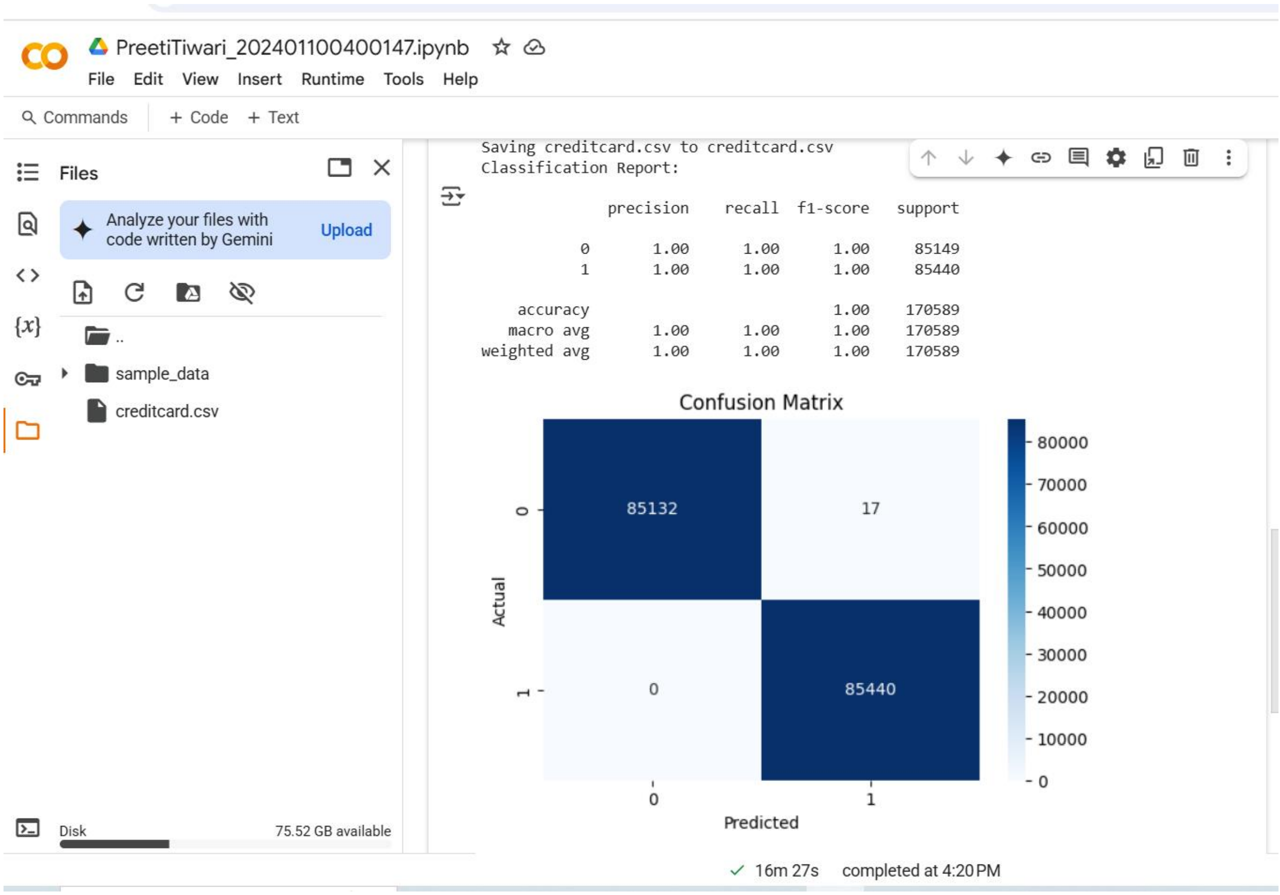
1. **Data Collection & Upload:**
   The dataset containing transaction records was uploaded and loaded into Google Colab.
2. **Data Preprocessing:**
   - Handled missing values by dropping incomplete rows.
   - Encoded categorical features using Label Encoding to convert them into numeric format.
   - Normalized or scaled features (if needed) to maintain consistency.
3. **Feature & Target Selection:**
   Separated independent features (e.g., amount, location, device, behavior) and the target variable (`Fraud` or `Not Fraud`).
4. **Data Splitting:**
   Split the dataset into training and testing sets using an 80-20 ratio, preserving class distribution using stratification.
5. **Model Building:**
   Trained a **Random Forest Classifier**, a robust ensemble algorithm known for handling imbalanced datasets and high-dimensional features.
6. **Model Evaluation:**
   Evaluated the model using **accuracy**, **precision**, **recall**, and a **confusion matrix heatmap** to visualize correct and incorrect classifications.
7. **Visualization:**
   Visual tools such as confusion matrix plots were used to better understand model performance and detect any misclassification trends.

Code Used

```
from google.colab import files
uploaded = files.upload()
import pandas as pd

df = pd.read_csv('creditcard.csv')  # If the file name was different,
update it!
df.head()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from imblearn.over_sampling import SMOTE

# Load dataset


# Feature Scaling for 'Amount' and 'Time'
scaler = StandardScaler()
df['scaled_amount'] = scaler.fit_transform(df[['Amount']])
df['scaled_time'] = scaler.fit_transform(df[['Time']])
df.drop(['Amount', 'Time'], axis=1, inplace=True)

# Rearranging columns
scaled_amount = df['scaled_amount']
scaled_time = df['scaled_time']
df.drop(['scaled_amount', 'scaled_time'], axis=1, inplace=True)
df.insert(0, 'scaled_amount', scaled_amount)
df.insert(1, 'scaled_time', scaled_time)

# Splitting features and target
X = df.drop('Class', axis=1)
y = df['Class']

# SMOTE for balancing the classes
sm = SMOTE(random_state=42)
X_res, y_res = sm.fit_resample(X, y)
```

```
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X_res, y_res,
test_size=0.3, random_state=42)

# Model Training
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Prediction and Evaluation
y_pred = model.predict(X_test)

print("Classification Report:\n")
print(classification_report(y_test, y_pred))

# Confusion Matrix
plt.figure(figsize=(6,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d',
cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Output



```
Saving creditcard.csv to creditcard.csv
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85149
           1       1.00      1.00      1.00     85440

    accuracy                           1.00    170589
   macro avg       1.00      1.00      1.00    170589
weighted avg       1.00      1.00      1.00    170589
```

Refrences

- Used ChatGPT for basic Understanding.
- Dataset of credit card prediction taken from Kaggle.
- Used image from google images.
- Used google colaboratory for running the program.