Credit Card Fraud Detection Using Machine Learning

Student Name: Parth Soni

Program: AI & ML with Python

Project Type: Major

Dataset Source: Kaggle – Credit Card Fraud Detection Dataset

1. Introduction

Credit card fraud is one of the most significant challenges in the financial world. Fraudulent transactions are rare but highly damaging, making early detection extremely important. In this project, machine learning techniques are applied to classify transactions as normal or fraudulent based on historical credit card transaction data.

2. Problem Statement

The objective of this project is to build a machine learning model capable of detecting fraudulent credit card transactions using anonymized numerical transaction features.

3. Dataset Description

The dataset contains 284,807 transactions, out of which only 492 are fraudulent, making it highly imbalanced. Features V1–V28 are PCA-transformed components to anonymize sensitive information. The 'Class' column is the target variable: 0 for normal transactions and 1 for fraud.

4. Methodology

• Loaded and explored the dataset in Google Colab.

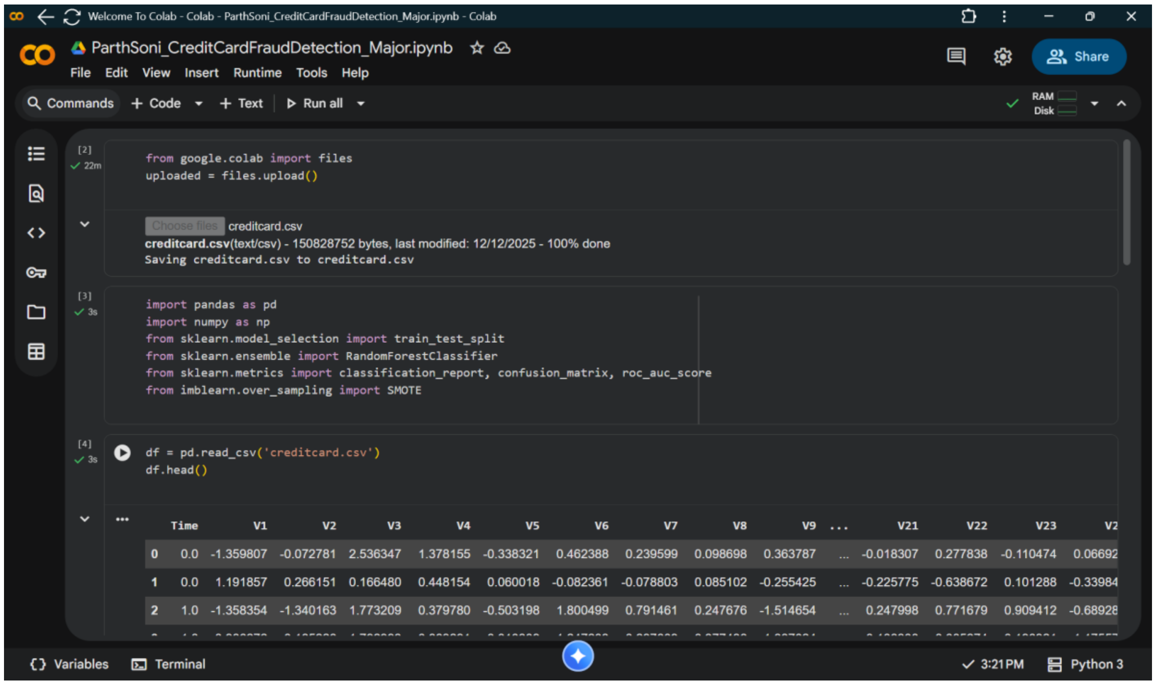
• Applied SMOTE (Synthetic Minority Oversampling Technique) to balance the dataset.

• Split the dataset into training and testing sets using an 80/20 ratio.

• Trained a Logistic Regression model with balanced class weights for efficient fraud detection.

• Evaluated the model using classification report, confusion matrix, and ROC AUC score.

Below is the screenshot of the model training and evaluation code:


Welcome To Colab - Colab - ParthSoni_CreditCardFraudDetection_Major.ipynb - Colab
ParthSoni_CreditCardFraudDetection_Major.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk
Variables Terminal
3:21 PM Python 3

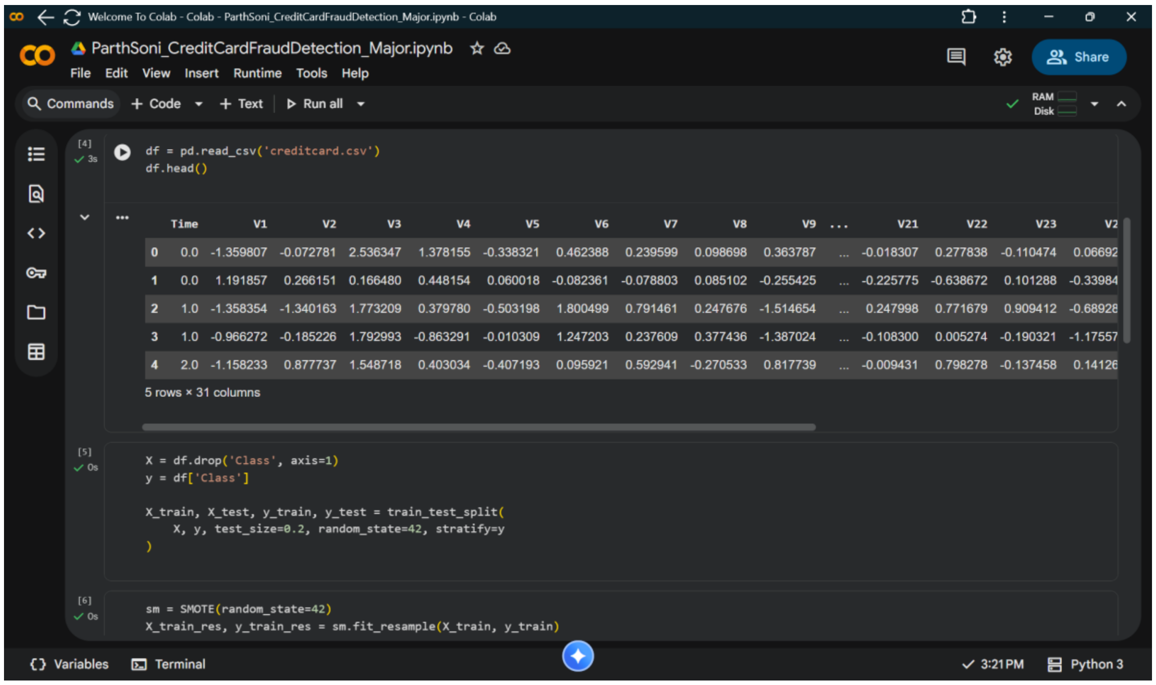
```
[2]
from google.colab import files
uploaded = files.upload()
```

```
Choose files creditcard.csv
creditcard.csv(text/csv) - 150828752 bytes, last modified: 12/12/2025 - 100% done
Saving creditcard.csv to creditcard.csv
```

```
[3]
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
from imblearn.over_sampling import SMOTE
```

```
[4]
df = pd.read_csv('creditcard.csv')
df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.06692
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.33984
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.68928


Welcome To Colab - Colab - ParthSoni_CreditCardFraudDetection_Major.ipynb - Colab
ParthSoni_CreditCardFraudDetection_Major.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk
Variables Terminal
3:21 PM Python 3

```
[4]
df = pd.read_csv('creditcard.csv')
df.head()
```

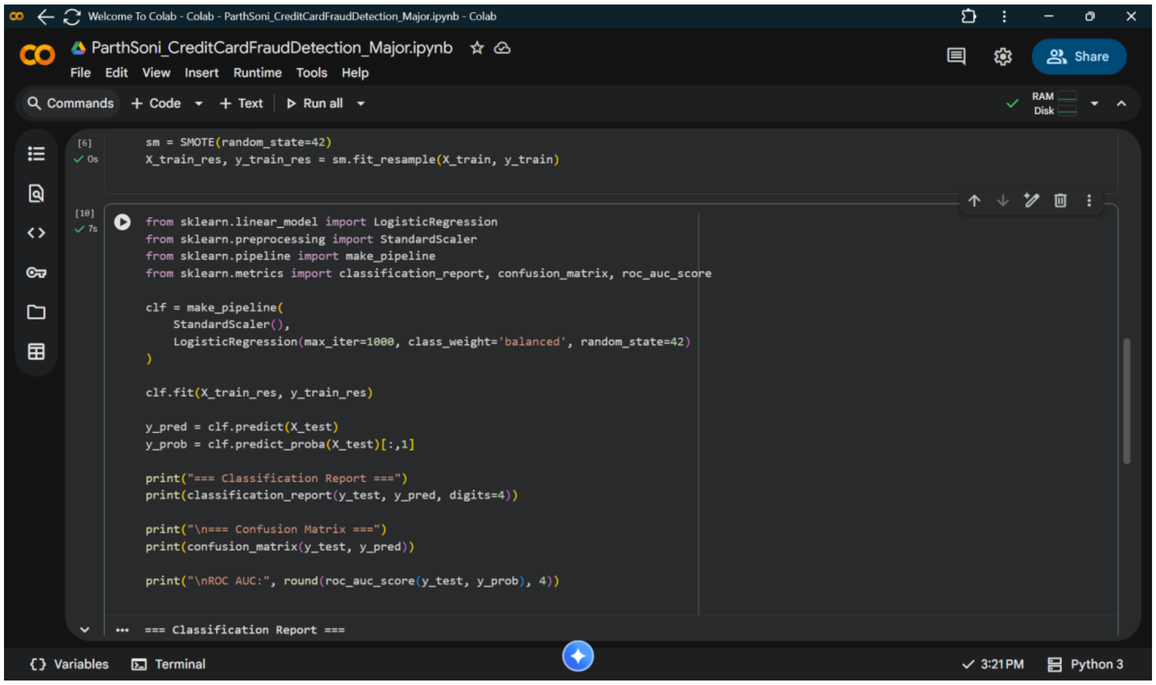
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.06692
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.33984
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.68928
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.17557
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.14126

5 rows × 31 columns

```
[5]
X = df.drop('Class', axis=1)
y = df['Class']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

```
[6]
sm = SMOTE(random_state=42)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
```


Welcome To Colab - Colab - ParthSoni_CreditCardFraudDetection_Major.ipynb - Colab
ParthSoni_CreditCardFraudDetection_Major.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk
Variables Terminal
3:21 PM Python 3

```
[6]
sm = SMOTE(random_state=42)
X_train_res, y_train_res = sm.fit_resample(X_train, y_train)
```

```
[10]
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score

clf = make_pipeline(
    StandardScaler(),
    LogisticRegression(max_iter=1000, class_weight='balanced', random_state=42)
)

clf.fit(X_train_res, y_train_res)

y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)[:,1]

print("=== Classification Report ===")
print(classification_report(y_test, y_pred, digits=4))

print("\n=== Confusion Matrix ===")
print(confusion_matrix(y_test, y_pred))

print("\nROC AUC:", round(roc_auc_score(y_test, y_prob), 4))
```

```
=== Classification Report ===
```

5. Results and Discussion

The following results were obtained after evaluating the model:

Classification Report:

Class 0 → Precision: 0.9998 | Recall: 0.9900 | F1-score: 0.9949 | Support: 56864

Class 1 → Precision: 0.1341 | Recall: 0.8980 | F1-score: 0.2334 | Support: 98

Accuracy: 0.9899

Macro Avg Recall: 0.9440

Weighted Avg Recall: 0.9899

Confusion Matrix:

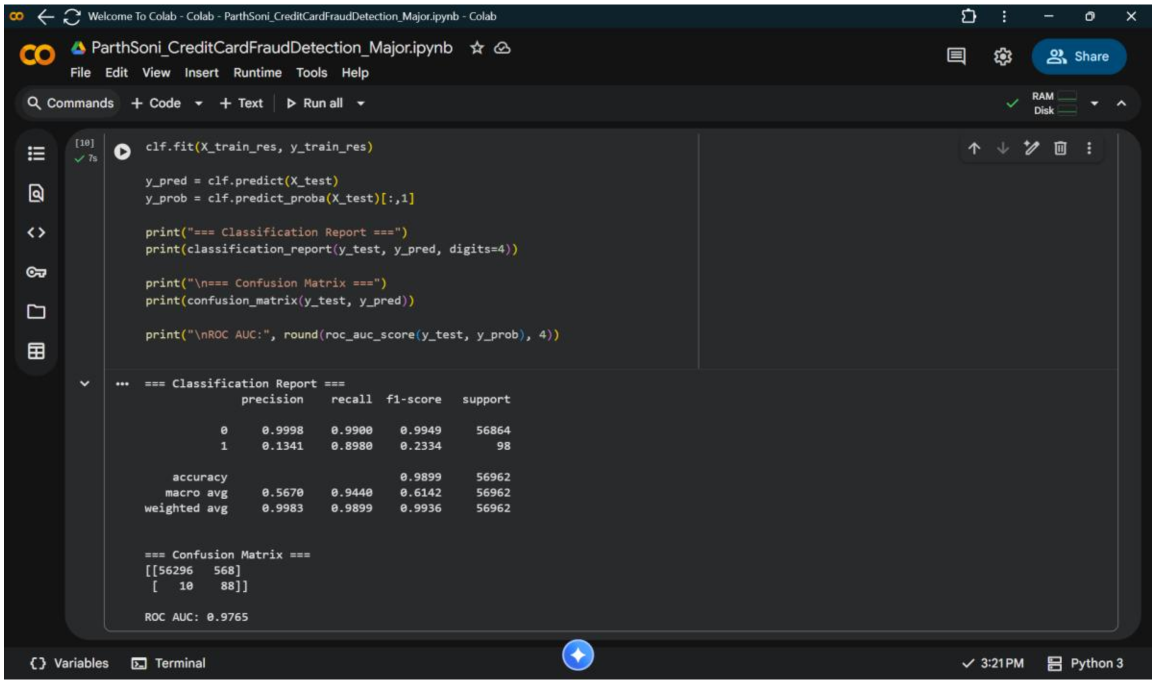
```
[[56296   568]
 [   10    88]]
```

ROC AUC Score: 0.9765

The model performs exceptionally well in detecting fraudulent transactions. A recall of

0.8980 for the minority class (fraud) is highly impressive, indicating that the model successfully captures most fraudulent cases. The ROC AUC score of 0.9765 demonstrates excellent discriminatory power.

Below is the screenshot of the model output:


Welcome To Colab - Colab - ParthSoni_CreditCardFraudDetection_Major.ipynb - Colab
ParthSoni_CreditCardFraudDetection_Major.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Variables Terminal 3:21PM Python 3

```
clf.fit(X_train_res, y_train_res)

y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)[:,1]

print("=== Classification Report ===")
print(classification_report(y_test, y_pred, digits=4))

print("\n=== Confusion Matrix ===")
print(confusion_matrix(y_test, y_pred))

print("\nROC AUC:", round(roc_auc_score(y_test, y_prob), 4))
```

```
=== Classification Report ===
              precision    recall  f1-score   support

           0     0.9998    0.9900    0.9949     56864
           1     0.1341    0.8980    0.2334        98

    accuracy                         0.9899     56962
   macro avg     0.5670    0.9440    0.6142     56962
weighted avg     0.9983    0.9899    0.9936     56962


=== Confusion Matrix ===
[[56296   568]
 [   10    88]]

ROC AUC: 0.9765
```

6. Conclusion

This project demonstrates that machine learning techniques, combined with effective data balancing methods such as SMOTE, can accurately detect fraudulent credit card transactions. The model achieved high recall for fraud detection and a strong ROC AUC score, making it reliable for real-world financial fraud analysis.

7. References

• Kaggle Credit Card Fraud Detection Dataset

• Scikit-learn Documentation

• Imbalanced-Learn Library

• Google Colab Platform

• Pandas & NumPy Libraries