Credit Card Fraud Detection – Detailed Project Report

Project Title: Credit Card Fraud Detection using Random Forest
Dataset: Kaggle Credit Card Fraud Dataset

Abstract:

This project focuses on detecting fraudulent credit card transactions using machine learning. We used a highly imbalanced dataset from 2013, which contains only 492 frauds out of 284,807 transactions. Our objective was to implement effective preprocessing, anomaly detection, and classification techniques — with a focus on the Random Forest classifier — to build a robust fraud detection system.

Problem Statement:

Fraudulent financial transactions are on the rise globally, resulting in billions in losses annually. Detecting such anomalies in real-time is critical for banks and payment gateways. However, the low occurrence of fraud in the dataset (less than 0.2%) presents challenges in modeling and evaluation. This project aims to build a model that:

- Accurately detects fraud with minimal false positives
- Handles imbalanced data effectively
- Can be deployed for predictions on new datasets

Tools & Technologies Used:

- **Language**: Python
- **Development Platform**: Google Colab
- **Libraries**:
  - pandas, numpy – Data manipulation
  - matplotlib, seaborn – Visualization
  - scikit-learn – ML models & metrics
  - xgboost (initial experiments)
  - pickle – Model saving
  - imbalanced-learn (optional: SMOTE, etc.)

Project Workflow:

1. Importing Libraries:
Standard Python libraries for data handling, modeling, and evaluation.

2. Uploading Dataset:
Used `files.upload()` in Colab to load the dataset.

3. Preprocessing:

- Dropped irrelevant features like `Time`
- Normalized `Amount` using `StandardScaler`
- Checked and visualized class imbalance

4. Splitting the Dataset:
Used a 70/30 train-test split with stratified sampling.

5. Model Building – Random Forest:
Trained a Random Forest model with:

```
Python

CopyEdit

from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(
    n_estimators=100,
    max_depth=8,
    class_weight='balanced',
    random_state=42
)
```

6. Model Evaluation:

- Confusion Matrix
- ROC Curve with AUC score

7. Saving the Model:
Used `pickle` to save the model and feature columns.

8. Prediction on New Data:

- Allowed prediction on new uploaded data
- Provided fraud probability for each transaction

9. Results Download:
Saved and downloaded the output prediction as `.csv`.

✅ Innovations / Value-Additions:

- Manual transaction prediction via user input
- Prediction results available in downloadable CSV format
- ROC curve-based evaluation for better performance validation
- No external dependencies like Streamlit or Flask for small-scale execution

Anomaly Detection:

Also tested methods like:

- Isolation Forest
- Local Outlier Factor

They helped find outlier transactions potentially fraudulent even before classification.

Deployment Notes:

Due to errors in running Streamlit on Colab, the app interface was replaced with:

- Manual Input through code
- Prediction-based upload + download CSV via Colab.

References:

- Kaggle Dataset: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- scikit-learn Docs: https://scikit-learn.org
- Random Forest Theory: https://towardsdatascience.com/random-forest-explained-9f2958e8ff16
- Imbalanced Data: https://imbalanced-learn.org

Conclusion:—

This project successfully implements a reliable credit card fraud detection pipeline using the Random Forest algorithm. It demonstrates real-world handling of imbalanced data, model evaluation using ROC-AUC, and prediction generation — all integrated in a beginner-friendly Google Colab environment.

By- Samruddhi Patil