Fraud Detection in Credit Card Transactions Using ML

1. Introduction

The project is part of an internship at Elevate Labs to build an intelligent fraud detection system that classifies whether a credit card transaction is **fraudulent or legitimate**, using a real-world, imbalanced dataset.The increase in online transactions and digital payments has also resulted in a surge in fraudulent activities involving credit cards. Detecting fraud in real time is a critical challenge, given the massive volume of transactions processed daily. This project aims to address this challenge by building a machine learning-based fraud detection system that can accurately classify whether a transaction is fraudulent or legitimate.

The dataset used is derived from real-world credit card transactions, with a significant class imbalance due to the rarity of fraud cases. Therefore, this project combines both unsupervised and supervised learning techniques to build a robust detection pipeline, ensuring minimal false positives while capturing actual fraud cases.

---

2. Abstract

This project focuses on identifying fraudulent credit card transactions using machine learning and anomaly detection techniques. The dataset used from Kaggle (Source: [Kaggle - Credit Card Fraud Detection](https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud))contains 284,807 transactions, with only 492 labeled as fraudulent — highlighting a significant class imbalance. We first performed data preprocessing, including feature scaling and balancing techniques, to prepare the data for modeling. For anomaly detection, unsupervised models such as Isolation Forest and Local Outlier Factor (LOF) were implemented to flag outlier patterns. These models help uncover rare fraudulent transactions based on feature behavior. After initial anomaly detection, we trained a supervised XGBoost classifier using the processed data. Model performance was evaluated using metrics like confusion matrix, precision, recall, and the ROC-AUC score. The model was then deployed as a Streamlit web app for real-time fraud prediction, where users can input transaction details and receive immediate classification feedback.

---

3. Tools Used

- **Python** – For overall programming and data handling
- **Pandas & NumPy** – For data manipulation and preprocessing
- **Scikit-learn** – For modeling, evaluation metrics, and anomaly detection
- **XGBoost** – As the main classifier for supervised learning
- **Matplotlib** – For plotting the ROC curve and visual evaluation
- **Streamlit** – For building a simple, interactive web application
- **Joblib** – For model serialization

---

4. Steps Involved in Building the Project

1. **Data Acquisition**
    - Downloaded the credit card dataset from Kaggle
    - Explored the structure and distribution of fraud vs. legitimate cases
2. **Data Preprocessing**
    - Scaled the ‘Amount’ feature using StandardScaler
    - Dropped irrelevant features (e.g., Time)
    - Handled class imbalance using undersampling techniques
3. **Anomaly Detection**
    - Implemented **Isolation Forest** and **Local Outlier Factor**
    - Identified outliers that may indicate fraudulent activity
4. **Model Training**
    - Used **XGBoost Classifier** to train on labeled data
    - Split data into training and test sets using stratified sampling
    - Evaluated using confusion matrix and classification report
5. **Model Evaluation**
    - Plotted the ROC Curve and calculated AUC Score
    - Achieved high accuracy and recall for fraud detection
6. **Deployment**
    - Saved the trained model using joblib
    - Successfully tested the model locally and deployed on Streamlit Cloud and it returned the local host as http://192.168.0.9:8501 in the cloud.
    - Enabled users to enter V1–V28 features and amount for classification

---

5. Conclusion

This project successfully demonstrates how machine learning can be applied to detect credit card fraud by leveraging both anomaly detection and supervised learning techniques. The combination of Isolation Forest and XGBoost provides strong predictive power while minimizing false positives.The Streamlit web app adds usability by allowing real-time predictions based on user input. While the model performs well, further improvements can be made by incorporating advanced balancing techniques.This project has improved my understanding of fraud detection challenges, handling imbalanced datasets, and deploying practical ML solutions.The outputs and delivaribles deployed here are as follows:

📊 Outputs

- Precision, Recall, F1-score
- ROC-AUC Score

- Confusion Matrix
- Interactive UI for live predictions

Deliverables

- CreditCardFraudDetection.ipynb – Notebook with complete ML pipeline
- app.py – Streamlit frontend
- model.pkl – Saved model
- fraud_report.pdf – Report of the project
- If deployed on streamlit cloud app

https://credit-card-fraud-detection05.streamlit.app click to open app

---

🧑‍💻 **Developed By**

R SIRI LALITHA BHUVANESWARI
Intern at Elevate Labs