Project Report: Online Payment Fraud Detection

Problem Statement

In the era of rapid digital transactions, fraud detection has become crucial for both financial institutions and users. Fraudulent transactions cause not just monetary loss but also undermine trust in digital systems. This project aims to build a reliable, machine-learning powered tool to identify fraudulent transactions in real time.

Project Overview

We developed a machine learning model that analyzes various transaction-related features and predicts whether a transaction is fraudulent or legitimate. The solution is deployed using a Streamlit web app, allowing users to interactively test the model with custom input values.

Dataset Summary

The dataset (Fraud Detection Dataset.csv) includes the following key features:

- Transaction_Amount
- Time_of_Transaction
- Previous_Fraudulent_Transactions
- Account_Age
- Number_of_Transactions_Last_24H
- Transaction_Type (e.g., Bill Payment, ATM Withdrawal)
- Location
- Device_Used
- Payment_Method

Each record is labeled as:

- 1 - Fraud
- 0 - Not Fraud

The dataset was clean, but standard preprocessing steps like encoding categorical variables and scaling were applied using a pipeline.

Project Report: Online Payment Fraud Detection

Exploratory Data Analysis (EDA Highlights)

- Class imbalance was evident: fraudulent transactions are rare.
- Some payment methods and device types were more associated with fraud.
- High transaction amounts and unusual times were more suspicious.
- Past fraud history strongly influenced the likelihood of current fraud.

Model Development

We trained multiple models including:

- Decision Tree
- Random Forest
- Support Vector Machine
- Gradient Boosting

Random Forest yielded the best results with:

- F1 Score: ~0.75
- Recall: ~0.76

This made it the final choice for deployment due to its balanced performance.

We used a pipeline to:

- Encode categorical features
- Scale numeric data
- Integrate the model seamlessly into the app

Evaluation Metric

To evaluate model accuracy and robustness, we used:

- F1 Score
- Recall
- Precision

- Mean Absolute Percentage Error (MAPE)

MAPE showed how far off predictions were, on average, in percent terms-useful for interpretability but not as relevant for binary classification.

Streamlit App

The frontend is a lightweight, intuitive Streamlit app that allows users to:

- Input transaction details
- Simulate various scenarios (e.g., suspicious payment method or location)
- Get real-time predictions with a clear fraud/not fraud outcome

Legitimate transactions are flagged with a green banner.
Fraudulent ones are highlighted in red with a warning.

Challenges

- Class imbalance made training tricky-precision-recall trade-off was carefully tuned.
- Ensuring interpretability for non-technical users via the app interface.

Impact

This tool can:

- Be integrated with payment gateways to flag high-risk transactions
- Help fraud analysts prioritize review cases
- Serve as a learning prototype for building more robust security systems

Tech Stack

- Python
- Scikit-learn
- Pandas, NumPy
- Streamlit

- Pickle (for model and pipeline deployment)

Future Work

- Integrate real-time streaming data
- Implement SHAP/LIME for explainability
- Improve detection of new types of fraud patterns
- Explore deep learning models for higher accuracy