Project Report: Real-Time Anomaly Detection in Financial Transactions using Python

by Shrushti Wable

# Objective

The objective of this project is to build a lightweight, real-time anomaly detection system that can flag suspicious financial transactions. This can be useful in scenarios such as fraud detection, risk assessment, and automated transaction monitoring.

# Problem Statement

In financial systems, identifying abnormal transactions in real-time is critical to preventing fraud and minimizing financial loss. This project uses an unsupervised learning model (Isolation Forest) to detect outliers in transaction data based on amount and time.

# Tools & Technologies

- Python

- pandas, numpy

- scikit-learn (Isolation Forest)

- matplotlib

- Jupyter Notebook

# How the Solution Works

1. The notebook starts by importing necessary libraries for data processing, machine learning, and plotting.

2. It then checks for a dataset file in the 'data/' directory. If not present, it creates a simulated dataset of transaction times and amounts.

3. An Isolation Forest model is trained using the 'Amount' and 'Time' fields to identify anomalies.

4. Transactions predicted as anomalies are labeled and visualized using matplotlib.

5. These anomalies are also saved to a CSV file in the 'output/' directory for further analysis.

# Code Overview

- os.makedirs(): Ensures required folders exist before reading or writing files.

- IsolationForest(): A machine learning algorithm designed to isolate anomalies.

- model.fit\_predict(): Trains the model and assigns anomaly scores (-1 for anomalies, 1 for normal).

- matplotlib.pyplot: Used to generate a scatter plot of normal and anomalous transactions.

# Output & Observations

The model flags approximately 2% of the transactions as anomalies (based on contamination parameter).

A scatter plot visually distinguishes normal vs. anomalous transactions.

Detected anomalies are saved to a CSV file for future review.

# Conclusion

This project demonstrates a simple, fast, and effective way to detect anomalies in financial transaction data. It is designed to be run locally and uses open-source tools, making it easily deployable or extensible for real-world use cases like fraud detection pipelines.