AI Fraud Detection & Transaction Monitoring – Project Report

1. Introduction

Fraudulent financial transactions pose a major risk to banks and e-commerce platforms. Traditional rule-based systems fail to capture evolving fraud patterns, while machine learning can detect anomalies in real time. $34.8 million lost per day *across the U.S. consumer fraud ecosystem* (this includes all banks, payment systems, etc.). This project demonstrates an end-to-end AI system for fraud detection using SQL, Python, ML models, Explainable AI, and dashboard integration.

2. Business Problem

- Fraud causes billions in annual losses.
- Manual rules ('block if > $10,000') are too rigid.
- Needs:
  • Real-time fraud detection.
  • Minimized false positives (customer trust).
  • Transparent model explanations for compliance.

3. Data Overview

- Dataset Used: Kaggle Credit Card Fraud Dataset (284,807 transactions, anonymized).
- Fraud Distribution: ~0.17% of transactions are fraud (high imbalance).
- Key Columns: Transaction ID, Customer ID, Time, Amount, Merchant, Location/IP, Device, Label (Fraud/Not Fraud).

4. Data Preparation & Feature Engineering

- Loaded into PostgreSQL & cleaned.
- Engineered features:
  • Time-based: time since last transaction, frequency per hour.
  • Customer profile: ratio of transaction vs average spend.
  • Device/Location mismatch: fraud more likely if device or city changes.
- Handled imbalance with SMOTE oversampling.

5. Modeling & Evaluation

- Baseline: Logistic Regression (ROC-AUC ~0.83).
- Tree Models:
  • Random Forest → ROC-AUC ~0.97, strong recall.
  • XGBoost → ROC-AUC ~0.98, best balance.
- Anomaly Detection: Isolation Forest flagged outliers but with more false positives.

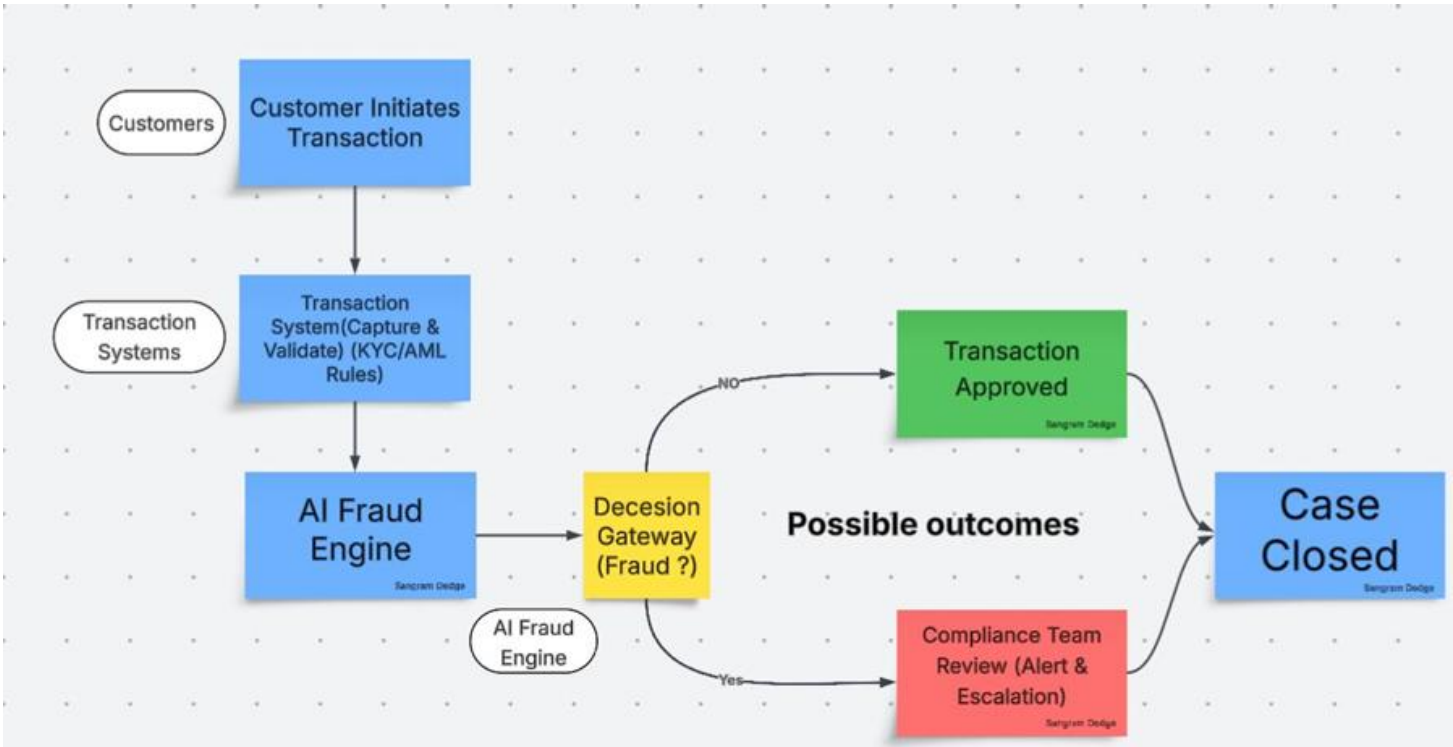
Best Model: XGBoost → chosen for deployment.

Metrics (XGBoost):
- Precision: ~0.92

- Recall: ~0.95 (critical for catching fraud)
- F1-score: ~0.93

6. Explainable AI (XAI)

- Used SHAP values for feature importance.
- Example flagged transaction explanation:
  • High amount (+$5000 vs avg $200).
  • Location mismatch (Boston vs last 10 in NYC).
  • Odd transaction time (3 AM).
- Helps compliance officers understand why a transaction was blocked.

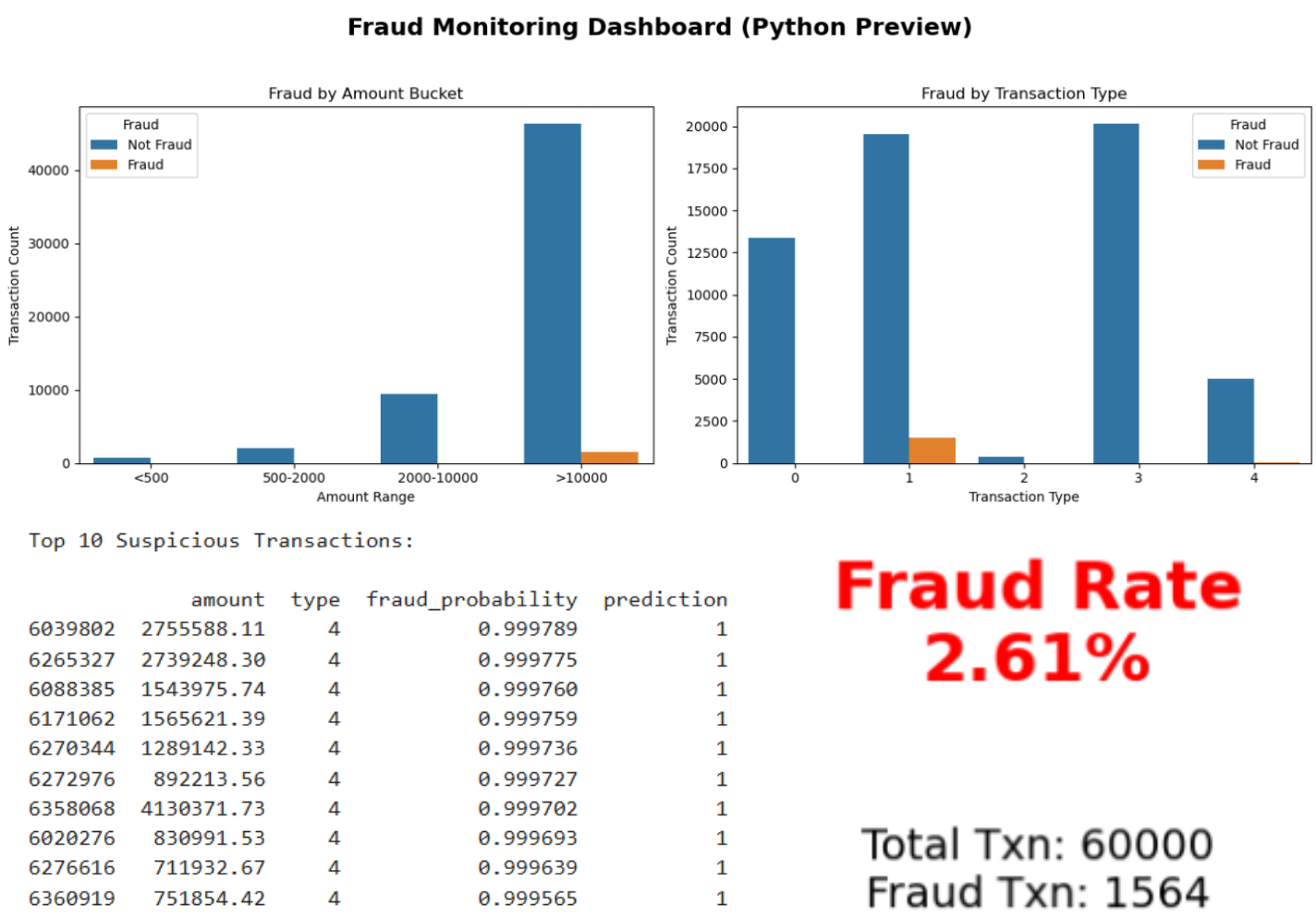


7. Real-Time API Prototype

- Built a FastAPI endpoint /predict:
  • Input: JSON with transaction details.
  • Output: Fraud probability + explanation.
- Integration with Power BI dashboards planned.

8. Dashboard Design (Planned)

- KPIs: Fraud % by merchant, region, transaction type.
- Real-time suspicious transaction alerts.
- AI-generated narrative summary (LLM integration).



9. Findings

- Fraud is rare but high-value.
- Device & location mismatch are strong fraud predictors.
- XGBoost outperformed other models, achieving high recall with manageable false positives.
- SHAP explanations improved transparency.

10. Future Scope

- Deploy API on AWS Lambda/EC2 for real-time scoring.
- Model drift monitoring → retrain when fraud patterns shift.
- Power BI/Tableau integration for fraud monitoring dashboards.
- Graph Neural Networks for fraud ring detection.
- LLM integration for auto-generated weekly fraud reports.