Fraud Detection System

Faris AlFaouri

April 29, 2025

Contents

List of Figures

Part I

Chapter 1

Chapter 1

Introduction

This project delivers a comprehensive **Fraud Detection System** for financial transaction security. It starts from raw data preprocessing, through feature engineering, model training, model evaluation, to full API deployment and continuous learning integration. This system identifies fraud in real time and is designed to continuously improve itself by retraining models based on newly inserted transaction data, representing a scalable machine learning MLOps approach combined with robust backend database connectivity.

1.1 Project Structure

- `Fraud.ipynb`: Full data cleaning, preprocessing, feature engineering, model training, and evaluation.
- `app.py`: FastAPI backend server for real-time fraud prediction and retraining.
- `Conn.py`: PostgreSQL database initialization and original data ingestion.
- `fraud_model.pkl`: Trained Random Forest model.
- `model_features.pkl`: Input features used by the model.
- `scaler.pkl`: StandardScaler model used for amount feature scaling.
- `templates/`: HTML frontend templates (form.html, result.html).
- `final_data.csv`: Final preprocessed dataset used for model training.

Chapter 2

Data Preprocessing

2.1 Data Loading

- Loaded the original dataset `fraud.csv` using pandas.

2.2 Data Cleaning

- Dropped rows with any null values.
- Removed extraneous quotation marks from all string fields.
- Dropped irrelevant columns: `customer`, `zipcodeOri`, `zipMerchant`.

2.3 Data Transformation

- Gender Encoding:
  - `M` → `Male`
  - `F` → `Female`
  - `U` → `Unknown`
  - `E` → `Enterprise`
- Age Encoding:
  - `0` → `<= 18`
  - `1` → `19-25`
  - `2` → `26-35`
  - `3` → `36-45`
  - `4` → `46-55`
  - `5` → `56-65`
  - `6` → `> 65`
  - `U` → `Unknown`

2.4 Feature Engineering

- Applied **One-Hot Encoding** to categorical variables: gender, age, category.
- Normalized the `amount` feature using **StandardScaler**.

Chapter 3

Model Evaluation and Results

3.1 Model Training

- Data was split into 80% training and 20% testing sets.
- Trained three models:
  1. Random Forest Classifier (best model selected)
  2. Logistic Regression
  3. Gradient Boosting Classifier

3.2 Model evaluation based on ROC AUC Score:

- Random Forest: **0.9999**
- Logistic Regression: **0.9976**
- Gradient Boosting: **0.9962**

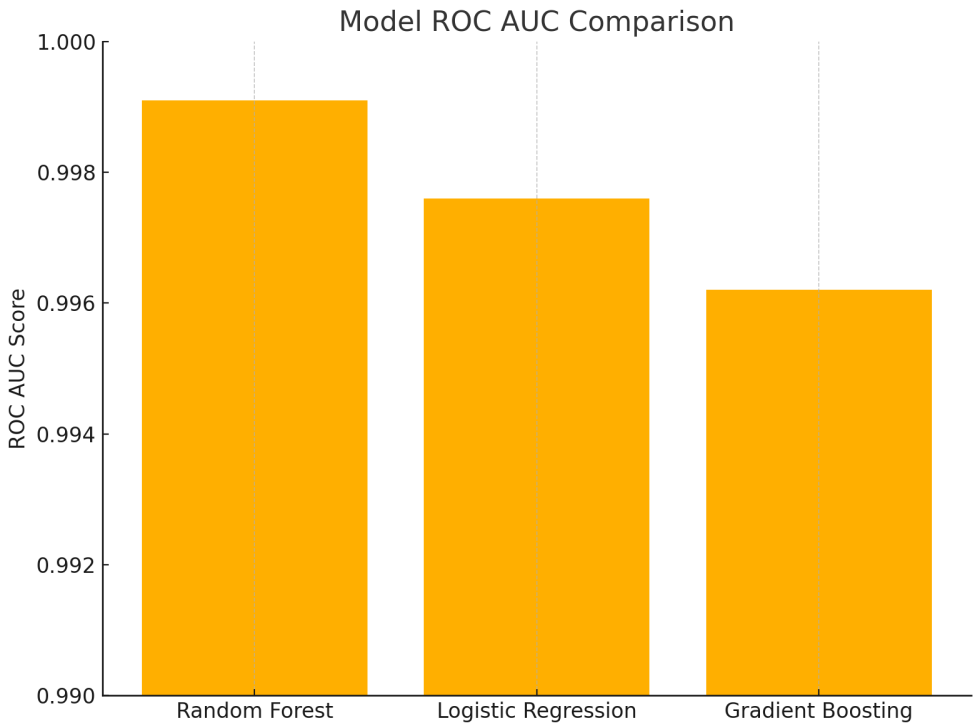

Figure 3.1: ROC AUC Compare between the models

3.3 Feature Importance

Top features based on Random Forest importance:

- amount
- category_es_transportation
- category_es_sportsandtoys
- category_es_health
- step
- category_es_food
- category_es_hotelservices
- category_es_wellnessandbeauty
- category_es_travel
- category_es_leisure

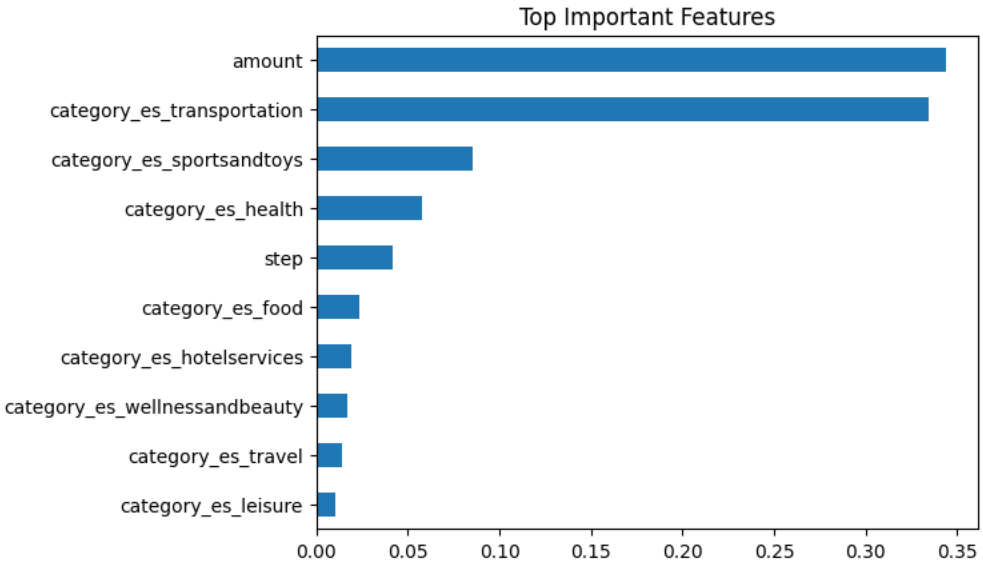

Figure 3.2: Top Important Features in Fraud Detection

Chapter 4

System Architecture

4.1 API Integration

- Real-time FastAPI server accepts transaction data.
- Prediction generated using deployed Random Forest model.
- PostgreSQL database stores transaction + prediction.
- Background task monitors input volume and retrains the model when new batches are accumulated.
- Model retraining includes continuous performance evaluation.

4.2 Database Connection

- PostgreSQL manages persistent storage of transactions.
- Transactions inserted into `fraud_table`.

4.3 MLOps Integration

- Model version control: Saving only improved models.
- Continuous Learning: Automatic retraining based on database triggers.
- After every 15 new entries, the system retrains the model and replaces it only if the new model improves the ROC AUC score.
- Monitoring model quality using ROC AUC during updates.

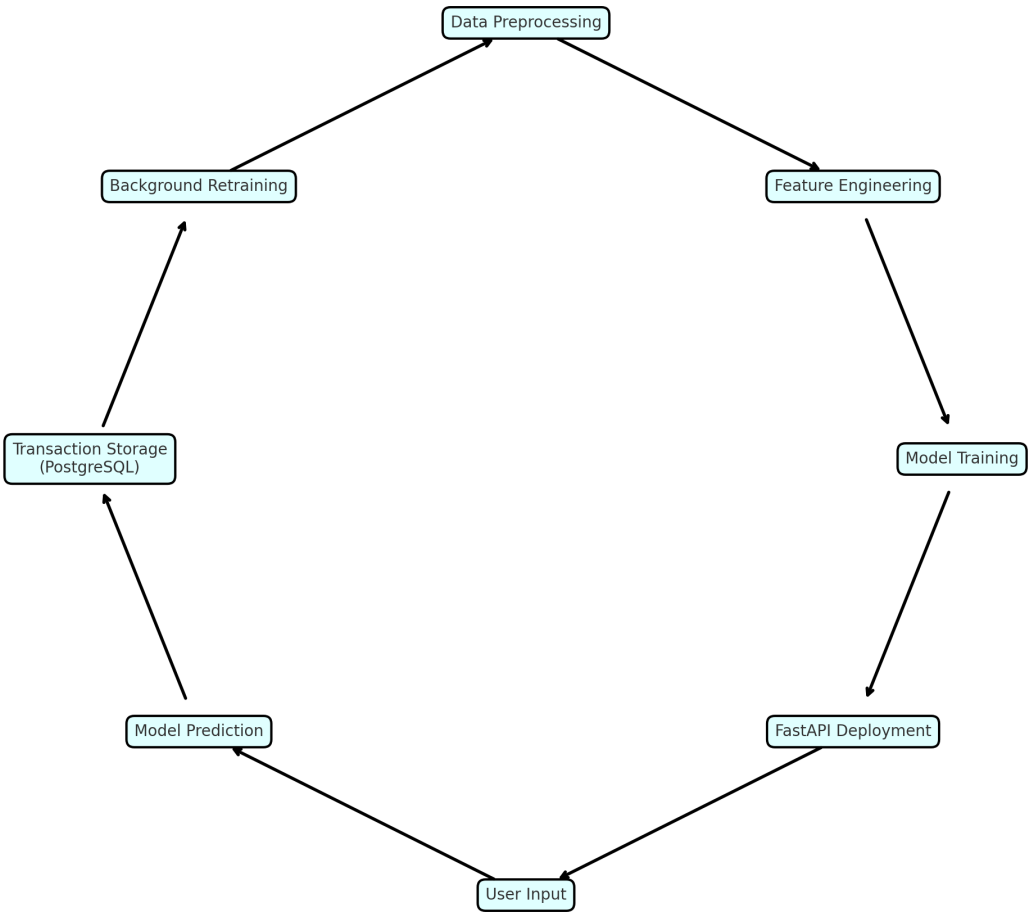

Figure 4.1: Continuous Learning Pipeline of Fraud Detection System

Chapter 5

Conclusion

This project showcases a full-stack deployment of an AI-driven fraud detection system. It includes comprehensive data preprocessing, multi-model training, real-time prediction via FastAPI, continuous monitoring with PostgreSQL integration, and automated retraining based on performance evaluation.

This system embodies the principles of machine learning operations (MLOps), ensuring scalability, resilience, and self-improvement over time, making it highly applicable to real-world financial security operations.