Machine Learning Application to

Fraud Detection Dataset

Using Python, MLFlow, and Tableau

Dhinaz Rangasamy

1. Project Overview

Objective: The goal of this project is to build a machine learning model that accurately detects fraudulent financial transactions in real time, improving upon existing fraud flagging mechanisms.

Dataset:

- Source: Synthetic dataset simulating mobile money transactions.
- Size: 6.3 million records.
- Key columns: type, amount, nameOrig, nameDest, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud

Goal: Build a fraud detection model with high recall and interpretability, capable of being deployed for real-time prediction.

---

2. Data Exploration Summary

- Fraud occurs in less than 1% of the transactions, indicating a significant class imbalance.
- Fraudulent transactions are heavily concentrated in TRANSFER and CASH_OUT types.
- In fraudulent transactions, the destination account often has a zero balance before and after the transaction.

---

3. Feature Engineering

Transaction Behavior Features:

- balanceChangeOrig: Difference in original account before and after transaction.
- balanceChangeDest: Change in destination balance.
- errorBalanceOrig: Balance discrepancy based on expected vs. actual balance.
- errorBalanceDest: Similar to above, for destination.

Interaction Features:

- isSameUser: Indicates if sender and receiver are the same.

Frequency & Pattern Features:

- liveTransactionsPerUser: Number of previous transactions for a user.
- liveFraudRatioPerUser: Ratio of fraudulent transactions previously associated with a user.

Categorical Encoding:

- type_encoded: Encoded form of transaction type using Label Encoding.

---

4. Model Selection & Training

Algorithm:

- XGBoost Classifier
- Chosen for its speed, accuracy, and ability to handle imbalanced datasets

Training Strategy:

- Train-test split: 70-30
- scale_pos_weight used to address class imbalance

Hyperparameters:

- n_estimators: 100
- learning_rate: 0.1
- max_depth: 6
- subsample: 0.8
- colsample_bytree: 0.8

Experiment Tracking:

- MLflow was used to log parameters, metrics, and artifacts across model training runs, enabling reproducibility and version control.

---

5. Model Evaluation

- **Confusion Matrix** and **Classification Report** used.
- **ROC AUC Score**: Indicates strong discriminative ability.

Focus:

- High recall prioritized to ensure fraudulent transactions are not missed.

---

6. Deployment Preparation

Artifacts Saved:

- Model: fraud_detection_xgb_model.json
- Label Encoder: `label_encoder.pkl`

Prediction Function:

- generate_features() replicates real-time feature logic for any new transaction.

---

7. Real-Time Prediction Pipeline

- Sorts transactions by time (`step`)
- Updates user history (transaction count, fraud history)
- Features derived in real-time and passed to model

---

8. Tools Used

- Python
- pandas, numpy
- xgboost, scikit-learn
- pickle
- MLflow (for experiment tracking)

---

9. Key Insights

- Fraud mainly happens in TRANSFER and CASH_OUT.
- Many fraud cases involve self-to-self transactions.
- Pre- and post-transaction balances provide strong signals for fraud.

---

10. Dashboard Interpretation

A Tableau dashboard was developed to visualize key trends and support model findings. The dashboard includes high-level KPIs and categorical breakdowns of transaction behavior, offering a clear window into the nature of fraud within the dataset.

Key Observations:

- **Fraud is rare but impactful.**
  With over 6 million transactions, the fraud rate is very low (well below 1%). This reinforces the need for precision-focused modeling, where the cost of missing a fraud far outweighs flagging a false positive.
- **Fraud is concentrated in specific transaction types.**
  The overwhelming majority of fraud cases occur in `TRANSFER` and `CASH_OUT` transactions. These types inherently involve money moving out of an account, making them natural targets for fraudulent activity.
- **Transaction distribution reflects platform use.**
  The general volume of transactions by type indicates that the platform is primarily used for CASH_OUT and PAYMENT transactions. However, despite TRANSFERs being less frequent overall, they contribute disproportionately to fraud cases.
- **Visual patterns support model feature engineering.**
  The categorical insights align with the model's reliance on features like `type_encoded`, `isSameUser`, and `errorBalanceOrig`, validating the use of these signals in identifying suspicious behavior.

These insights reinforce the logic behind the chosen model architecture and justify the emphasis placed on transaction type and balance features. The dashboard effectively bridges exploratory data analysis with model validation, helping stakeholders visualize the fraud landscape and understand where algorithmic intervention is most needed.