Fraudulent Transaction Detection for a Bank

August 15, 2025

Project Report

Contents

1 Executive Summary

This project develops a machine learning model to detect fraudulent transactions in a banking dataset containing over 1 million records. The dataset includes transaction details such as type, amount, and account balances, with a binary label indicating fraud (isFraud). The primary challenge is the severe class imbalance, with only 0.11% of transactions being fraudulent. We implemented a logistic regression model within a scikit-learn pipeline, incorporating preprocessing steps like scaling numerical features and encoding categorical variables, and addressed imbalance using class weighting. The model achieved a test accuracy of 93.48%, with a recall of 0.97 for fraud cases, though precision for fraud was low (0.02) due to many false positives. This report details the data exploration, methodology (emphasizing machine learning components), results, and recommendations for improving fraud detection.

2 Introduction

Fraudulent transactions pose significant risks to financial institutions, leading to monetary losses and erosion of customer trust. This project analyzes a synthetic banking dataset (AIML Dataset.csv) simulating real-world payment transactions to build a predictive model for identifying fraud in real-time. The dataset contains 1,048,575 entries with 11 columns:

- step: A unit of time (e.g., hour).
- type: Transaction type (e.g., PAYMENT, TRANSFER, CASH_OUT).
- amount: Transaction amount.
- nameOrig: Originator's account identifier.
- oldbalanceOrg/newbalanceOrig: Originator's balance before/after transaction.
- nameDest: Recipient's account identifier.
- oldbalanceDest/newbalanceDest: Recipient's balance before/after transaction.
- isFraud: Binary label (1 for fraud, 0 otherwise).
- isFlaggedFraud: Flag for potentially fraudulent large transfers (all 0 in this dataset).

The dataset exhibits severe class imbalance, with 1,047,433 non-fraudulent and 1,142 fraudulent transactions, necessitating specialized machine learning techniques to ensure effective fraud detection.

3 Data Exploration

The dataset was loaded using pandas, revealing no missing values across its 1,048,575 records. The isFraud column showed a significant imbalance: 99.89% non-fraudulent (1,047,433)

versus 0.11% fraudulent (1,142) transactions. The isFlaggedFraud column was uniformly zero, indicating it was not useful for modeling and was excluded from further analysis.

Numerical features (amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest) were examined for distributions and correlations. The type column, categorical with values like PAYMENT and TRANSFER, was analyzed for its relationship with fraud. Fraudulent transactions were predominantly associated with TRANSFER and CASH_OUT types, suggesting these transaction types are critical for modeling.

4 Methodology

4.1 Data Preprocessing

The machine learning pipeline was designed to handle both numerical and categorical features effectively:

- Numerical Features: The features amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, and newbalanceDest were standardized using StandardScaler to ensure zero mean and unit variance, improving model convergence.
- Categorical Features: The type column was encoded using OneHotEncoder with the drop='first' parameter to avoid multicollinearity, creating binary columns for transaction types (e.g., CASH_OUT, TRANSFER).
- Feature Selection: Columns nameOrig, nameDest, and isFlaggedFraud were excluded, as they provided no predictive value (identifiers or constant values).

The preprocessing steps were integrated into a scikit-learn ColumnTransformer, ensuring seamless application to both training and test sets.

4.2 Model Selection and Training

Given the class imbalance, a logistic regression model was chosen for its interpretability and effectiveness with imbalanced datasets when paired with class weighting. The model was configured with class_weight='balanced' to assign higher weights to the minority (fraud) class, mitigating bias toward the majority class. The maximum iterations were set to 1000 to ensure convergence. The pipeline combined preprocessing and modeling:

- ColumnTransformer: Applied scaling and encoding.
- LogisticRegression: Trained with balanced class weights.

The dataset was split into 70% training and 30% test sets (approximately 314,573 test samples), maintaining the class distribution using stratified sampling.

4.3 Model Evaluation

The model was evaluated using multiple metrics to account for the imbalance:

- Accuracy: Overall correctness of predictions.
- Precision, Recall, F1-Score: Focused on the fraud class to assess true positive rate and false positive trade-offs.
- Confusion Matrix: Detailed breakdown of true positives, false positives, true negatives, and false negatives.

The classification_report and confusion_matrix functions from scikit-learn were used to generate these metrics.

5 Results

The logistic regression model achieved the following performance on the test set (314,573 samples):

- Accuracy: 93.48%, indicating strong overall performance.
- Classification Report:
  - Non-fraud (0): Precision = 1.00, Recall = 0.93, F1-Score = 0.97.
  - Fraud (1): Precision = 0.02, Recall = 0.97, F1-Score = 0.03.
- Confusion Matrix:

	Predicted Non-Fraud	Predicted Fraud
Actual Non-Fraud	293,734	20,496
Actual Fraud	11	332

Table 1: Confusion Matrix

The high recall (0.97) for the fraud class indicates the model successfully identified most fraudulent transactions (332 out of 343). However, the low precision (0.02) reflects a high number of false positives (20,496), where non-fraudulent transactions were incorrectly flagged as fraud. This trade-off is common in imbalanced datasets, where prioritizing recall for the minority class increases false positives.

6 Recommendations

To improve the model's performance, particularly its precision for fraud detection, we propose:

- Advanced Algorithms: Explore ensemble methods like Random Forest or XGBoost, which may better handle class imbalance and capture complex patterns.
- SMOTE: Apply Synthetic Minority Oversampling Technique to generate synthetic fraud samples, potentially improving model training.
- Threshold Tuning: Adjust the decision threshold to balance precision and recall, reducing false positives while maintaining high fraud detection.
- Feature Engineering: Derive new features, such as transaction frequency or balance discrepancies, to enhance model discriminatory power.
- Real-Time Integration: Deploy the model in a real-time system with continuous monitoring to adapt to evolving fraud patterns.

These enhancements could reduce false positives and improve the model's practical utility for banking applications.

7 Conclusion

This project successfully developed a logistic regression-based pipeline for detecting fraudulent transactions, achieving 93.48% accuracy and 0.97 recall for fraud cases. Despite the low precision for fraud, the model demonstrates strong potential for identifying suspicious transactions. Future work should focus on improving precision through advanced algorithms and feature engineering to create a more robust fraud detection system.