Internship Task : Accredian

Data Science & Machine Learning

Report

By

Aditya Kumar Pandey

IIIT Bhagalpur, MTech (CSE)

AI & Data Science

Fraud Detection Project Report

1. Project Overview

This project focused on proactively detecting fraudulent transactions for a financial company using a large, real-world dataset. The goal was to develop a model that can flag suspicious transactions, provide actionable insights, and recommend infrastructure updates to reduce fraud losses.

2. Data Preparation

- Data Loading & Initial Inspection: Imported and examined data (over 6 million transactions, 10–11 features).
- Missing Values: No missing values were found; all data was complete.
- Outliers: EDA revealed no extreme outliers requiring removal. Transaction amounts were log-transformed to reduce skew.
- Multicollinearity: After feature engineering, several features were collinear. Used VIF to identify and remove collinear/engineered features (e.g., balance deltas, multiple time features). Post-cleanup, all remaining features had acceptable VIF values.

3. Exploratory Data Analysis (EDA)

- Transaction Type Analysis: Identified that fraud occurs only in “TRANSFER” and “CASH_OUT” types; filtered out other types.
- Class Imbalance: Fraud cases are extremely rare (0.13% of all transactions), necessitating class imbalance handling.
- Fraud Patterns: Visualized fraud rates by transaction type and amount; found that fraud is associated with emptying originator accounts and moderate-sized transfers.

4. Feature Engineering

- Domain-Driven Features: Created features like “is_account_emptied”, “type_TRANSFER”, log-transformed amount, “amount”, and origin/destination balances.

- Filtered Features: Removed redundant, collinear, and low-importance features.
- One-Hot Encoding: Encoded categorical variables like transaction type.
- Final Feature Set: Included transaction amount, old/new balances, log amount, transaction type, and indicators for emptied accounts and merchant destinations.

5. Model Development

- Models: Trained and evaluated Logistic Regression, Random Forest, and XGBoost.
- Handling Imbalance: Used class weights and scale_pos_weight to address class imbalance.
- Train-Test Split: Split data into 70% training and 30% test, stratified by fraud label.

6. Model Comparison

- Logistic Regression: High recall (91% of frauds caught), but low precision (6% of flags are real frauds), resulting in many false alarms. ROC-AUC: 0.987.
- Random Forest: Very high precision (97% of flags are real frauds), but lower recall (79%). ROC-AUC: 0.994.
- XGBoost: Highest recall (99% of frauds caught) and ROC-AUC (0.998), with moderate precision (55%). Best overall balance of catching frauds and limiting false alarms.

7. Feature Importance

- Top Features: “is_account_emptied”, “type_TRANSFER”, “oldbalanceOrg”, “newbalanceDest”, “log_amount”.
- Interpretation: Emptying accounts via transfers is a strong fraud indicator. Large transactions were less likely to be fraud in this dataset, possibly due to legitimate business activity.

8. Business Recommendations

- Real-Time Monitoring: Flag transactions that empty accounts, especially via transfers.
- Rule-Based Alerts: Combine ML predictions with simple business rules for rapid response.

- Human Review: Use staff to review ML-generated alerts to reduce false positives.
- Behavioral Analytics: Monitor user patterns over time for emerging fraud tactics.
- Continuous Improvement: Regularly retrain models and update rules based on new data.

9. Outcome Validation

- Track KPIs: Measure fraud detection rate, false positive rate, and time to detection.
- A/B Testing: Compare new system performance against the old or control groups.
- Feedback Loop: Collect input from analysts and customers to refine the system.
- Regular Audits: Periodically review and update the fraud detection pipeline.

10. Summary and Conclusion

By following a rigorous data science process—data cleaning, feature engineering, multicollinearity checks, and model evaluation—we built a robust fraud detection system. XGBoost is recommended for deployment due to its superior balance of recall and precision, though Random Forest is also a strong choice if minimizing false alarms is the priority. The top predictors of fraud were emptying the originator's account, transfer type, transaction amount, and balance changes—all consistent with real-world fraud patterns. Continuous monitoring and stakeholder engagement will ensure the system stays effective as fraud tactics evolve.