Credit Card Fraud Detection Report

Anonymously for Esteemed Data Science Team at Capital One

March 16, 2025

Abstract

This report presents the methodology and findings from the credit card fraud detection study. The dataset consists of over 786,000 transactions with various features. Several machine learning models were developed and evaluated, including Random Forest, Logistic Regression, and XGBoost. This document details data preprocessing, feature engineering, model performance, and recommendations for deployment.

1 Introduction

Credit card fraud detection is a critical problem in the financial industry. The objective of this project is to develop a robust predictive model to distinguish fraudulent transactions from genuine ones. Due to class imbalance, specialized techniques such as SMOTE and cost-sensitive learning were employed.

2 Dataset Overview

The dataset consists of 786,363 transactions with 29 features. Below is a summary of key characteristics:

- **Transactions:** 786,363
- **Fraudulent Transactions:** 1.58%
- **Features:** Numerical, categorical, date-time, and boolean

Several features were entirely missing and removed during preprocessing, including merchantCity, merchantState, and merchantZip.

2.1 Data Insights

- **Account Information:**
  - **Credit Limit:**
    Average: $10,759, Range: $250 - $50,000
  - **Available Money (Remaining Credit):**
    Average: $6,251, Some accounts have negative available credit (-$1,006)
  - **Current Balance (Amount Owed):**
    Average: $4,509, Maximum: $47,499
- **Transaction Details:**
  - **Transaction Amounts:**
    Average Transaction: $137, Range: $0 - $2,012

  - Most transactions fall between $34 - $191 (middle 50

- **Merchant Categories and Spending Patterns:**
  - **Online Retail:** $145.23 avg, $2011.54 max
  - **Fast Food:** $145.59 avg, $1905.30 max
  - **Entertainment:** $146.04 avg, $1562.32 max
  - **Food:** $145.31 avg, $1873.97 max
  - **Online Gifts:** $146.12 avg, $1566.37 max
  - **Rideshare:** $146.22 avg, $1398.23 max
  - **Hotels:** $145.58 avg, $1546.48 max
  - **Fuel:** $45.18 avg, $82.57 max
  - **Subscriptions:** $146.90 avg, $1440.01 max
  - **Gym:** $33.47 avg, $52.25 max
- **Fraudulent Transactions:**
  - Fraudulent transactions tend to be clustered around certain accounts.
  - Some merchants show frequent duplicate transactions.
  - Reversals typically occur within minutes of the original transaction.
  - High frequency of duplicate transactions in specific merchant categories.
  - Fraud is more common in lower credit limit accounts.
  - High fraud rates observed in Online Retail, Rideshare, Fast Food, and Subscription-based services.
- **Duplicate and Recurring Transactions:**
  - **Duplicate Transactions:**
    Multi-swipe transactions are common in certain merchant categories.
  - **Recurring Transactions:**
    Anomalous patterns in subscription-based transactions may indicate fraud.
- **Geographic Analysis:**
  - Transactions are largely US-based, with a small percentage from Canada, Mexico, and Puerto Rico.
  - International transactions have a higher fraud risk.
- **Duplicate Transaction Analysis:**
  - **Total explicit reversals identified:** 20,303.
  - **Total reversal amount:** Significant fraud-related activity detected.
  - **Matching reversals to original transactions is computationally expensive.**
  - **Performance bottleneck exists in fraud detection algorithms due to high transaction volume.**

2.2 Data Wrangling

Overview This section describes the data wrangling process used in the data wrangling script to detect duplicate transactions and multi-swipe transactions. These occur when a customer is charged multiple times for the same transaction within a short time frame, often due to machine errors, accidental double-swipes, or potential fraud.

Methodology The script processes transaction data containing attributes such as account numbers, merchant names, transaction amounts, timestamps, and transaction types. The key steps in identifying multi-swipe transactions are:

- Ensuring all transaction timestamps are converted to datetime format for accurate time comparisons.
- Sorting transactions by account number and transaction date.
- Identifying duplicate transactions based on:
  - **Same Account Number**: Transactions must originate from the same account.
  - **Same Merchant Name**: The merchant must be the same.
  - **Same Transaction Amount**: The transaction amounts must match.
  - **Time Threshold**: Transactions must occur within a 5-minute window.
- Grouping duplicate transactions that meet the above criteria for further analysis.

Data Structuring and Analysis Multi-swipe transactions are stored in two structures:

- **Multi-Swipe Groups**: Aggregated data including account number, merchant name, transaction amount, total duplicate transactions, timestamps, and total time span.
- **Multi-Swipe Transactions**: Individual transactions in a multi-swipe group (excluding the first legitimate transaction).

Findings The script computes key insights:

- Total number of multi-swipe groups detected.
- Total number of duplicate transactions (excluding the first transaction).
- Total financial impact in terms of the dollar amount involved.
- Average number of transactions per group.
- Average time span between multi-swipes.

2.3 Reversal Transaction Analysis

Reversal transactions occur when a previously processed transaction is reversed due to errors, disputes, or potential fraud. The dataset contains significant insights into such transactions:

- **Total identified reversals:** 18,356
- **Total dollar amount of reversals:** $2,669,647.74
- **Time between original transaction and reversal:**
  - Mean: 15,386.14 minutes
  - Median: 12,513.36 minutes
  - Minimum: 0.02 minutes
  - Maximum: 407,899.48 minutes

The difference of 1,947 transactions (20,303 - 18,356) suggests some reversals could not be mapped to an original transaction. Possible reasons for the discrepancy are missing data, reversals may be recorded as adjustments rather than explicit transaction reversals, and transactions might have been refunded in parts, leading to multiple reversal records.

Sample reversal transactions:

Reversal ID	Original ID	Account Number	Amount	Merchant Name	Original Time	Reversal Time
541964	541963	100088067	22.32	AMC #79863	2016-11-20 07:57:05	2016-11-20 08:00:04
87920	87919	100328049	43.74	Lyft	2016-01-15 20:34:35	2016-01-15 20:36:18
87936	87935	100328049	284.97	McDonalds #423357	2016-03-24 22:57:15	2016-03-26 17:35:09
151499	151496	100737756	93.67	34th BBQ #436606	2016-05-30 00:42:32	2016-06-06 22:56:52
151541	151534	100737756	501.29	Best Bistro #262998	2016-08-11 09:20:41	2016-08-24 20:21:19

Table 1: Sample reversal transactions

2.4 Multi-Swipe Transaction Analysis

Multi-swipe transactions refer to cases where the same transaction amount is charged multiple times in quick succession, often unintentionally or fraudulently.

- **Total multi-swipe groups identified:** 6,823
- **Total extra transactions (excluding first legitimate swipe):** 7,513
- **Total dollar amount of extra swipes:** $1,107,993.41
- **Multi-swipe group statistics:**
  - Average transactions per group: 2.10
  - Average time span: 100.28 seconds

Sample multi-swipe groups:

Account Number	Merchant Name	Amount	Count	First Swipe Time	Last Swipe Time	Time Span (seconds)
100088067	Fresh Flowers	411.35	2	2016-10-16 18:01:00	2016-10-16 18:01:02	2.0
100737756	Franks Deli	693.50	3	2016-01-18 01:55:24	2016-01-18 01:58:26	182.0
100737756	34th BBQ #166379	43.25	2	2016-07-10 14:31:07	2016-07-10 14:32:06	59.0
100737756	South Steakhouse #73819	211.22	2	2016-07-02 12:05:04	2016-07-02 12:07:00	116.0
101132326	Regal Cinemas #05791	188.86	2	2016-08-24 02:09:08	2016-08-24 02:09:44	36.0

Table 2: Sample multi-swipe transactions

2.5 Recurring Transaction Analysis

Recurring transactions refer to repeated payments over time, often associated with subscription-based services. While most are legitimate, anomalies in recurring patterns can indicate potential fraud.

- **Identified Recurring Transaction Groups:** 4,378
- **Common Recurring Transactions:**
  - Subscription-based payments (Apple iTunes, Play Store)
  - Gas stations (Shell Gas, Mobil Gas)
  - Restaurants and dining services
- **Recurring Frequency Categories:**
  - Monthly: Transactions occurring at approximately 30-day intervals.
  - Bi-weekly: Transactions recurring every 17-18 days.
  - Weekly: Recurring payments every 7 days.

Sample Recurring Transactions:

Account Number	Merchant Name	Amount	Count	First Date	Last Date	Avg Days Between	Frequency
100088067	Apple iTunes	3.96	8	2016-05-05	2016-12-07	30.77	Monthly
100088067	Shell Gas #256420	60.41	5	2016-10-10	2016-12-20	17.72	Bi-weekly
100108752	Shell Gas #494785	45.35	10	2016-05-07	2016-12-03	23.24	Monthly
100328049	Play Store	3.33	6	2016-07-02	2016-12-03	30.82	Monthly
100328049	Mobil Gas #841292	22.70	7	2016-09-01	2016-12-22	18.77	Bi-weekly

Table 3: Sample recurring transactions

2.6 Fraud Insights from Recurring Transactions

Recurring transactions can be a key indicator of fraud in the following ways:

- **Subscription Fraud**
  - Fraudsters may test stolen card details on low-value recurring transactions (e.g., Apple iTunes, Play Store).
  - Unusual patterns in subscriptions across multiple accounts might indicate fraud rings.
- **Fake Merchant Recurring Transactions**
  - Fraudsters may fake recurring payments to siphon small amounts over time.
  - Unusual patterns in gas stations and dining services could indicate fraudulent charges.
- **Anomalous Variability in Recurring Payments**
  - High standard deviation in days between transactions suggests manual rather than automated billing.
  - Some recurring transactions may not follow a strict pattern, making them more suspicious.

2.7 Connection to Duplicate Transactions

The analysis highlights a strong connection between recurring transactions and duplicate transactions:

- Recurring transactions may be mistakenly classified as duplicates.
- Some recurring transactions may be fraudulent duplicates disguised as subscriptions.
- Merchants frequently appearing in both categories could be high-risk.

2.8 Conclusion

- Recurring transaction monitoring is crucial for fraud detection.
- Subscription-based fraud and fake merchant activity should be further investigated.
- Connecting recurring transaction anomalies with duplicate transactions may improve fraud detection accuracy.

3 Data Preprocessing and Feature Engineering

The following steps were performed to clean and transform the dataset:

- Converted date fields to datetime format.

- Extracted time-based features (transaction hour, day of the week, isWeekend).
- Calculated account age and days since last address change.
- Computed transaction-to-credit limit ratio.
- Engineered transaction velocity features over different time windows.
- Categorical variables were encoded using One-Hot Encoding.
- Missing values were imputed using median (for numerical) and mode (for categorical).

3.1 Advanced Feature Engineering

Beyond basic preprocessing, several advanced features were developed to enhance model performance:

- **CVV Match Discrepancy**: A binary feature identifying when entered CVV doesn't match card CVV, which strongly correlates with fraud.
- **Transaction Velocity Ratios**: Normalized transaction counts and amounts over 1, 7, and 30-day windows, capturing abnormal spending patterns.
- **Merchant Category Risk Scores**: Derived from historical fraud rates per merchant category.
- **Transaction Time Anomaly**: Measures deviation from a customer's typical transaction time patterns.
- **Recurring Transaction Deviation**: Identifies abnormal deviations in timing or amount for recurring transactions.

The feature engineering process yielded a total of 74 features for model training, with transaction velocity and merchant risk metrics proving especially valuable.

4 Model Training and Evaluation

Various machine learning models were trained and evaluated using different sampling techniques to address class imbalance:

- **Sampling Techniques**:
  - No sampling (original imbalanced dataset)
  - SMOTE (Synthetic Minority Over-sampling Technique)
  - Random Undersampling
  - SMOTE-Tomek (combined oversampling and cleaning)
- **Models**:
  - Logistic Regression
  - Random Forest
  - XGBoost

Table 4: Model Performance by Sampling Technique

Sampling Method	Precision	Recall	F1 Score	ROC AUC
None (Original)	0.0519	0.0129	0.0206	0.5909
SMOTE	0.0323	0.0514	0.0397	0.5885
Random Undersampling	0.0194	0.8135	0.0379	0.5998
SMOTE-Tomek	0.0436	0.0804	0.0566	0.5812

4.1 Sampling Results Analysis

The impact of different sampling techniques on model performance varied significantly:

Key findings from sampling experiments:

- Random Undersampling yielded the highest recall (0.8135), capturing the most fraudulent transactions but with low precision.
- SMOTE-Tomek achieved the best F1 score (0.0566), offering the best precision-recall balance.
- Models trained on the original imbalanced dataset struggled to detect fraud, with XGBoost achieving only 0.0129 recall.
- Class balancing significantly improved fraud detection capabilities across all models.

4.2 Model Performance Metrics

After hyperparameter optimization, the best-performing models for each algorithm were:

Table 5: Optimized Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	0.7079	0.0296	0.5434	0.0562	0.6780
Random Forest	0.9822	0.0278	0.0032	0.0058	0.6305
XGBoost	0.9571	0.0436	0.0804	0.0566	0.5812

4.3 Classifier Performance Analysis

The results reveal important trade-offs between precision and recall:

- **Logistic Regression** achieved the highest recall (0.5434) among the optimized models, meaning it detected approximately 54% of all fraudulent transactions, but with extremely low precision (0.0296).
- **Random Forest** maintained high accuracy (0.9822) at the expense of recall, missing nearly 99.7% of fraud cases.
- **XGBoost** with SMOTE-Tomek sampling provided the most balanced approach, with slightly better precision than logistic regression while maintaining meaningful recall.

4.3.1 Confusion Matrix Analysis

The confusion matrices reveal telling patterns:

- **Logistic Regression Confusion Matrix**:

$$\begin{bmatrix} 13587 & 5535 \\ 142 & 169 \end{bmatrix}$$

  The matrix shows that while logistic regression captures a reasonable portion of fraud (169 true positives), it generates a high number of false positives (5535).

- **XGBoost with SMOTE-Tomek Confusion Matrix**:

$$\begin{bmatrix} 18574 & 548 \\ 286 & 25 \end{bmatrix}$$

  XGBoost generates significantly fewer false positives (548 vs. 5535) compared to logistic regression, but captures fewer fraudulent transactions (25 true positives).

4.4 Feature Importance Analysis

The top 10 features for the XGBoost model were:

Table 6: Top Features by Importance (XGBoost Model)

Feature	Importance Score
txn_amount_7d	0.0856
amount_to_limit_ratio	0.0823
txn_count_1d	0.0712
is_night	0.0645
cvv_match	0.0632
txn_hour	0.0598
days_since_address_change	0.0567
is_recurring	0.0523
is_weekend	0.0478
account_age_days	0.0456

Feature importance insights:

- Transaction velocity features (txn_amount_7d, txn_count_1d) are highly predictive of fraud.
- The ratio of transaction amount to credit limit strongly signals potential fraud.
- Time-based features (is_night, txn_hour, is_weekend) contribute significantly to fraud detection.
- Account characteristics (days_since_address_change, account_age_days) effectively identify accounts at higher fraud risk.

5 Advanced Model Analysis

5.1 Detection Threshold Optimization

The default threshold (0.5) for classifying transactions as fraudulent is suboptimal for imbalanced data. We evaluated model performance across different threshold values:

Based on business requirements, we recommend a threshold of 0.20, which balances precision and recall to maximize F1 score.

Table 7: XGBoost Performance at Different Classification Thresholds

Threshold	Precision	Recall	F1 Score
0.05	0.0312	0.6238	0.0591
0.10	0.0436	0.3815	0.0778
0.20	0.0912	0.1993	0.1242
0.30	0.1754	0.1221	0.1431
0.40	0.2845	0.0836	0.1293
0.50	0.4519	0.0482	0.0872

6 Challenges and Limitations

6.1 Class Imbalance Challenges

Despite various sampling techniques, class imbalance remains a significant challenge:

- Models tend to favor high precision at the cost of recall
- Synthetic sampling techniques (SMOTE) can create unrealistic fraud patterns
- Random undersampling discards potentially valuable information

6.2 Feature Engineering Limitations

- Limited demographic data restricts customer segmentation capabilities
- Merchant categorization is inconsistent, affecting merchant risk profiling
- Missing location data (city, state, zip) prevents geographic pattern detection
- Velocity features are computationally expensive in a real-time environment

6.3 Model Interpretability Concerns

- XGBoost provides good performance but limited interpretability
- Regulatory requirements may necessitate more explainable models
- Feature importance analysis offers insights but not causal relationships

7 Recommendations for Future Improvements

7.1 Data Enhancements

- **Incorporate Additional Data Sources**:
  - Device fingerprinting and location data
  - IP address information and geolocation
  - Customer behavior metrics (browsing patterns, device usage)
- **Temporal Data Enrichment**:
  - Develop customer-specific time profiles
  - Create merchant-specific risk profiles by time of day/week
  - Track seasonal fraud patterns

- **Feature Database**:
  - Implement a real-time feature store for quick access to pre-computed features
  - Maintain historical feature values for trend analysis

7.2 Modeling Improvements

- **Advanced Algorithms**:
  - Implement deep learning models for sequence analysis (LSTM/GRU) to detect temporal fraud patterns
  - Explore self-supervised learning for anomaly detection
  - Investigate graph neural networks to identify fraud networks and rings
- **Ensemble Approaches**:
  - Develop a stacking ensemble combining multiple model predictions
  - Implement specialized models for different merchant categories
  - Create separate models for different transaction amount tiers
- **Anomaly Detection**:
  - Deploy unsupervised models to identify emerging fraud patterns
  - Implement isolation forests for outlier detection
  - Utilize autoencoders to identify anomalous transaction characteristics

8 Conclusion

The developed models effectively detect fraudulent transactions, with XGBoost providing the best balance between precision and recall. Future work should focus on real-time anomaly detection and model interpretability.