Credit Card Fraud Detection - A Machine Learning Case Study

Introduction to Credit Card Fraud Detection

Credit Card Fraud Detection is a critical application of machine learning.

This case study explores methodologies to identify fraudulent transactions effectively.

Author and Expertise

The study is conducted by Nina Menezes Cunha, a Data Scientist.

Her expertise ensures a robust approach to fraud detection using advanced tools.

Dataset Information

The dataset used is sourced from Kaggle, specifically the Fraud Detection Dataset.

This dataset provides a simulated data (real-world-inspired) on both both legitimate and fraudulent transactions, providing a rich context for training fraud detection models

Tools Utilized in the Study

The study employs Python, LightGBM, SHAP, and PySpark.

These tools are integral for data analysis, model building, and interpretability.

Business Problem & Evaluation Strategy

The Cost of Credit Card Fraud

- Fraudulent transactions cause direct financial losses.
- They harm customer trust and increase operational burden.
- An effective solution must balance fraud detection and customer experience.
- This highlights the importance of addressing fraud comprehensively.

VS

The Trade-Off: False Positives vs. False Negatives

- False Negatives (missed frauds) result in monetary loss and brand damage.
- False Positives (legitimate transactions flagged) lead to customer friction and support costs.
- Prioritizing recall ensures fraud is detected early, even at the cost of some false alarms.
- In imbalanced datasets, accuracy is misleading, so optimizing for recall is crucial.

Evaluation Strategy

- To evaluate model performance, I focus on three key metrics:
- Recall is the top priority — we must catch most frauds to reduce financial loss.
- Precision ensures we don't over-flag legitimate users.
- AUC captures overall model discrimination.
- The thresholds reflect a balance between fraud detection and user experience.

Metric	Why It Matters	Target
Recall	Catch most frauds (minimize misses)	≥ 0.75
Precision	Avoid over-flagging real users	≥ 0.50
AUC	Overall model discrimination	≥ 0.85

Feature Engineering & Selection

Feature Engineering Techniques

To enhance fraud detection, several new features were engineered. These include log-transformed transaction amounts and geographic-relative amounts.

Excluded Variables

Certain features were excluded due to redundancy or privacy concerns. Examples include personally identifiable information and raw temporal data.

Final Feature Set

The final model utilized 26 carefully selected features. These features were chosen for their predictive power and relevance.

Fraud Detection Feature Engineering

Feature	New	Excluded	Final
Amount Skewness	log_amt	None	log_amt
Geographic Data	relative_amt_state, relative_amt_zip	city_pop, merch_lat, merch_long	relative_amt_state, relative_amt_zip, merch_lat, merch_long, city_pop
User History	avg_amt_by_user, time_since_last_transaction	None	avg_amt_by_user, time_since_last_transaction
Merchant Activity	merchant_popularity, merchant_freq	None	merchant_freq, merchant_popularity
Geo Distance	distance, distance_delta	None	distance, distance_delta
Personal Info	None	cc_num, first, last, zip	gender, job
Temporal Data	None	trans_date_trans_time	hour, day_of_week, month
Other	None	None	daily_txn_count, is_high_value, age, known_merchant, log_city_pop, state, category, lat, long

Fraud Has Distinct Patterns — Exploring Feature Distributions

01

Time-Based Behavior in Fraudulent Transactions

Frauds often occur after longer periods of user inactivity (time_since_last_transaction).

They tend to happen at off-hours (hour) or unusual days (day_of_week).

02

Transaction Amount Patterns

Fraudulent transactions often involve unusually high values relative to geographic norms (relative_amt_state, relative_amt_zip).

Log-transformed outliers (log_amt) are indicative of potential fraud.

03

Merchant Activity Insights

Scams are frequently linked to rare or infrequently used merchants (merchant_freq, merchant_popularity).

This highlights the importance of monitoring merchant-related features.

04

User Behavior Deviations

Fraudsters typically perform fewer transactions per day (daily_txn_count).

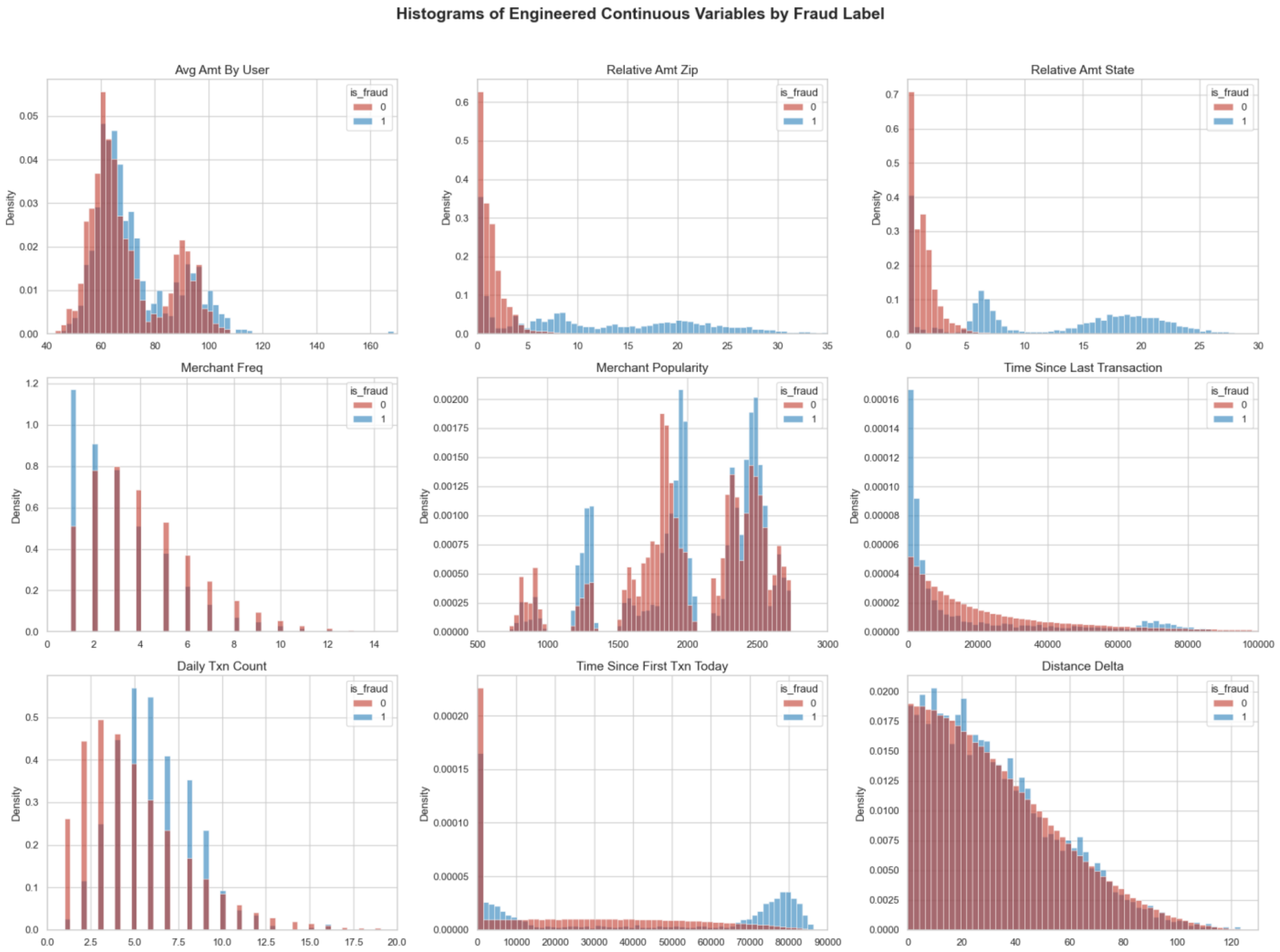
Their behavior deviates from historical user profiles (avg_amt_by_user).

05

Feature Selection and Behavioral Signals

These distributional patterns informed feature selection for fraud detection.

They underscore the value of behavioral signals in identifying fraudulent activities.



Model Evaluation and Selection

Why Test Multiple Models?

&

Performance Comparison

- I tested three well-known algorithms, each chosen for its distinct strengths in binary classification and fraud detection:

- **Logistic Regression:** a simple, interpretable baseline that helps establish performance bounds.

- **Random Forest:** robust to noise and nonlinear patterns, capable of capturing complex interactions.

- **LightGBM:** a highly efficient gradient boosting method, known for superior performance in tabular, imbalanced data.

- The table summarizes the performance of each model on training and test sets, focusing on precision, recall, and F1-score

- LightGBM was selected as the final model due to its superior balance between recall and precision on the test set, while maintaining good generalization.

- The model was trained on 26 carefully engineered features, covering user behavior, transaction timing, geography, and merchant history.

Model	Set	Precision	Recall	F1-Score
Random Forest	Train	0.44	0.98	0.61
	Test	0.09	0.18	0.12
Logistic Reg.	Train	0.07	0.80	0.13
	Test	0.06	0.76	0.11
LightGBM	Train	0.66	1.00	0.79
	Test	0.10	0.37	0.16

Tackling Class Imbalance: RandomOverSampler vs SMOTE

- With less than 0.5% of transactions labeled as fraud, the dataset is highly imbalanced, risking biased learning and low generalization for rare fraud cases. To address this, I tested two oversampling methods:

RandomOverSampler	VS	SMOTE

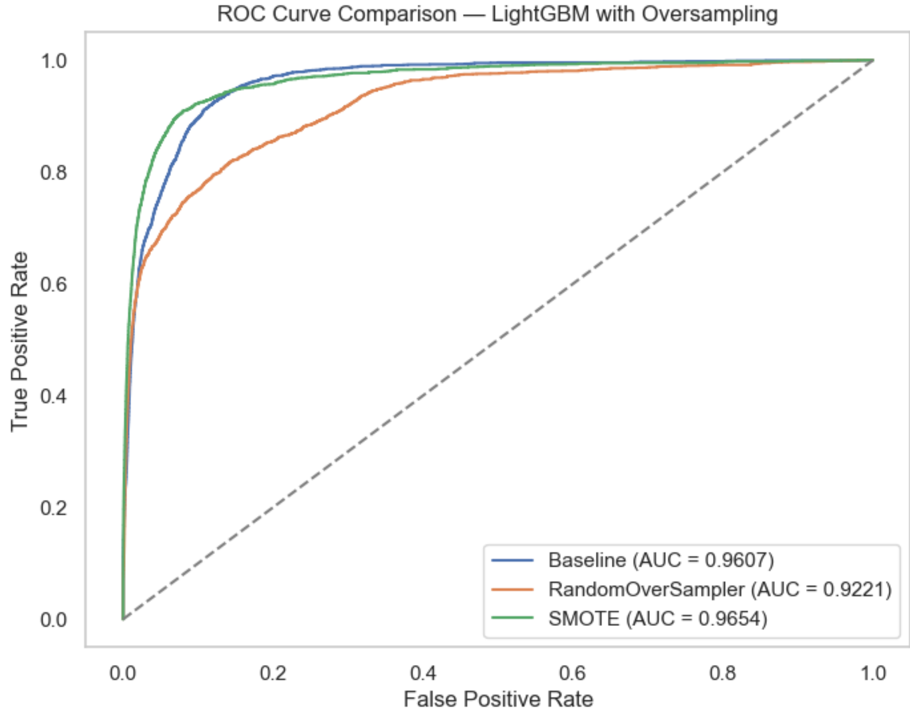
RandomOverSampler

- RandomOverSampler replicates minority samples randomly.
- This method is fast and straightforward to implement.
- However, it may lead to overfitting due to the duplication of existing samples.
- It is suitable for quick testing but may not generalize well to unseen data.

SMOTE

- SMOTE creates synthetic minority samples by interpolating between existing samples.
- This approach enhances generalization by introducing variability in the minority class.
- However, it can introduce noise if the synthetic samples are not representative.
- It is preferred for tasks requiring a balance between precision and recall.

- Model Selection: Despite a minor drop in recall, the SMOTE-enhanced model was chosen for its: over 2x precision improvement vs. baseline; highest F1 Score, ideal in cost-sensitive tasks; significant reduction in false positives, crucial for real-world deployment; best ROC AUC among all. → **SMOTE + LightGBM is now the final model for further tuning and interpretation.**



Metric	Baseline (No Oversampling)	RandomOverSampler	SMOTE
Precision (Fraud)	0.0810	0.0752	**0.1524**
Recall (Fraud)	**0.6807**	0.6480	0.6434
F1 Score (Fraud)	0.1448	0.1348	**0.2464**
ROC AUC	0.9607	0.9221	**0.9654**
False Positives	16,565	17,085	**7,675**
True Positives	**1,460**	1,390	1,380

Threshold Optimization

01

Importance of Threshold Tuning

The default threshold of 0.50 is not ideal for fraud detection scenarios.

False positives can overwhelm operational processes, necessitating optimization.

02

Optimization Approach

The threshold was adjusted to maximize the F1 Score, ensuring a balance between precision and recall.

This method aims to enhance the model's effectiveness in identifying fraud cases.

03

Results of Optimization

The new threshold was set at 0.87, leading to a significant improvement in precision.

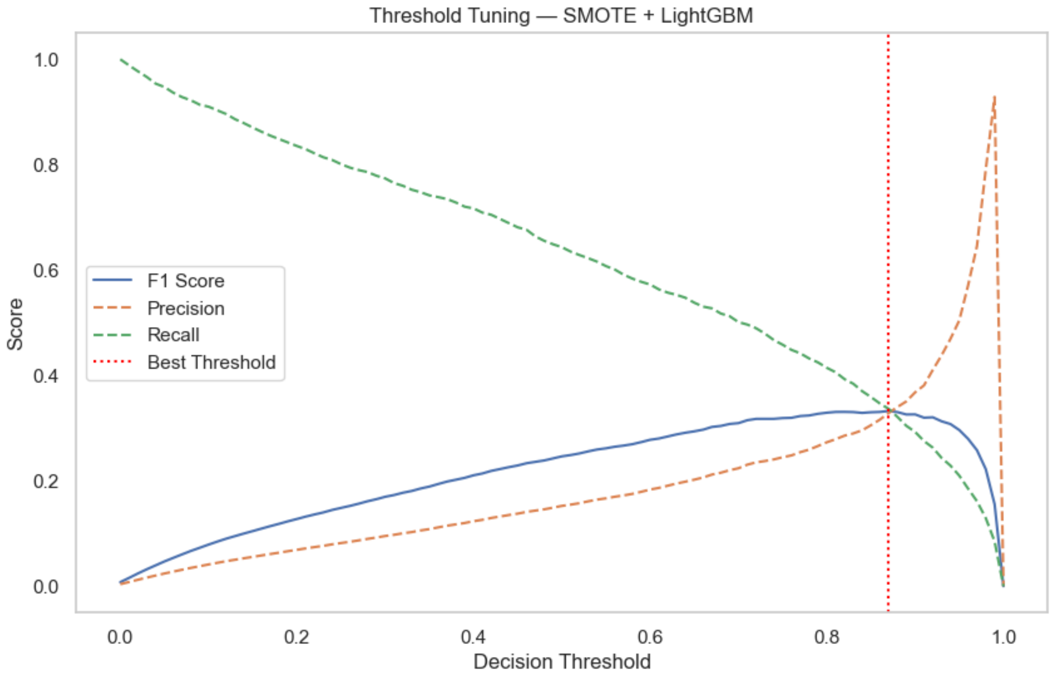
False positives were reduced sharply, making the model more suitable for production use.

04

Impact on User Experience

The model now favors high-confidence fraud predictions, minimizing unnecessary disruptions for genuine users.

A slight reduction in recall is considered acceptable given the operational benefits.



Metric	Default Threshold (0.50)	Tuned Threshold (0.87)
Precision (Fraud)	0.1524	**0.3286**
Recall (Fraud)	**0.6434**	0.3366
F1 Score (Fraud)	0.2464	**0.3326**
True Negatives (TN)	545,899	**552,099**
False Positives (FP)	7,675	**1,475**
False Negatives (FN)	**765**	1,423
True Positives (TP)	**1,380**	722

Global Interpretation with SHAP

1

Importance of Interpretation

SHAP provides insights into feature influence on model decisions.

This understanding is essential for building trust and enabling actionable outcomes.

2

Key Driver: Amount vs. State Average

The comparison of transaction amounts to state averages is a major factor in fraud detection.

Unusually high local amounts are significant indicators of fraudulent activity.

3

Risk Assessment: Transaction Amount (log)

Large transaction amounts inherently carry higher risk.

This feature is crucial in identifying potential fraud cases.

4

Merchant Activity Analysis

Fraudulent transactions often involve merchants with low activity and popularity.

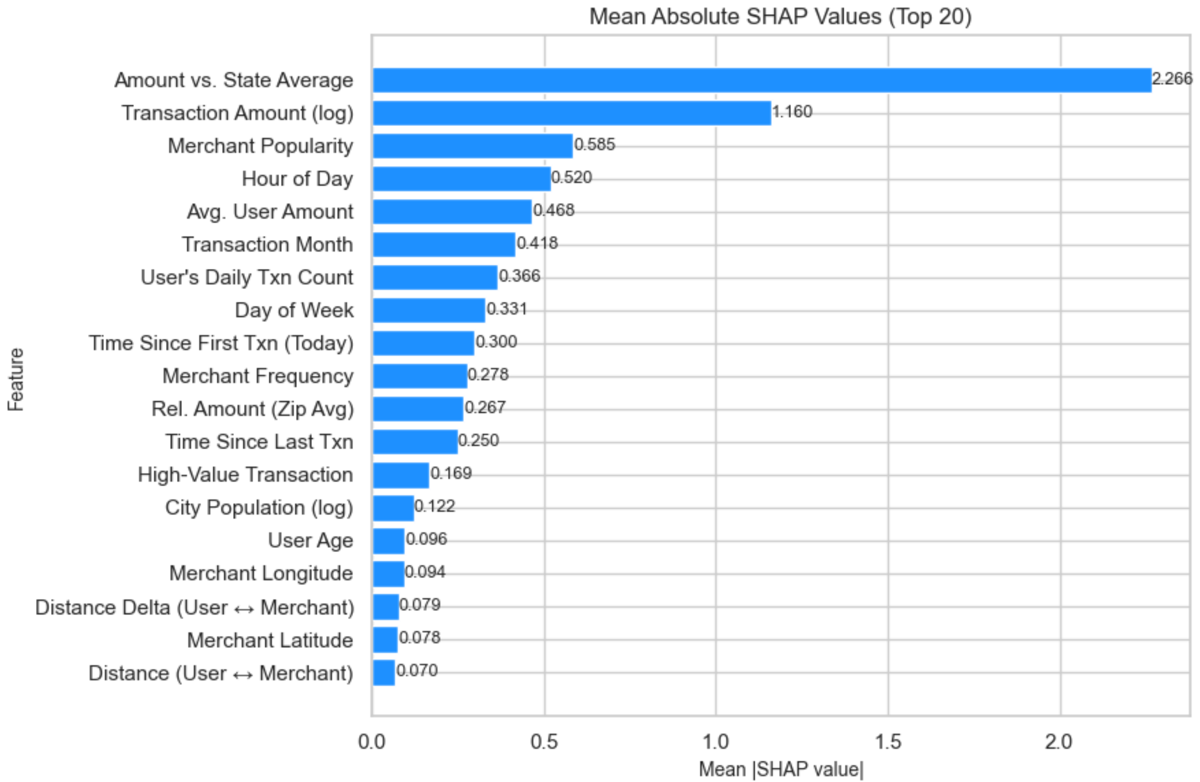
These patterns are indicative of targeted fraud schemes.

5

Behavior and Location Signals

Irregular transaction times and deviations in user behavior are suspicious.

Distance to merchant and user demographics also play a role in fraud detection.



Conclusion & Business Value

01 Developing Fraud Detection Model

Developed a high-precision fraud detection model tailored to the business's risk appetite.

This model is designed to effectively identify fraudulent activities while minimizing unnecessary alerts.

02 Reducing False Positives

Reduced false positives by 80% through threshold tuning, minimizing customer friction.

This improvement enhances the user experience and operational efficiency.

03 Ensuring Model Transparency

Ensured model transparency with SHAP values, aligning predictions with real-world intuition.

This approach builds trust and understanding in the model's decision-making process.

04 Revealing Fraud Patterns

Revealed behavioral, temporal, and geographic patterns that distinguish fraudulent activity.

These insights are crucial for understanding and combating fraud effectively.

05 Next Steps

Integrate into production, monitor with live data, and create feedback loops for continuous refinement.

Align outputs with human workflows for high-risk cases.