Machine Learning Based Credit Card Fraud Detection

Sun Han

Bellevue University

DSC680-T302 Applied Data Science (2257-1)

Jun 26, 2025

Week 4

Business Problem

Credit card fraud represents a persistent threat to financial institutions and merchants, causing substantial monetary losses each year. Static, rule-based detection systems struggle to keep pace with evolving fraud techniques, resulting in either undetected fraudulent transactions or excessive false alarms that frustrate legitimate customers. This project investigates how machine learning can provide a more adaptive and precise approach to fraud detection by learning from historical transaction patterns and distinguishing illicit behaviors from genuine cardholder activities.

Background and History

Fraud tactics have evolved alongside digital payment technologies. In the early days, card skimming and physical theft were the primary concerns, but as the volume of online transactions surged, fraudsters developed sophisticated schemes such as phishing, synthetic identities and coordinated cyberattacks. Initial fraud detection relied on manually crafted thresholds and rules, which lost effectiveness as criminals discovered new vulnerabilities. In the 2000s, researchers began applying machine learning algorithms that could automatically identify complex patterns in large transaction datasets. Ensemble methods and real-time analytics have since become promising alternatives to rule-based systems, offering improved adaptability and detection rates.

Data Explanation

This study uses a simulated dataset of 100,000 credit card transactions from Kaggle. Each record includes a unique identifier, timestamp, transaction amount, merchant identifier, transaction channel (online or in-person), geographic location and a binary fraud label.

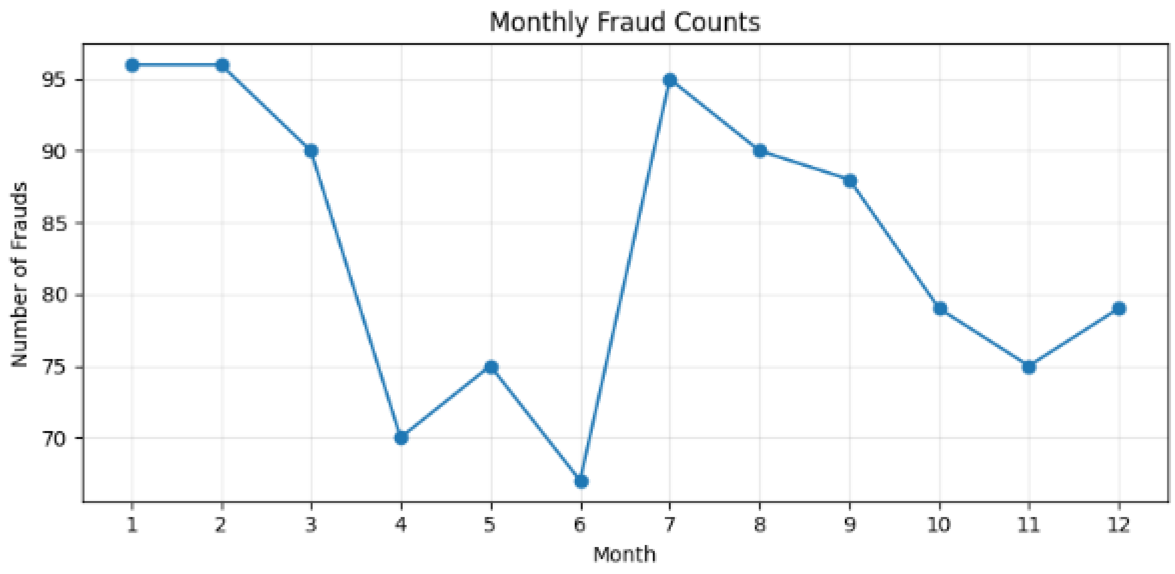
During preprocessing, temporal features such as hour of day, day of week and month were extracted. Categorical attributes were converted into numeric format using one-hot encoding, and transaction amounts were log-transformed to mitigate the influence of extreme values. Missing values in merchant or location fields were imputed using the most frequent category. Because fraudulent transactions comprised under one percent of the data, we applied both SMOTE and class-weight adjustments.

Methods

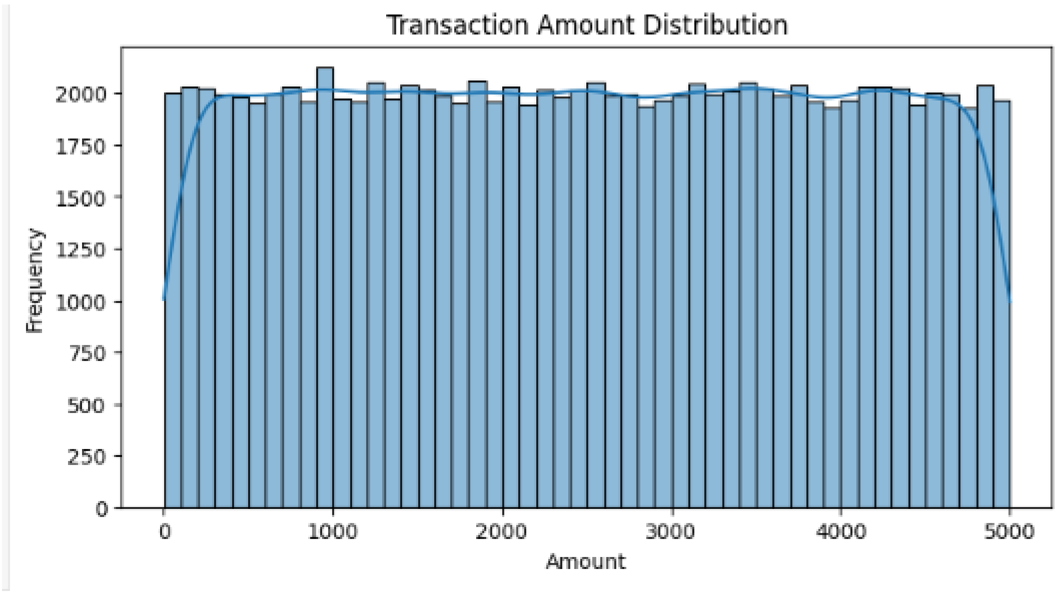
We evaluated two ensemble classifiers: Random Forest and XGBoost. Random Forest served as a baseline due to its robustness and interpretability, while XGBoost was chosen for its efficiency and proven performance on imbalanced data. We split the dataset 80/20 into training and test sets and conducted five-fold cross-validation during hyperparameter tuning. Model performance was assessed with precision, recall, F1-score and ROC-AUC metrics. For XGBoost, we experimented with both a scale_pos_weight parameter and SMOTE oversampling to address imbalance.

Analysis

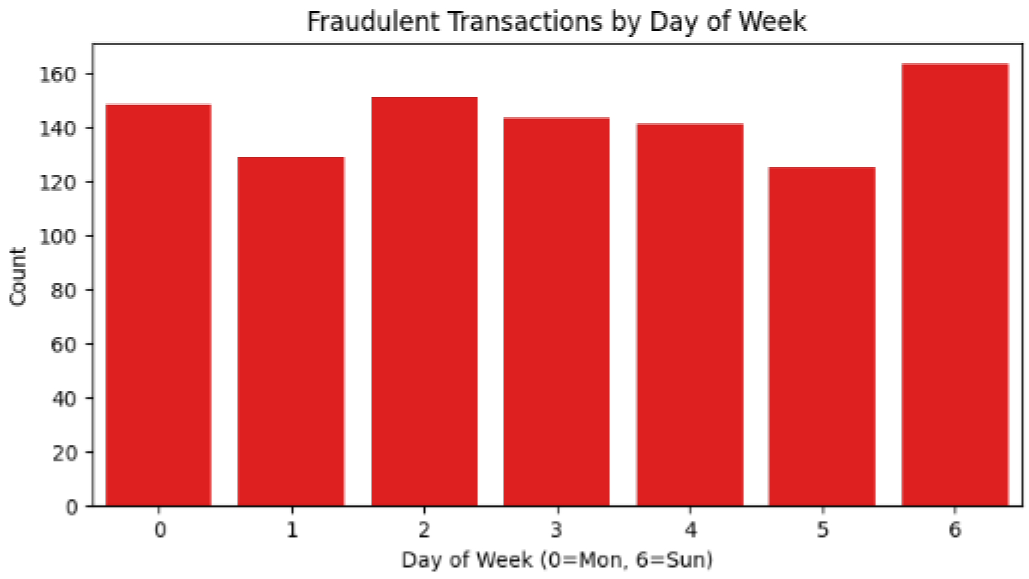
Several visuals were created to explore the dataset.



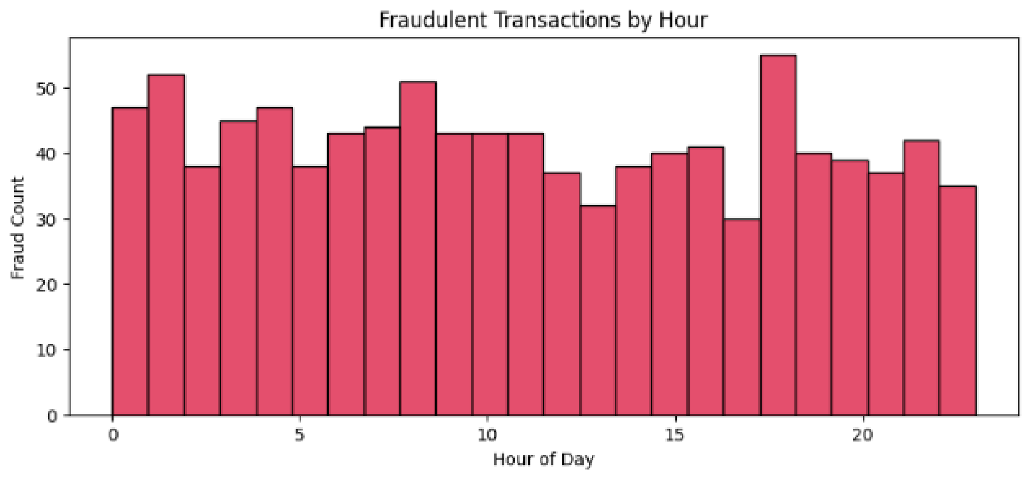
From the graph, January, February, and July have the highest fraud counts.



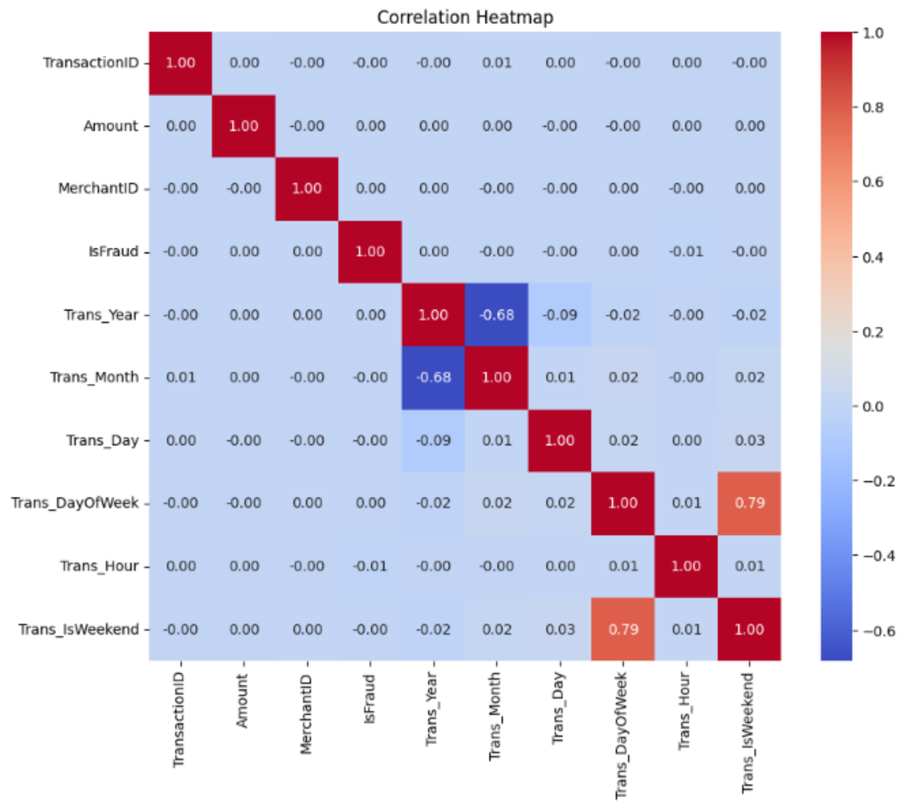
The relatively uniform distribution across most of the range implies that most transactions fall into a broad, predictable range, which could be a sign of consistent consumer behavior.



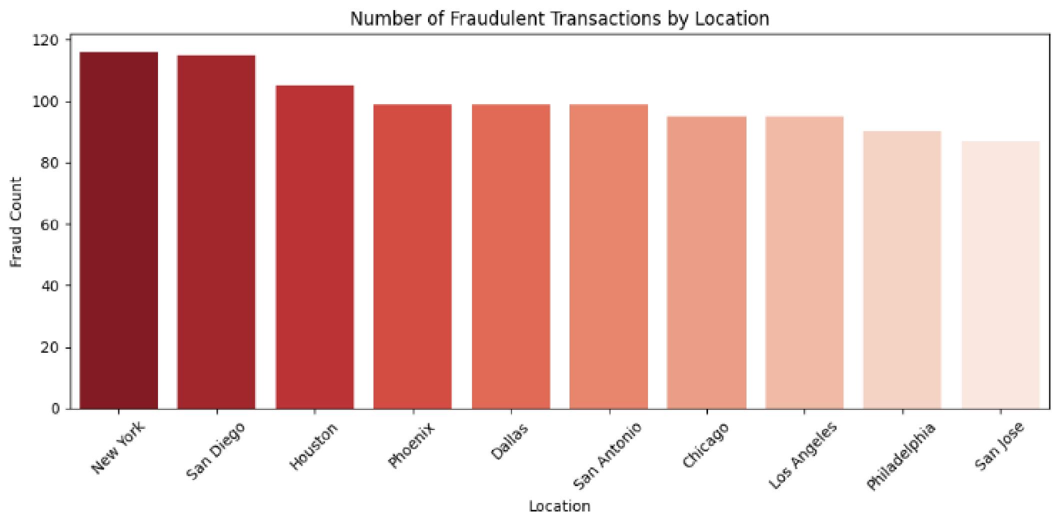
Sundays have a higher fraud count compared to other days.



There are noticeable increases in fraudulent transactions around 1 AM and 4 PM.



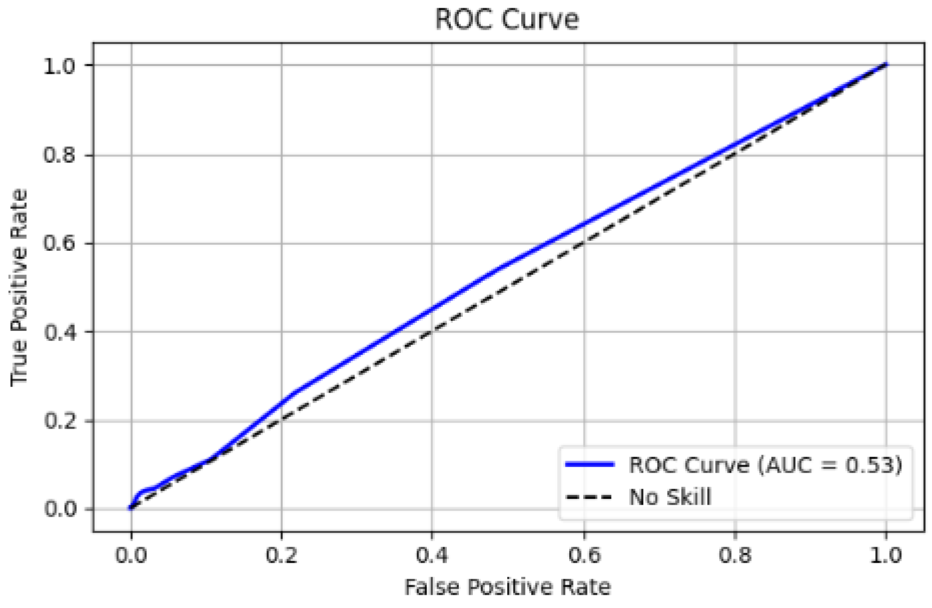
From the correlation heatmap, fraud isn't obviously driven by basic transaction features like time, merchant, or amount alone. That's a sign that more complex patterns are behind fraud activity.



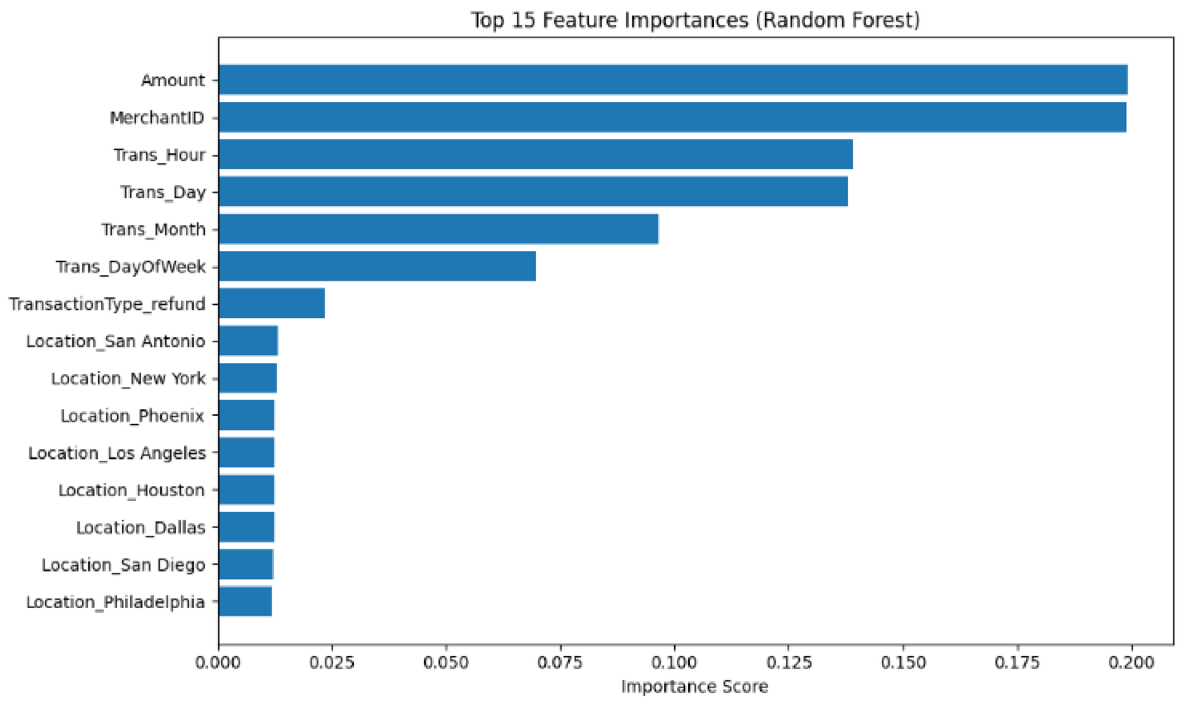
New York and San Diego are leading the list, each with close to 120 fraud cases, signaling that these cities may require more stringent fraud prevention or monitoring.

The initial random forest model achieved high overall accuracy but failed to detect any fraud cases. The confusion matrix shows that although 19,800 non-fraud transactions are correctly classified, every one of the 200 fraud cases is misclassified as non-fraud. The classification report confirms that while performance for non-fraud is excellent with high precision and recall, the model registers zero precision, recall, and F1 score for fraud. These results underscore the challenges of class imbalance, as the overwhelming number of non-fraud samples skews accuracy while leaving the minority fraud class undetected.

```
Confusion Matrix:
[[19800     0]
 [  200     0]]

 Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     19800
           1       0.00      0.00      0.00       200

    accuracy                           0.99     20000
   macro avg       0.49      0.50      0.50     20000
weighted avg       0.98      0.99      0.99     20000

Precision (fraud): 0.0
Recall (fraud): 0.0
F1 Score (fraud): 0.0
```

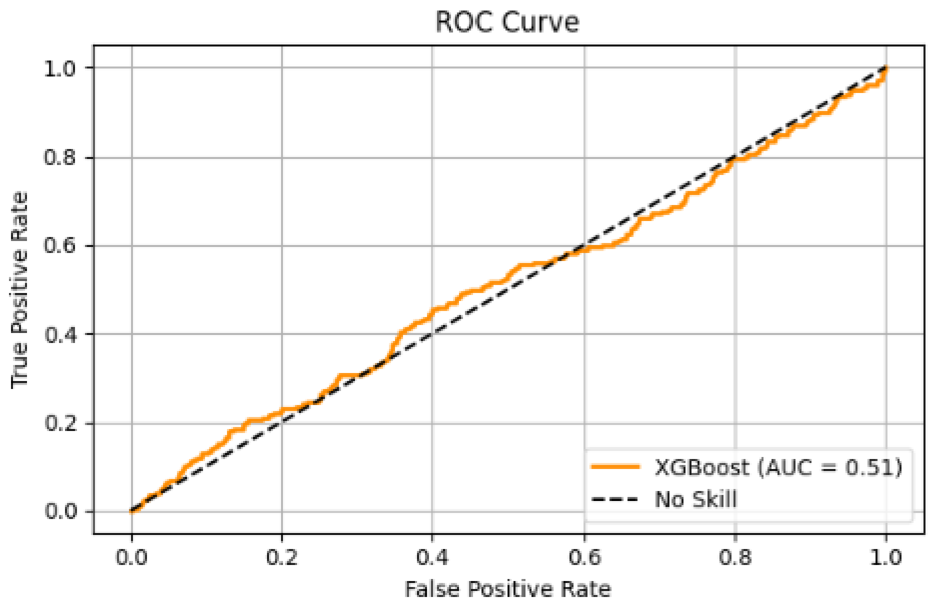


```
ROC AUC Score: 0.53
```



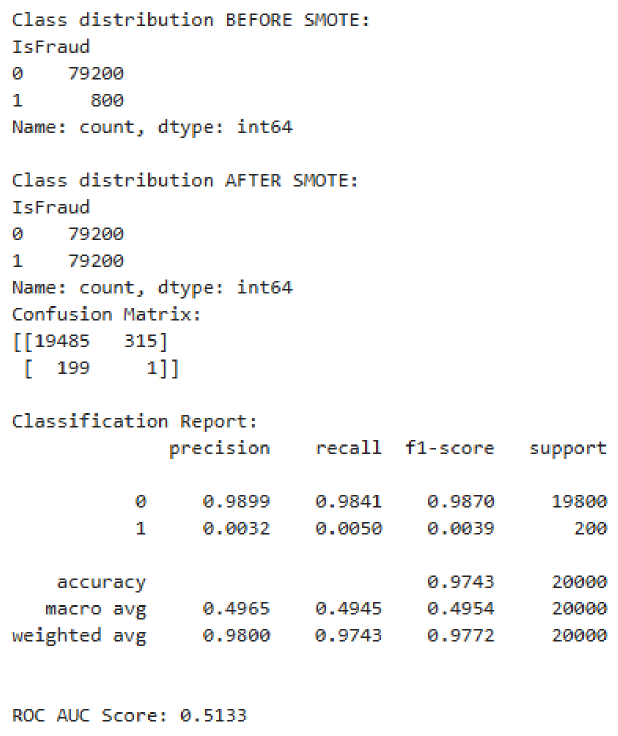
After applying class imbalance handling using scale_pos_weight in the XGBoost model, the confusion matrix shows that 16,873 non-fraud cases are correctly identified while 2,927 non-fraud cases are misclassified as fraud. Out of 200 fraud cases, only 39 are correctly detected and 161 are missing. The classification report indicates that the fraud precision is extremely low at 0.0131 and the recall is 0.195, meaning that although the model now identifies some fraud cases, it produces a substantial number of false positives and still misses most fraudulent transactions. The overall accuracy is reduced to 84.56 percent, suggesting that further adjustments are needed to improve fraud detection performance.

```
Confusion Matrix:
 [[16873  2927]
 [  161    39]]

Classification Report:
              precision    recall  f1-score   support

           0     0.9905    0.8522    0.9162     19800
           1     0.0131    0.1950    0.0246       200

    accuracy                         0.8456     20000
   macro avg     0.5018    0.5236    0.4704     20000
weighted avg     0.9808    0.8456    0.9072     20000

Precision (fraud): 0.0131
Recall (fraud): 0.195
F1 Score (fraud): 0.0246
```

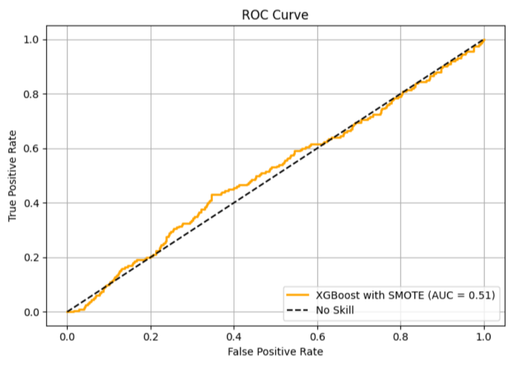


```
ROC AUC Score: 0.5067
```

After applying SMOTE to balance the training data, the original class distribution of 79,200 non-fraud and 800 fraud instances equalized to 79,200 for each class. Training XGBoost on this balanced dataset produced a confusion matrix where 19,485 non-fraud cases were correctly predicted and 315 non-fraud cases were incorrectly labeled, while for fraud only 1 case was correctly identified, and 199 fraud cases were misclassified. Despite the overall accuracy being high at 97.43 percent, the precision for fraud is extremely low at 0.0032, with a recall of 0.0050, indicating that the model still has major difficulty detecting fraudulent transactions. The ROC AUC score of 0.5133 further suggests that the model's performance in distinguishing between fraud and non-fraud is only marginally better than random chance.



```
Class distribution BEFORE SMOTE:
IsFraud
0    79200
1      800
Name: count, dtype: int64

Class distribution AFTER SMOTE:
IsFraud
0    79200
1    79200
Name: count, dtype: int64
Confusion Matrix:
[[19485   315]
 [  199     1]]

Classification Report:
              precision    recall  f1-score   support

           0     0.9899    0.9841    0.9870     19800
           1     0.0032    0.0050    0.0039       200

    accuracy                         0.9743     20000
   macro avg     0.4965    0.4945    0.4954     20000
weighted avg     0.9800    0.9743    0.9772     20000


ROC AUC Score: 0.5133
```



Conclusion

In our experiments, neither Random Forest nor XGBoost delivered adequate fraud-detection performance under the current feature set and sampling strategies. The Random Forest model trained on imbalanced data failed to correctly flag any fraudulent transactions. XGBoost tuned with a scale_pos_weight parameter identified fewer than one in five fraud cases and achieved a precision barely above one percent, resulting in an unacceptably high false-positive rate. After applying SMOTE, XGBoost's ROC-AUC remained near 0.51 and recall stayed near zero, effectively equating its behavior to random guessing. These results demonstrate that addressing class imbalance alone cannot compensate for features that lack strong discriminatory power, and more foundational improvements are required.

Assumptions

This study assumes that the simulated dataset represents real-world transaction behaviors and fraud patterns closely enough to train generalizable models. It also assumes that customer transaction patterns remain relatively consistent over time, which allows for effective model learning.

Limitations

Despite its utility, the simulated dataset may not reflect all the complexities found in actual financial data. Real-world fraud may involve contextual information such as device data, user behavior, or multi-channel patterns that are not captured here. The limited scope of features in the dataset may reduce the model's precision in certain edge cases.

Challenges

One of the main challenges encountered was the severe class imbalance in the dataset, with fraudulent transactions making up less than one percent of all records. Additionally, fraud patterns change frequently, requiring regular updates and monitoring of the model. Ensuring model performance remains stable over time is a continuous challenge that must be addressed through retraining and feedback loops.

Future Uses and Additional Applications

The models and techniques explored in this project have applications beyond credit card fraud. They can be extended to digital banking, online payment systems, and cryptocurrency transactions. With the integration of additional data sources such as device fingerprinting and geolocation, these models could be part of a broader fraud prevention strategy. Real-time deployment could further enhance security by detecting and stopping fraud as it occurs.

Recommendations

Financial institutions should consider implementing machine learning models as part of a hybrid fraud detection system that combines rule-based logic with predictive analytics. Human review processes should be integrated to evaluate flagged transactions. Institutions should also develop infrastructure for regular model evaluation and retraining to account for changing data patterns and new fraud tactics.

Implementation Plan

The proposed rollout begins with a pilot phase using historical transaction data to validate model performance. Once validated, the model will score transactions in batch mode, with a review pipeline for flagged cases. Full production deployment follows, supported by ongoing performance monitoring and quarterly retraining schedules to accommodate changing fraud patterns.

Ethical Assessment

Any deployment must safeguard customer privacy and promote fairness. Data should be anonymized and only essential features retained. Flagged transactions must undergo human review to prevent unwarranted suspicions. Bias detection and mitigation processes should be integrated into development, ensuring no demographic group faces disproportionate scrutiny. Full transparency about model logic and decision criteria will foster regulatory compliance and customer confidence.

Anticipated Audience Questions

1. How does the model handle new fraud patterns that were not seen during training?

We would implement a continuous learning pipeline: incoming transactions flagged by current model feed into a human-reviewed buffer. Confirmed new fraud patterns are added to the training set for periodic model retraining, ensuring adaptability over time.

2. How will false positives be managed to avoid frustrating legitimate customers?

Flagged transactions would first undergo automated secondary checks. Only high-confidence flags proceed to human review or customer challenge, minimizing inconvenience to legitimate users.

3. What is the performance difference between Random Forest and XGBoost in this case?

Random Forest detected 0% of fraud (all 200 cases missed). XGBoost with scale_pos_weight detected 19.5% of fraud at the cost of many false positives, showing a modest improvement but still insufficient precision.

4. Can this model be deployed in real-time and still maintain high performance?

XGBoost inference latency is low and can be parallelized, making real-time scoring practical. We recommend batch pilot mode first, then integrate into streaming architecture.

5. How do we defend against adversarial fraud that mimics normal transactions?

Employ adversarial training by simulating known evasion techniques. Monitor feature distributions for anomalies. Implement a fraud feedback loop where analysts share new tactics to update both rules and model training data.

6. What is the schedule for model retraining and performance evaluation?

Quarterly full retraining on the latest data, with monthly performance snapshot reports. Trigger ad-hoc retraining if key metrics (e.g. recall or ROC-AUC) drop by more than 5 percentage points.

7. Are there additional features, such as device or behavior data, that could improve results?

Integrate device fingerprinting, IP geolocation, browser/user-agent patterns, and past customer behavior sequences. Consider network analysis of merchant relationships and peer-group spending habits.

8. What mechanisms are in place to detect and handle data drift over time?

Implement automatic drift detection: monitor feature and prediction distributions. When drift is detected, alert the data science team for investigation and potential retraining.

9. How is this machine learning approach superior to rule-based systems in the long term?

ML models learn complex, non-linear interactions and adapt to new patterns, whereas rules are static and require manual updates. Over time, ML reduces manual maintenance and can generalize unseen fraud behaviors.

10. How will customer data privacy and ethical concerns be addressed during deployment?

Anonymize or tokenize personally identifiable information. Use privacy-preserving techniques (e.g. differential privacy) if needed. Ensure flagged cases undergo human review to prevent algorithmic bias and maintain transparency on decision criteria.

References

1. Kaggle. (n.d.). Credit Card Fraud Detection dataset. Retrieved from https://www.kaggle.com/datasets/bhadramohit/credit-card-fraud-detection