Fraud Detection Analysis

Exploratory Analysis, Fraud Detection Rules and Model Evaluation

Aswathy Nandakumar

February 1, 2025

1. Executive Summary

1. Introduction

This mock analysis outlines a structured **fraud detection strategy** using model improvements, feature selection and fraud prevention rules. Based on a **simulated dataset** mimicking real-world fraud patterns, it evaluates old vs. new detection models, recommends prevention rules and identifies key fraud patterns through data-driven insights.

Key Questions Answered:

- Which features distinguish fraudulent vs genuine orders?
- How does new fraud detection model compare to old one?
- What rules can be implemented to detect and prevent fraud effectively?

2. Exploratory Data Analysis

Key Insights:

- **Fraud Distribution**: In this simulated dataset, fraud cases were intentionally imbalanced (7.4% of transactions) to reflect real-world fraud detection challenges. **SMOTE** (Synthetic Minority Oversampling Technique) applied. *[Fraud vs Genuine Transactions plot]*
- **Feature Selection**:
  - **Top fraud indicators**: *newModelScore* (0.50), *oldModelScore* (0.36), *orderNumberFeature42* (0.21), and *skuCountFeature47* (0.10). *[correlation matrix] and [feature correlation values]*
- **Time-Based Fraud Trends**:
  - Fraud peaks in between **4 PM - 9 PM**, with highest cases on **Wednesdays & Thursdays**. *[Time-based plots for fraudsters]*
  - Suggests need for increased fraud monitoring in peak hours.

Recommendations:

- Prioritize highly correlated features in fraud detection model.
- Apply dynamic fraud thresholds to counter peak fraud periods.
- Ensure dataset balance to prevent bias in fraud detection.

3. Model Performance Evaluation

Old vs. New Model Comparison:

- **Fraud Score Distribution**: New model assign higher fraud scores to confirmed fraudsters, increasing detection accuracy. *[Fraud Score Distributions of Old vs New Models]*
- **Precision-Recall & AUC-ROC**:
  - The model comparison demonstrated improvements in fraud detection, with a higher AUC score indicating better fraud separation. *[Precision-Recall Curve]* and *[ROC]*
  - **Threshold Analysis**: At **0.1 threshold**, new model detect **692 fraud cases**, outperforming the old model (**625 fraud cases**). *[Threshold Analysis for Old vs New Models]*
- Key Takeaway: **New model is superior** in detecting fraud with **fewer false positives**.

Recommendations:

- **Adopt new model** for fraud detection.
- **Optimize fraud score thresholds** to balance fraud prevention and customer experience.

- **Monitor false positives** to ensure genuine transactions are not flagged wrongly.

4. Fraud Prevention Rules & Performance Evaluation

Best Performing Rules:

- **Rule A** (Strict, High-Precision)
  - Capture fraud based on: **Frequent orders (5+/day), high fraud scores, unusual payments, peak fraud hours.**
  - **Precision: 76.83%** (low false positives), but **Recall: 15.75%** (misses some fraud).
- **Rule D** (Balanced, High Recall)
  - Flags fraud with **4+ orders in 5 hours**, strict payment checks, and new accounts making high-value purchases.
  - **Precision: 21.60%**, **Recall: 25.25%** (catch more fraud cases).
- Rules B and C were discarded due to poor performance evaluation metrics
  [Discarded rules, Rule B and Rule C included in Appendix]

Recommendations:

- Combine **Rule A** & **Rule D** to balance fraud detection & minimize false positives.
- Apply **dynamic risk-based scoring** to refine fraud detection threshold.
- Continuously update fraud rules based on evolving fraud patterns.

5. Business Recommendations

Key Business Takeaways:

- **New Model Outperforms** old model with better fraud detection & fewer false positives.
- **Fraud Patterns** Identified**: Peak fraud hours (4 PM - 9 PM), high risk days (Wed & Thu),** and **new account activity.**
- **Effective Fraud Prevention Rules: Rule A (high precision) + Rule D (high recall**) provide the best fraud detection tradeoff.

Recommendations:

- **Optimize fraud thresholds** dynamically based on real-time fraud trends.
- Enhance model with **geolocation** & **transaction velocity checks**.
- **Refine fraud** rules continuously to adapt to new fraud patterns.
- **Strengthen monitoring** in high-risk periods.

6. Conclusion

This analysis provides **structured fraud detection strategy** leveraging **model improvements**, **feature selection** and **fraud prevention rules**. By adopting a **hybrid approach** (ML + Rules), e-commerce company can improve fraud prevention, reduce false positives and improve customer trust.

This executive summary provides a high-level overview of the findings and recommendations. The following sections delve into the detailed analysis.

2. Introduction

Purpose of the Analysis

This mock analysis outlines a structured **fraud detection strategy** using model improvements, feature selection and fraud prevention rules. Based on a **simulated dataset** mimicking real-world fraud patterns, it includes genuine and fraudulent transactions with fraud scores from both an old and newly trained machine learning model.

Key Business Questions

- What feature patterns distinguish fraudulent vs. genuine orders?
- How does the new fraud detection model compare to the old one?
- What rules can be implemented to catch fraudsters effectively?

3. Dataset Overview & Data Cleaning

Summary of Work Done

To enhance the reliability and quality of the dataset for fraud detection, several preprocessing techniques were applied. The key steps undertaken include:

- **Dropping unnecessary columns**: Removed identifier columns, duplicate columns (keeping one copy), and features with excessive missing values.
- **Handling missing values and placeholders**: Replaced placeholder values (-9999999) with NaN and filled missing numerical values using the median.
- **Transforming categorical features**: Converted categorical variables such as *isEWallet* into a numerical format to facilitate analysis.
- **Standardizing date formats**: Converted *orderTime* to datetime format for time-based fraud analysis.

These steps ensured that the dataset was clean, structured, and optimized for accurate fraud detection analysis.

Findings

Dataset Overview

- The dataset contains numerical, categorical, and timestamp-based features relevant to fraud detection analysis. *[Appendix - Figure 1, Figure 4]*
- Initial data inspection show **mix of numerical, categorical** and **timestamp-based features** requiring conversions to ensure compatibility for fraud modeling. *[Appendix - Figure 1, Figure 2, Figure 4]*

Handling Missing Data

- Several features had **high missing values**, requiring removal or imputation. *[Appendix - Figure 5]*
- Placeholder values (-9999999) were identified and replaced with NaN.
- **Numerical missing values were filled with the median** to prevent bias.
- Features Removed Due to Excessive Missing Data: *skuPopularityFeature21*, *skuPopularityFeature24*, *skuPopularityFeature35*, *anonymousFeature99*, *isEWallet* (~95% missing data)

Duplicate Data Handling

- **No duplicate rows** were found in the dataset. *[Appendix - Figure 6]*

- **One duplicate column** (*accountAgeFeature24*) was removed as it was identical to *accountAgeFeature12*. *[Appendix - Figure 6]*

Insights & Recommendations

- **Handling missing values** helped maintain dataset consistency in this mock project, ensuring a more structured fraud detection analysis.
- **Removing duplicate and irrelevant columns** enhanced dataset efficiency without losing critical information.
- **Standardizing datetime and categorical variables** improved feature usability for fraud analysis.

For future fraud detection datasets:

- **Minimize missing values** through better data collection strategies.
- **Ensure dataset consistency** by avoiding redundant features.
- **Retain well-structured categorical and datetime fields** for better fraud trend analysis.

Final Notes on Data Preprocessing

The dataset is now cleaned, structured, and ready for fraud detection analysis. These refinements directly contribute to better fraud pattern identification and improved model accuracy.

4. Exploratory Data Analysis (EDA)

Summary of Work Done

The exploratory data analysis focused on understanding fraud trends within the dataset, selecting the most important features for fraud detection and identifying time-based fraud patterns. The following steps were performed:

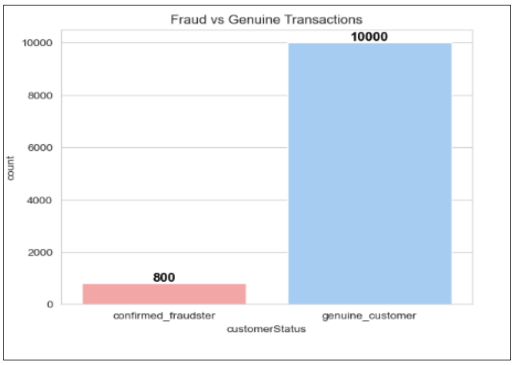
- **Fraud vs. Genuine Transaction Distribution**: Analyzed ratio of fraudsters to genuine customers, highlighting dataset imbalance.
- **Feature Correlation Analysis**: Identified features with the highest positive and negative correlation to fraud.
- **Time-Based Fraud Patterns**: Investigated fraud trends across different time periods (hourly, daily, and over dataset's timeframe).
- **Threshold Analysis Consideration**: performed in the model comparison part which includes role of model thresholds in real-time fraud detection.

Each of these analyses was supported by appropriate visualizations.

Findings and Visualizations

Fraud vs. Genuine Transaction Distribution

- **Fraud** cases account for **800** transactions compared to **10,000 genuine transactions**, making the dataset highly **imbalanced**.



- To ensure that this imbalance does not skew the analysis, we applied **Synthetic Minority Oversampling Technique (SMOTE)** as an investigative step.
    - **Before SMOTE:** Fraud = **640**, Genuine = **8,000**
    - **After SMOTE:** Fraud = **8,000**, Genuine = **8,000**
- SMOTE confirmed class imbalance did not distort fraud-driving features, reinforcing the robustness of our detection strategy while keeping the original dataset for real-world accuracy.

Feature Correlation Analysis

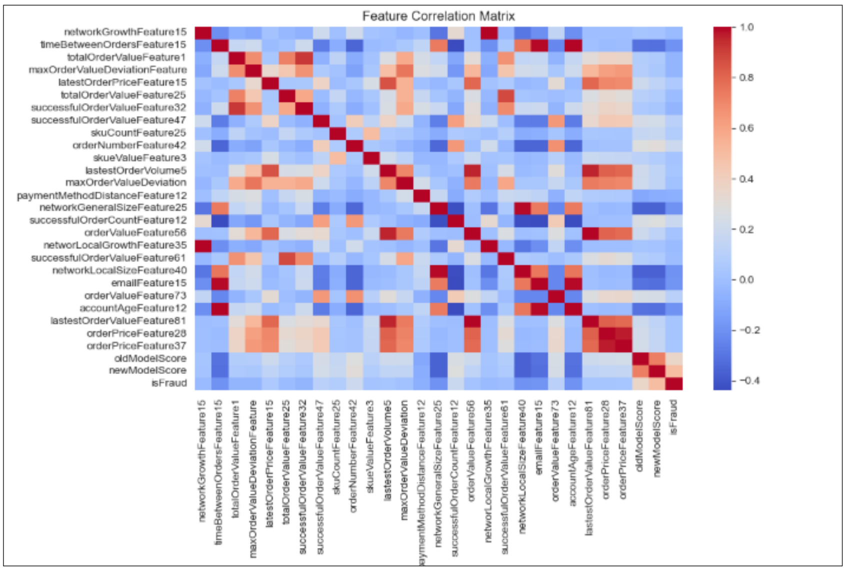
Correlation analysis revealed that the following features have **strongest positive** correlation with fraud and should be included in a fraud detection model:

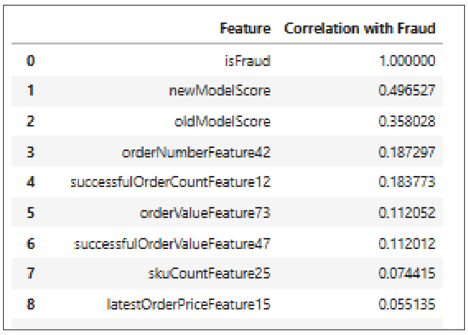
- *newModelScore* (0.50) – Strongest fraud indicator
- *oldModelScore* (0.36) – Still relevant, but not as strong as *newModelScore*
- *orderNumberFeature42* (0.21) – More orders = higher fraud risk
- *successfulOrderCountFeature73* (0.11) – Prior success doesn't always mean genuine
- *skuCountFeature47* (0.10) – More items per order may be fraud

Additionally, some features were found to be **negatively correlated** with fraud, indicating patterns more common among genuine customers:

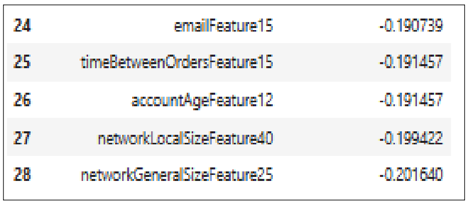
- *timeBetweenOrdersFeature15* (-0.19) – Legitimate users have longer time gaps
- *networkLocalSizeFeature25* (-0.19) – Smaller networks are often fraudsters
- *networkGeneralSizeFeature15* (-0.20) – same observation as above

These negatively correlated features can help **reduce false positives** by distinguishing legitimate customers from fraudsters.





	Feature	Correlation with Fraud
0	isFraud	1.000000
1	newModelScore	0.496527
2	oldModelScore	0.358028
3	orderNumberFeature42	0.187297
4	successfulOrderCountFeature12	0.183773
5	orderValueFeature73	0.112052
6	successfulOrderValueFeature47	0.112012
7	skuCountFeature25	0.074415
8	latestOrderPriceFeature15	0.055135



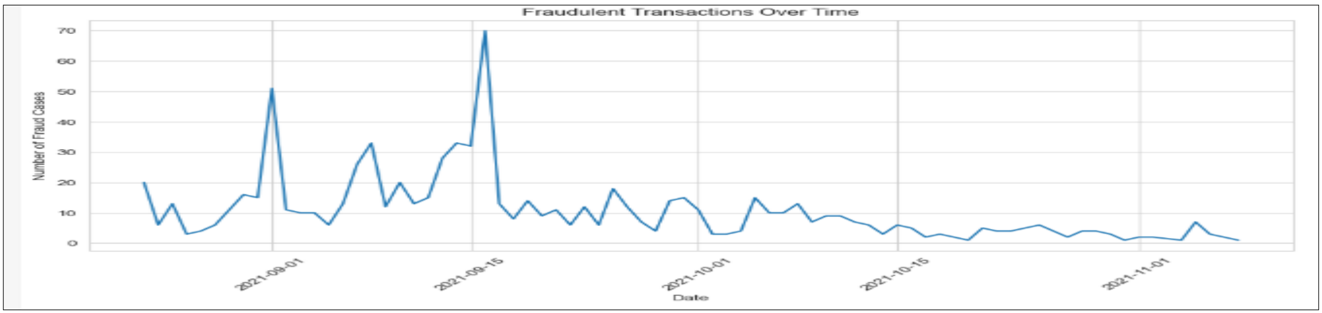
24	emailFeature15	-0.190739
25	timeBetweenOrdersFeature15	-0.191457
26	accountAgeFeature12	-0.191457
27	networkLocalSizeFeature40	-0.199422
28	networkGeneralSizeFeature25	-0.201640

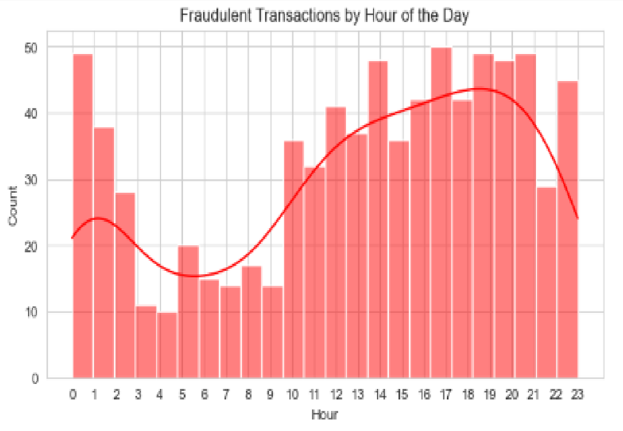
Time-Based Fraud Patterns

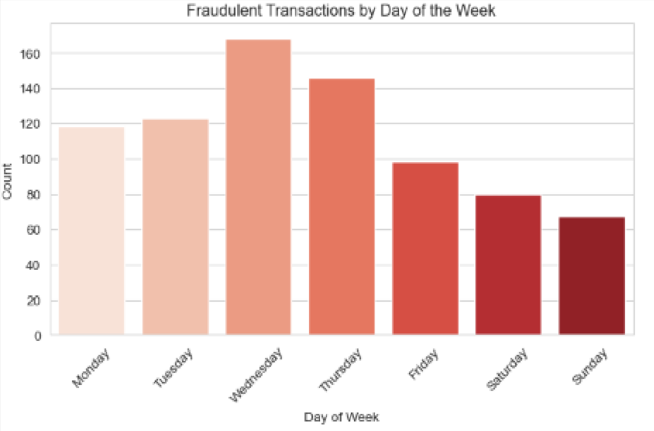
Analyzing fraud trends over time provided key insights:

- **Fraudulent activity peaks** in the **late afternoon and evening (4 pm - 9 pm)**, suggesting fraudsters operate during high-traffic hours.
- **Wednesdays and Thursdays** show **higher fraud rates** compared to other days.
- Fraud spikes were observed on specific dates, potentially indicating **organized fraud attacks**.

These findings suggest that **fraud detection systems should increase monitoring and trigger additional verification checks during high-risk periods**.







Insights

- **Dataset imbalance is significant**, meaning fraud detection models must be **trained carefully** to avoid bias toward genuine transactions.
- **Feature correlation analysis identified strong predictors of fraud**, helping in model feature selection.
- **Time-based fraud analysis shows clear patterns**, suggesting that fraud detection strategies should be **adaptive based on time of day and day of the week**.
- **Threshold analysis is key** here in fraud prevention since setting the right fraud score threshold determines how many fraudulent transactions are blocked.

Recommendations

- **Prioritize high-correlation features in fraud detection model**:
    - Include ***newModelScore*** and ***oldModelScore*** as primary indicators.
    - Utilize **order-related features** like ***orderNumberFeature42*** to detect unusual order behaviours.
- **Deploy stricter fraud detection thresholds for new merchants**:
    - Since **fraud rates tend to be high from start**, **new merchants should have lower fraud threshold initially** to block suspicious transactions early.
    - Threshold settings should be continuously **optimized based on fraud detection performance**.
- **Enhance fraud monitoring during peak hours and high-risk days**:
    - Increase **monitoring and verification checks between 4:00 pm - 9:00 pm** when fraud is most active.
    - Implement additional security **on Wednesdays and Thursdays** when fraud attempts are more frequent.
- **Adjust detection models to handle dataset imbalance**:
    - Class imbalance analysis confirmed that fraud-related features remained consistent, even after applying SMOTE-based balancing.
    - Final fraud detection rules and model comparisons can be performed on original dataset to maintain real-world accuracy.

Final Notes on EDA

The exploratory data analysis provides **critical insights into fraud detection**, enabling the identification of the **most relevant features, peak fraud periods,** and **optimal prevention strategies**. These findings directly inform model development, fraud rule design, and real-time fraud detection improvements.

5. Model Comparison: Old vs New Fraud Detection Models

Summary of Work Done

The old and new fraud detection models were evaluated based on their fraud score distributions, threshold-based fraud detection, and overall model performance. The analysis included:

- **Fraud Score Distribution Analysis**: Comparing score distributions for fraudulent and genuine transactions.
- **Threshold-Based Performance**: Assessing how fraud detection varies at different score thresholds.
- **Precision-Recall and AUC-ROC Analysis**: Evaluating model accuracy in detecting fraud.
- **Key Insights and Recommendations**: Determining which model performs better in real-world fraud detection scenarios.

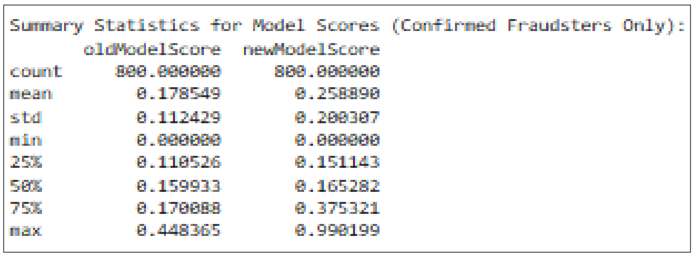
These evaluations help determine the most effective fraud detection strategy.

Findings and Visualizations

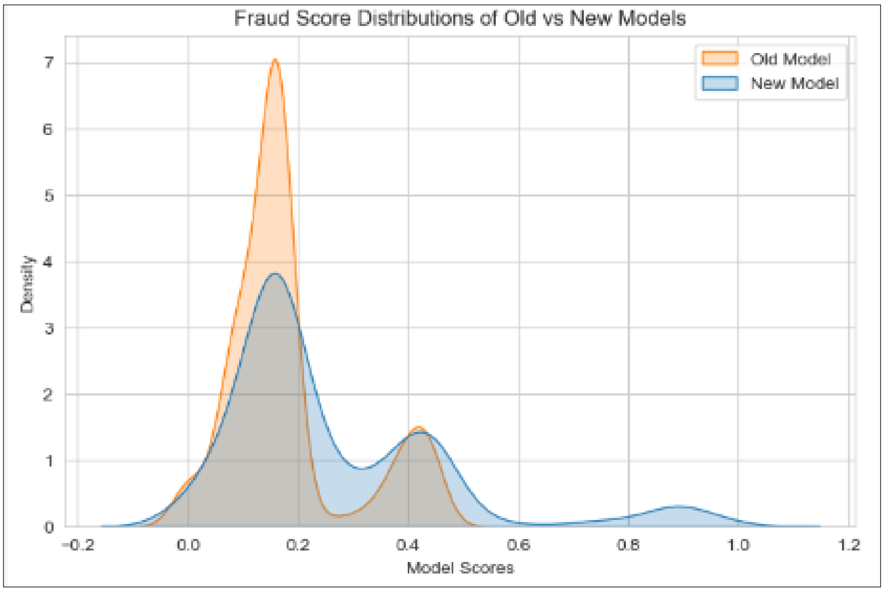
Fraud Score Distributions

The **Kernel Density Estimation (KDE) plots** show how the models assign fraud scores to transactions.

- **Overall Score Comparison**
  - The **new model** has a **wider score distribution**, capturing a more diverse fraud profile.
  - The **old model** clusters scores more tightly, suggesting **less separation between fraudulent and genuine cases**. *[Appendix - Figure 7]*
- **Confirmed Fraudsters**
  - The **new model assigns higher fraud scores** to confirmed fraudsters (mean score: **0.259**) than the old model (**0.179**). (refer image below showing summary stats for fraudsters)
  - A broader score spread suggests **better fraud risk differentiation**. *(refer KDE plot below)*
- **Genuine Customers**
  - The old model has a slightly higher mean fraud score (**0.078**) for genuine customers compared to the new model (**0.074**), suggesting it might be **less effective at minimizing false positives**. *[Appendix - Figure 8]*

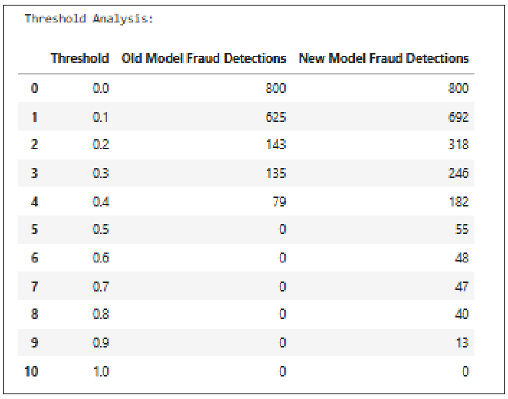


```
Summary Statistics for Model Scores (Confirmed Fraudsters Only):
       oldModelScore  newModelScore
count     800.000000     800.000000
mean        0.178549       0.258890
std         0.112429       0.200307
min         0.000000       0.000000
25%         0.110526       0.151143
50%         0.159933       0.165282
75%         0.170088       0.375321
max         0.448365       0.990199
```



Threshold-Based Fraud Detection

- Fraud detection models use a **threshold-based approach** to flag transactions as fraudulent.
- The threshold analysis shows that **the new model captures more fraud cases at higher thresholds**, demonstrating **better fraud separation**.
- Example: At a **0.1 threshold**, the **new** model detected **692 fraud cases**, while the **old** model detected **625**.

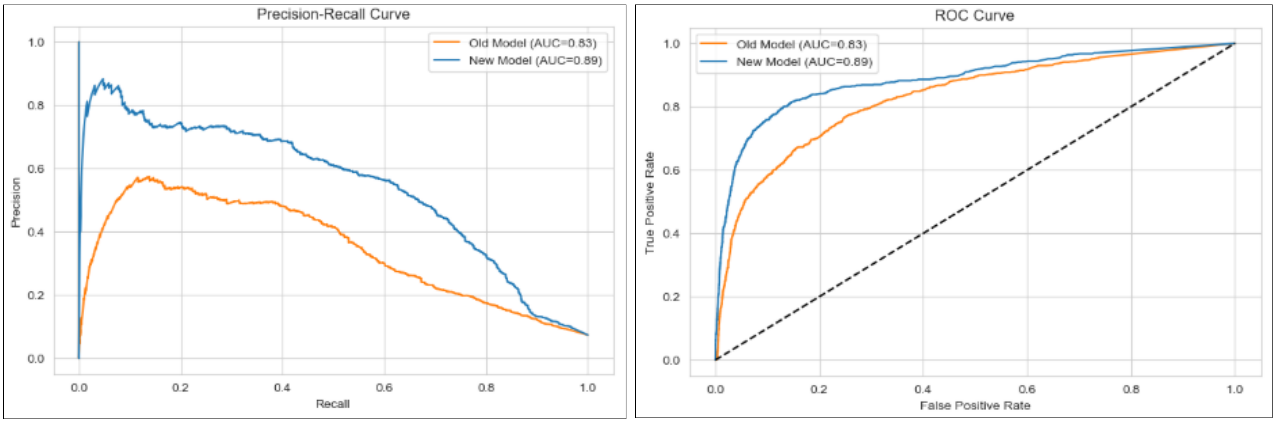


Threshold Analysis:

	Threshold	Old Model Fraud Detections	New Model Fraud Detections
0	0.0	800	800
1	0.1	625	692
2	0.2	143	318
3	0.3	135	246
4	0.4	79	182
5	0.5	0	55
6	0.6	0	48
7	0.7	0	47
8	0.8	0	40
9	0.9	0	13
10	1.0	0	0

Precision-Recall & AUC-ROC Analysis

- **Precision-Recall Curve**: The new model **maintains better recall** while achieving comparable precision to the old model.
- **AUC-ROC Analysis**: The **new model** (**AUC = 0.886**) outperforms the **old model** (**AUC = 0.829**), indicating **superior fraud detection capability**.
- **Key takeaway**: The new model achieves **better fraud separation with fewer false positives**.



Insights

- The **new model assigns higher fraud scores to actual fraudsters**, improving fraud detection accuracy.
- **Threshold analysis** confirms that the **new model captures more fraud cases** at various cutoff points.
- **AUC-ROC and Precision-Recall metrics** demonstrate the **new model's superiority**, making it the preferred choice.
- The **old model tends to assign higher fraud scores to genuine customers**, which could lead to **more false positives**.
- The **new model is better suited for real-world fraud detection**, but score threshold tuning is required.
- The **old model tends to flag more genuine customers** as fraud, which could lead to customer dissatisfaction.

Recommendations

- **Adopt the New Model** for fraud detection, as it consistently **outperforms the old model**.
- **Optimize Fraud Score Thresholds** based on business needs to balance fraud prevention and false positives.

- **Monitor False Positives** to ensure that genuine transactions are not unfairly flagged.
- **Implement Dynamic Fraud Detection Strategies**, adjusting score thresholds based on transaction patterns.

Final Notes on Model Comparison

The **new model outperforms the old one** and **is the recommended choice**, detecting more fraud with fewer false positives. Further refinements include using a **confusion matrix** to assess false detections, **optimizing the fraud score threshold** and **score overlap analysis** to fine-tune fraud cutoffs. Evaluating **time-based performance** can also enhance real-time fraud prevention, improving overall accuracy.

6. Fraud Prevention Rules & Performance Evaluation

Summary of Work Done

To improve fraud detection, we formulated **fraud prevention rules** based on key fraud indicators from EDA and Model Comparison. The rules were designed to **flag fraudulent transactions efficiently** while minimizing false positives. We implemented and evaluated multiple rules, selecting **Rule A** and **Rule D** as the most effective

- **Rule A**: High-precision rule capturing clear fraud patterns.
- **Rule D**: Balanced rule with better recall, detecting more fraud cases.
- Performance evaluation was conducted using **precision** and **recall** metrics to measure effectiveness.

Findings

Rule A (Strict, High-Precision Rule)

Rule A focuses on **common fraud behaviours**, flagging transactions based on:

- **Frequent orders** (5 or more per day).
- **High fraud scores & unusual payment methods**.
- **New accounts making large transactions**.
- **Peak fraud hours** (**4 PM - 9 PM**) and **high-risk days** (**Wed/Thu**).

Performance Evaluation:

- **Precision: 76.83%** (Most flagged transactions are actual fraud).
- **Recall: 15.75%** (Misses many fraud cases).

Rule A is highly **precise**, ensuring that **most flagged transactions are actual fraud**, but its recall is low, meaning it **misses some fraudsters**.

Rule D (Balanced Precision-Recall Rule)

Rule D refines fraud detection by adjusting time constraints and transaction volume thresholds.

- **Orders in a shorter time frame** (4 or more in 5 hours with high fraud score).
- **Stricter payment method checks** (unusual method + high score).
- **New accounts making very high-value purchases.**
- **Short time between consecutive orders** (fraudsters exploiting rapid transactions).

Performance Evaluation:

- **Precision: 21.60%** (Lower than Rule A, but still reasonable).
- **Recall: 25.25%** (Best recall among all rules, capturing more fraud cases).

Rule D **detects more fraud cases** than Rule A, though it allows for **more false positives**.

[Discarded rules, Rule B and Rule C included in Appendix]

Insights

- **Rule A is highly effective in minimizing false positives**, making it ideal for strict fraud prevention.
- **Rule D captures more fraud cases** but requires **further fine-tuning** to reduce unnecessary flags.
- **Balancing precision and recall** is crucial; a **combined approach** using both rules can **optimize fraud prevention**.

Recommendations

After evaluating all rules, **Rule A** and **Rule D** are the best choices for fraud prevention.

- Why Keep Rule A?
  - **Highest precision** (76.83%) ensures **minimal false positives**.
  - Captures **high-confidence fraud cases** effectively.
- Why Keep Rule D?
  - **Best recall** (25.25%) among all rules.
  - **Balances fraud detection with false positives**, preventing **missed fraud cases**.

Rules B and C were discarded as they **offered no significant improvement** over **Rule A** and **D**.

Final Notes on EDA

The fraud rules designed above follow a **structured approach**, with Rule A ensuring **high precision** and Rule D **improving recall** while keeping false positives manageable. Though not perfect, they provide a **balanced tradeoff** between fraud detection and minimizing false flags. Further refinements, such as **adaptive thresholds**, **machine learning models**, **fraud scoring systems**, and **business validation**, can enhance accuracy and reduce false positives.

7. Conclusion and Business Recommendations

Key Takeaways

- **New Model Outperforms** the old one (AUC-ROC: **0.886** vs. 0.829).
- Fraud Patterns: **New accounts, unusual payments, peak hours (4 PM - 9 PM) and high-risk days (Wednesdays & Thursdays).**
- Best Rules: **Rule A (high precision, minimal false positives)** and **Rule D (higher recall, better fraud capture)**.
- **Dynamic Thresholds**: Balance detection and business impact.
- **Hybrid Approach**: **Combine rules** and **model scoring** for **adaptive detection**.

Next Steps

- **Optimize fraud thresholds**.
- Enhance models with **geolocation and transaction velocity**.

- Continuously **refine Rule A** & **Rule D**.
- Shift to **dynamic risk-based scoring**.
- Strengthen **monitoring during peak hours and high-risk days**.

This ensures adaptive, efficient fraud detection and prevention with minimal disruption.

8. References

The following references were consulted for **learning purposes** while developing this mock project. The methodologies and techniques used in this project were inspired by **industry best practices**.

- **Pradeep, L.** (n.d.). *Building a Fraud Detection Model* https://pradeepl.com/blog/building-a-fraud-detection-model/

  This blog post outlines **end-to-end process of building a fraud detection model**, including **data preprocessing, feature selection, model training, and evaluation**. The insights helped in **refining fraud detection strategies.**

- **Gould, J.** (n.d.). *Fraud Detection in Python – Jupyter Notebook* https://github.com/gouldju1/Fraud-Detection-in-Python/blob/master/Fraud_Project.ipynb

  This Jupyter Notebook provides **end-to-end fraud detection implementation in Python**, covering **feature engineering, model training, and evaluation**. The approach was referenced to validate **feature selection strategies** and **fraud detection methodologies** used in this report.

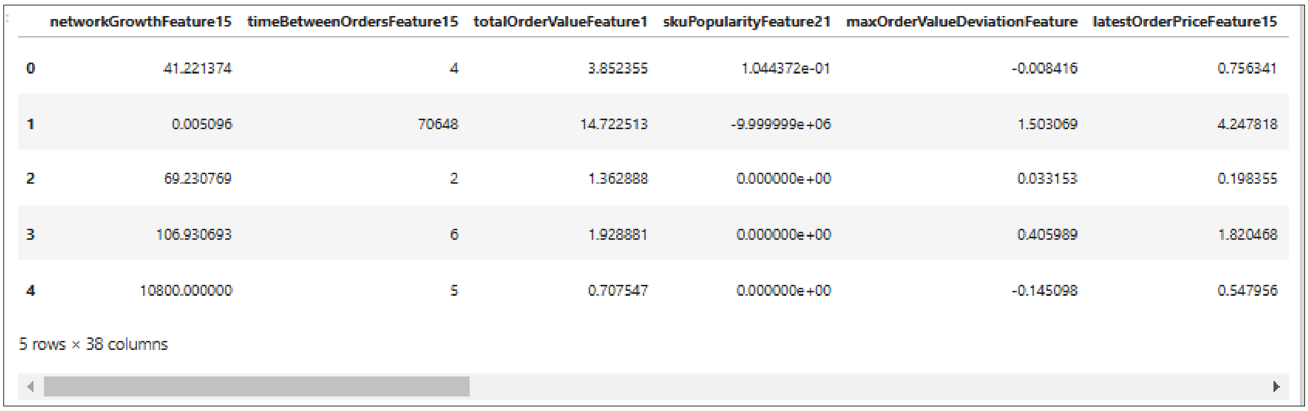
- **DataCamp.** (n.d.). *Python Tutorial: Introduction to Fraud Detection* https://www.youtube.com/watch?v=eu1NQW5Z5wk

  This tutorial provides an introduction to fraud detection using Python, covering essential concepts and practical implementation strategies.

- **General Online Search**. Various industry articles, blogs, and technical documentation were consulted through online searches to support fraud detection methodologies and model evaluation.

9. Appendix

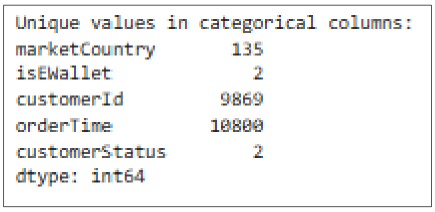
Figure 1 – Dataset sample



	networkGrowthFeature15	timeBetweenOrdersFeature15	totalOrderValueFeature1	skuPopularityFeature21	maxOrderValueDeviationFeature	latestOrderPriceFeature15
0	41.221374	4	3.852355	1.044372e-01	-0.008416	0.756341
1	0.005096	70648	14.722513	-9.999999e+06	1.503069	4.247818
2	69.230769	2	1.362888	0.000000e+00	0.033153	0.198355
3	106.930693	6	1.928881	0.000000e+00	0.405989	1.820468
4	10800.000000	5	0.707547	0.000000e+00	-0.145098	0.547956

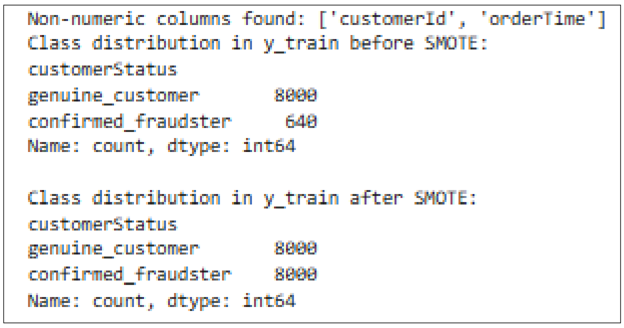
5 rows × 38 columns

Figure 2 – Categorical columns



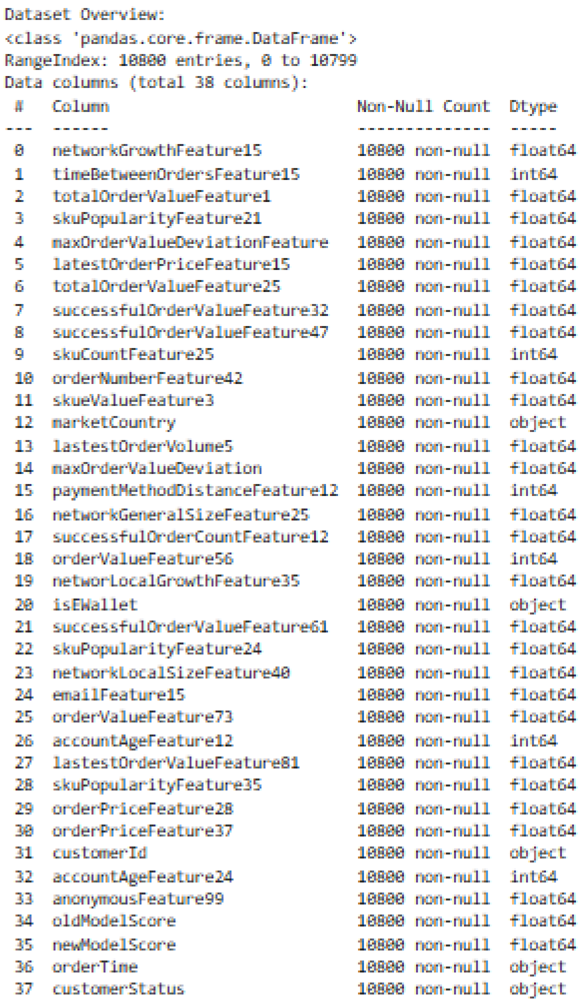
```
Unique values in categorical columns:
marketCountry        135
isEWallet              2
customerId          9869
orderTime          10800
customerStatus         2
dtype: int64
```

Figure 3 – Before & After SMOTE



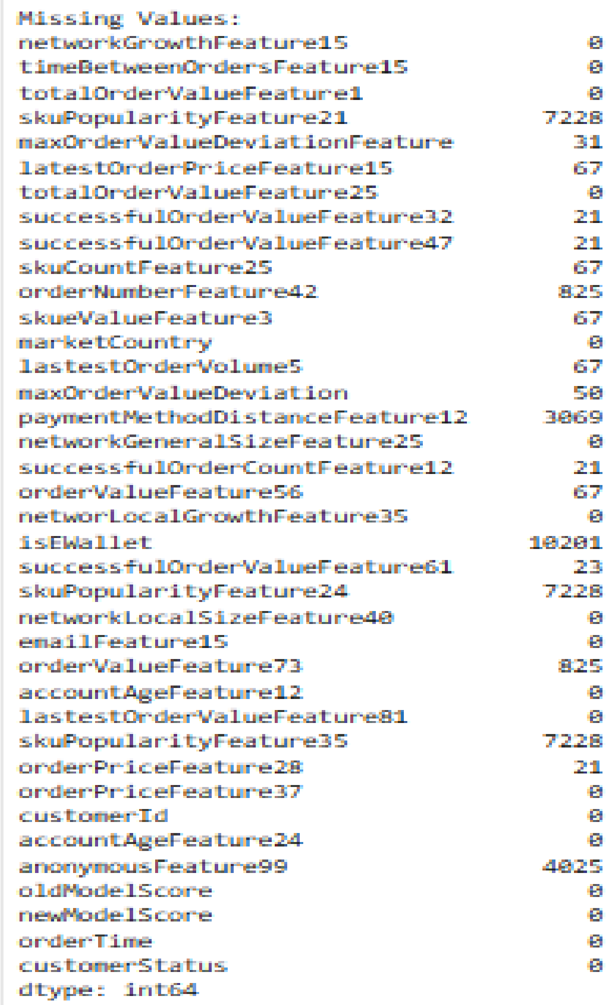
```
Non-numeric columns found: ['customerId', 'orderTime']
Class distribution in y_train before SMOTE:
customerStatus
genuine_customer       8000
confirmed_fraudster     640
Name: count, dtype: int64

Class distribution in y_train after SMOTE:
customerStatus
genuine_customer       8000
confirmed_fraudster    8000
Name: count, dtype: int64
```

Figure 4 – Dataset Overview



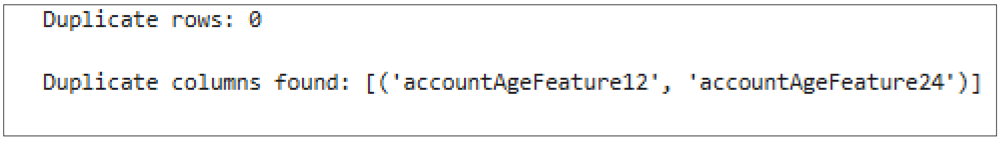
```
Dataset Overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10800 entries, 0 to 10799
Data columns (total 38 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   networkGrowthFeature15            10800 non-null  float64
 1   timeBetweenOrdersFeature15        10800 non-null  int64
 2   totalOrderValueFeature1           10800 non-null  float64
 3   skuPopularityFeature21            10800 non-null  float64
 4   maxOrderValueDeviationFeature     10800 non-null  float64
 5   latestOrderPriceFeature15         10800 non-null  float64
 6   totalOrderValueFeature25          10800 non-null  float64
 7   successfulOrderValueFeature32     10800 non-null  float64
 8   successfulOrderValueFeature47     10800 non-null  float64
 9   skuCountFeature25                 10800 non-null  int64
 10  orderNumberFeature42              10800 non-null  float64
 11  skueValueFeature3                 10800 non-null  float64
 12  marketCountry                     10800 non-null  object
 13  lastestOrderVolume5               10800 non-null  float64
 14  maxOrderValueDeviation            10800 non-null  float64
 15  paymentMethodDistanceFeature12    10800 non-null  int64
 16  networkGeneralSizeFeature25       10800 non-null  float64
 17  successfulOrderCountFeature12     10800 non-null  float64
 18  orderValueFeature56               10800 non-null  int64
 19  networLocalGrowthFeature35        10800 non-null  float64
 20  isEWallet                         10800 non-null  object
 21  successfulOrderValueFeature61     10800 non-null  float64
 22  skuPopularityFeature24            10800 non-null  float64
 23  networkLocalSizeFeature40         10800 non-null  float64
 24  emailFeature15                    10800 non-null  float64
 25  orderValueFeature73               10800 non-null  float64
 26  accountAgeFeature12               10800 non-null  int64
 27  lastestOrderValueFeature81        10800 non-null  float64
 28  skuPopularityFeature35            10800 non-null  float64
 29  orderPriceFeature28               10800 non-null  float64
 30  orderPriceFeature37               10800 non-null  float64
 31  customerId                        10800 non-null  object
 32  accountAgeFeature24               10800 non-null  int64
 33  anonymousFeature99                10800 non-null  float64
 34  oldModelScore                     10800 non-null  float64
 35  newModelScore                     10800 non-null  float64
 36  orderTime                         10800 non-null  object
 37  customerStatus                    10800 non-null  object
```

Figure 5 – Missing Values



```
Missing Values:
networkGrowthFeature15                0
timeBetweenOrdersFeature15            0
totalOrderValueFeature1               0
skuPopularityFeature21             7228
maxOrderValueDeviationFeature        31
latestOrderPriceFeature15            67
totalOrderValueFeature25              0
successfulOrderValueFeature32        21
successfulOrderValueFeature47        21
skuCountFeature25                    67
orderNumberFeature42                825
skueValueFeature3                    67
marketCountry                         0
lastestOrderVolume5                  67
maxOrderValueDeviation               50
paymentMethodDistanceFeature12     3069
networkGeneralSizeFeature25           0
successfulOrderCountFeature12        21
orderValueFeature56                  67
networLocalGrowthFeature35            0
isEWallet                         10201
successfulOrderValueFeature61        23
skuPopularityFeature24             7228
networkLocalSizeFeature40             0
emailFeature15                        0
orderValueFeature73                 825
accountAgeFeature12                   0
lastestOrderValueFeature81            0
skuPopularityFeature35             7228
orderPriceFeature28                  21
orderPriceFeature37                   0
customerId                            0
accountAgeFeature24                   0
anonymousFeature99                 4025
oldModelScore                         0
newModelScore                         0
orderTime                             0
customerStatus                        0
dtype: int64
```

Figure 6 – Duplicate rows & columns



```
Duplicate rows: 0

Duplicate columns found: [('accountAgeFeature12', 'accountAgeFeature24')]
```

Figure 7 – Models Comparison for entire dataset

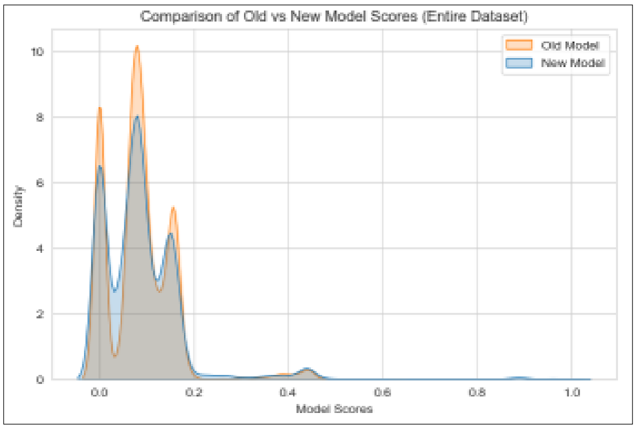
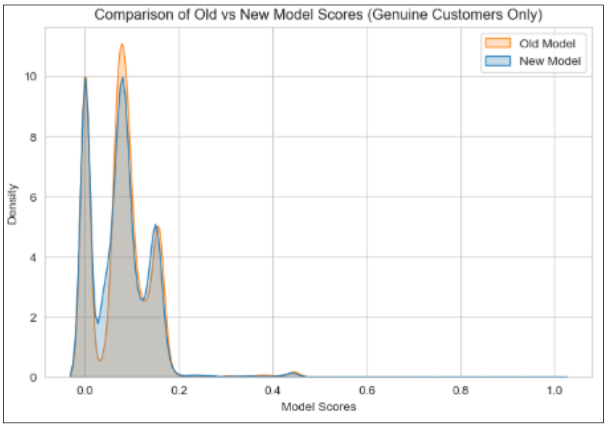


Figure 8 – Models Comparison for Genuine Customers



Rule B *(discarded due to low recall (4.62%), meaning it missed too many fraudsters compared to Rule A)*

Rule B is a stricter version of Rule A as seen below:

- Frequent Orders on Same Day: Customers placing 5 or more orders per day are flagged.
- High Fraud Score & Multiple Payment Methods: If a transaction has fraud score > 0.6 and uses an unusual payment method, it is flagged.
- New Accounts Making Large Transactions: Accounts less than 30 days old that make high-value transactions (>75th percentile) are flagged.
- Stricter Version: Additional condition – only flag customers with 5 or more orders if fraud score > 0.5.

Performance Evaluation:

- Precision: 52.11% (fairly good, but not as strong as Rule A).
- Recall: 4.62% (worse than Rule A, missing even more fraud cases).

Rule C *(discarded* as it did not offer significant improvement over Rule A**,** making it unnecessary)

Rule C introduces tighter fraud detection criteria while keeping false positives low:

- Very High Fraud Score: Any transaction with fraud score > 0.6 is flagged.
- Time-Based Fraud Detection (Stricter Version): If a customer places 5 or more orders within 6 hours and has a fraud score > 0.6, they are flagged.
- Unusual Payment Method Activity: If a transaction has a fraud score > 0.65 and uses an uncommon payment method, it is flagged.
- High-Value Transactions from New Accounts: If an account is less than 20 days old and places a high-value order (>85th percentile), it is flagged.
- Excluding Legitimate Customers: Customers with a large account network size are not flagged unless they have a fraud score > 0.6.

Performance Evaluation:

- Precision: 75.00% Excellent precision, similar to Rule A).
- Recall: 6.00% (Recall is too low, even lower than Rule A).