# Final Repo

## 1. Overview and Objective

Bati Bank, in partnership with a leading eCommerce platform, aims to launch a Buy-Now-Pay-Later (BNPL) service. The core objective of this project is to develop a robust Credit Scoring Model that assesses the creditworthiness of customers based on their transaction history. Since traditional credit history is unavailable, the project innovates by engineering a proxy target variable derived from behavioral data—specifically Recency, Frequency, Monetary, and Stability (RFMS) metrics. This approach segments users into "High Risk" (Bad) and "Low Risk" (Good) categories, creating a labeled dataset for supervised learning. The project successfully engineered a comprehensive feature set, validated predictive power using Weight of Evidence (WoE) and Information Value (IV) analysis, and trained machine learning models (including Random Forest and Gradient Boosting) to predict default probability. The final deliverable is a deployable model that assigns a risk probability score to new applicants, enabling data-driven decisions on loan approval, optimal loan amounts, and repayment durations, thereby minimizing default risk while maximizing service adoption.

## 2. Data Structure and Quality

The dataset consists of 95,662 transactions and 16 columns.

- Completeness: No missing values were detected in the preliminary scan.
- Data Types: Key timestamps (TransactionStartTime) required conversion to datetime. Categorical flags like FraudResult and PricingStrategy were identified.
- Granularity: The data contains both AccountId and CustomerId, revealing that single customers often operate multiple accounts.

## 3. Statistical Summary

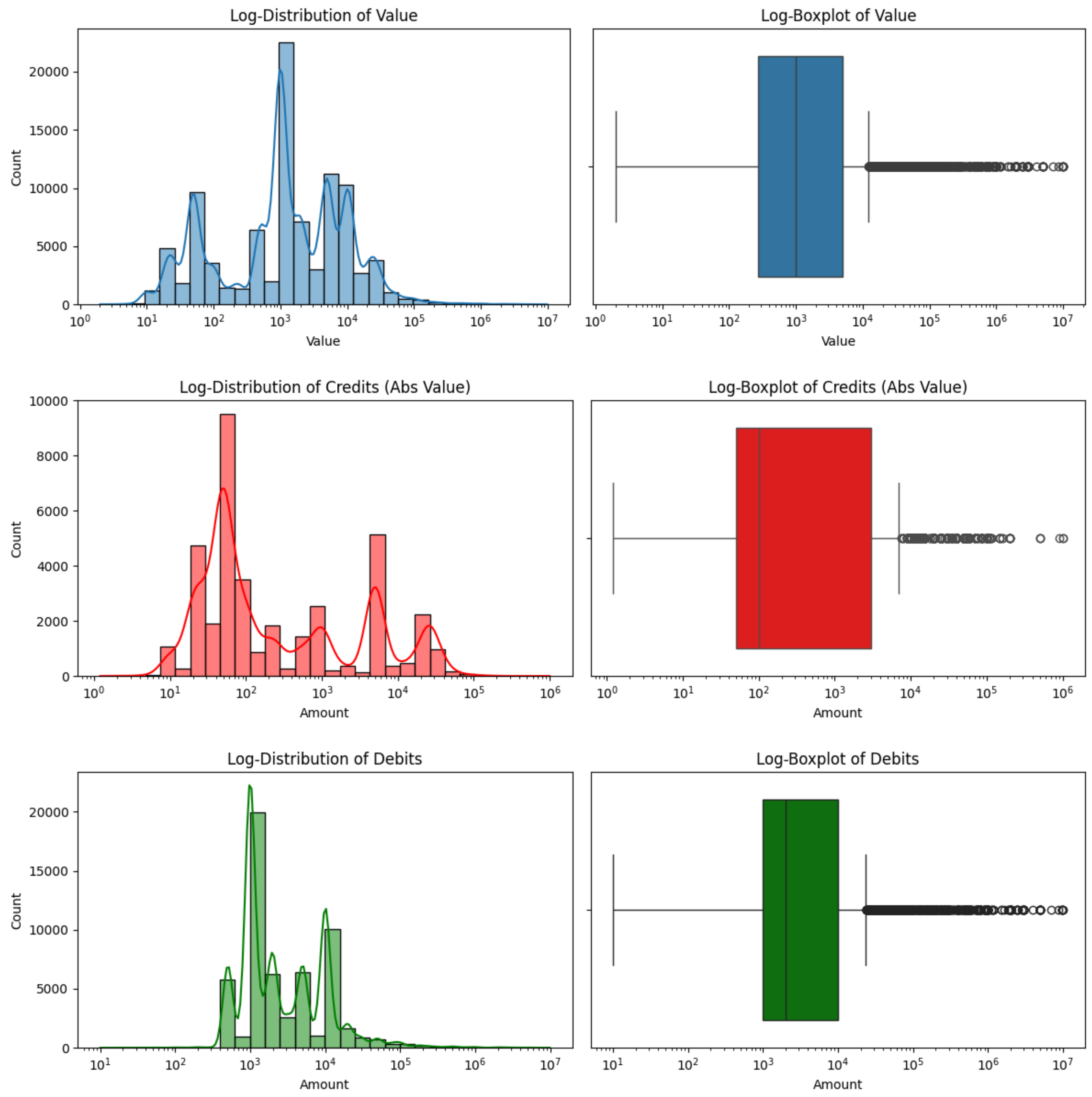
The analysis of central tendency and dispersion revealed a highly skewed dataset driven by micro-transactions.

- Value (Absolute): The median transaction value is low (1,000 UGX), while the mean is significantly higher (~9,900 UGX), indicating a heavy right tail.
- Amount (Signed): The dataset contains both debits (positive) and credits (negative). Both directions show extreme outliers, necessitating separate handling for "inflow" vs. "outflow" features.

## 4. Distribution Analysis

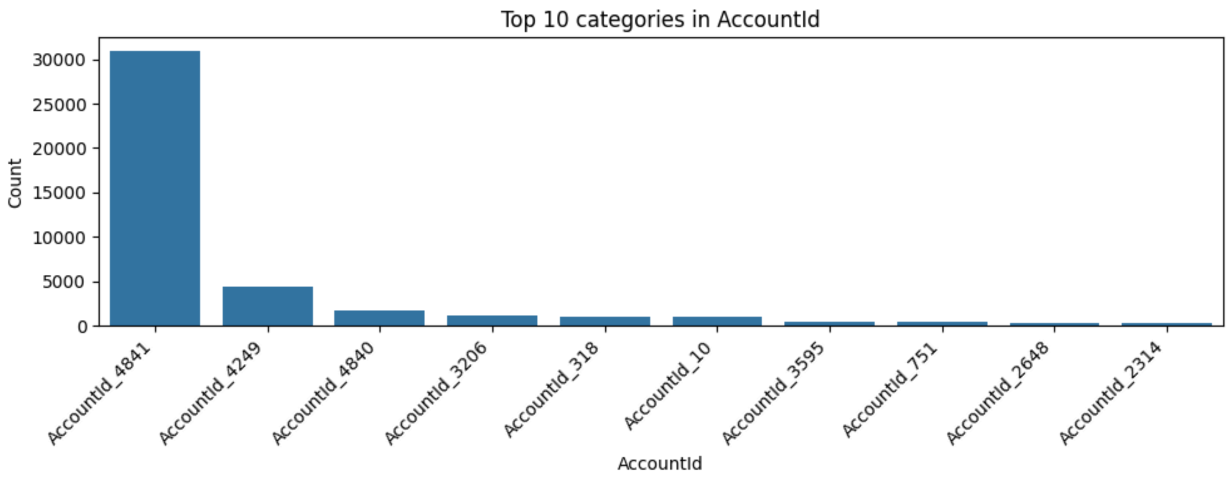
**Numerical Features**

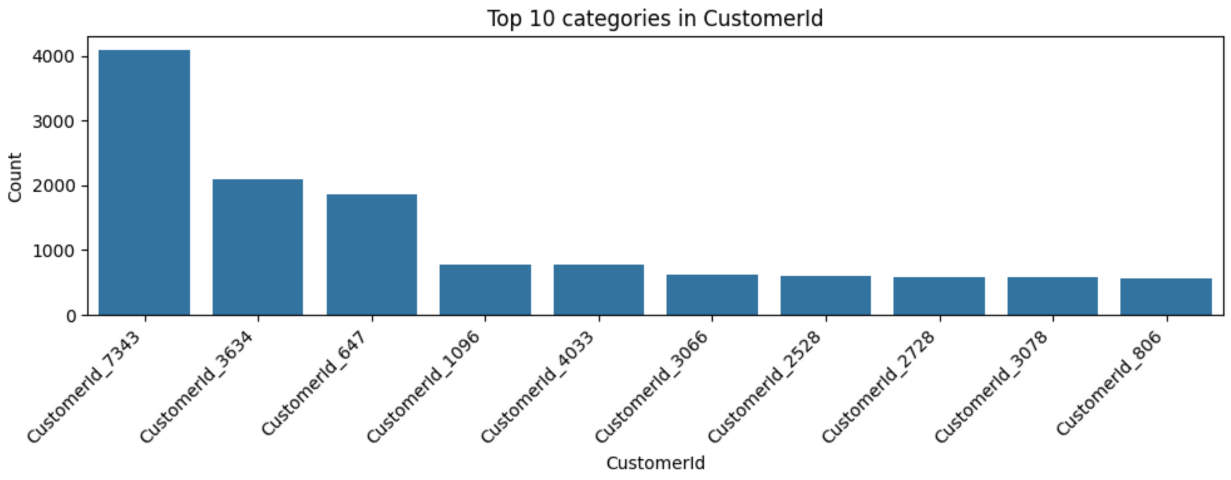
The distribution of transaction values is multi-modal, suggesting distinct user tiers:
1. Tier 1 (50-500 UGX): Casual users (likely airtime/fees).
2. Tier 2 (1,000-5,000 UGX): Standard users (P2P/bills).
3. Tier 3 (10,000+ UGX): Power users or business accounts.

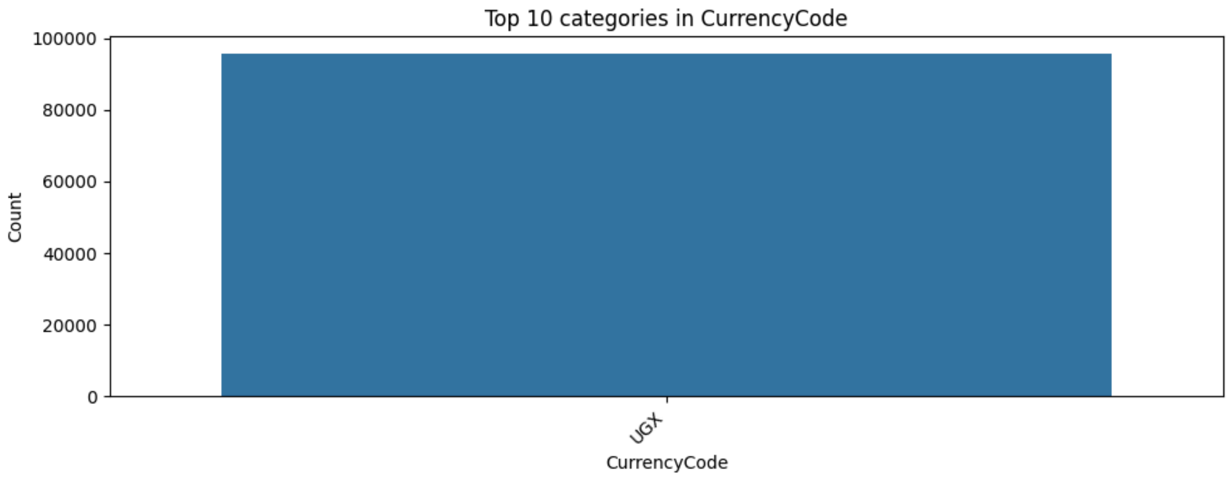


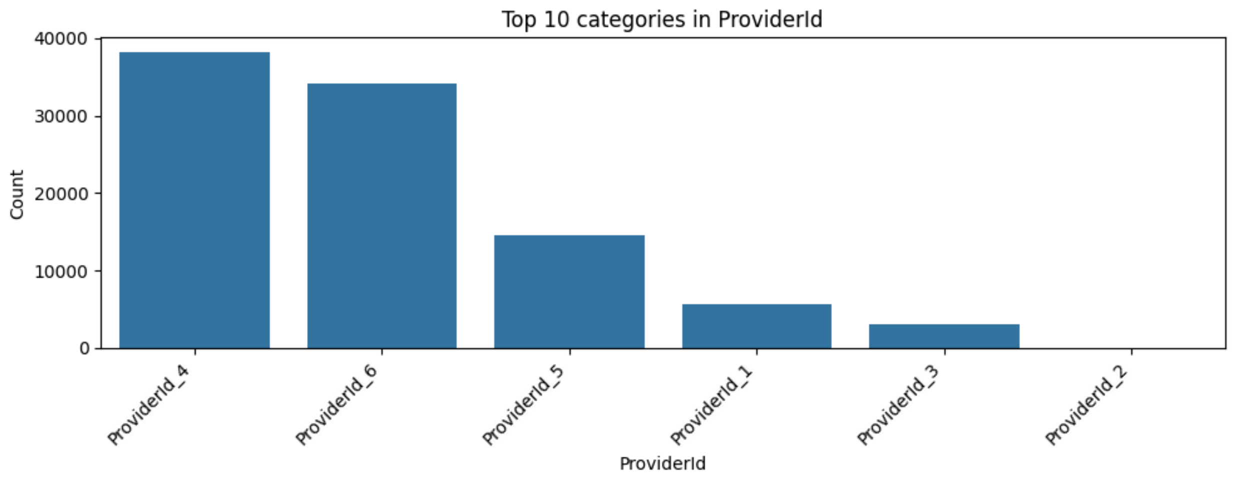
## Categorical Features

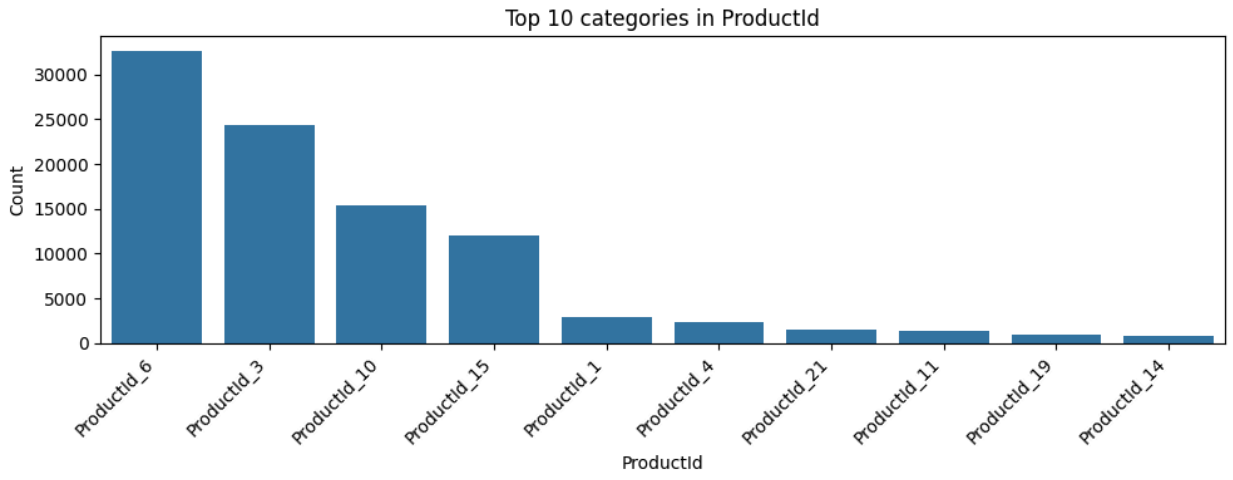
- Product Category: Dominated by financial_services and airtime (~95% of volume).
- Channel: ChannelId_1 (Web) is the primary channel, with mobile channels underrepresented.
- Provider: ProviderId_4 holds the largest market share.

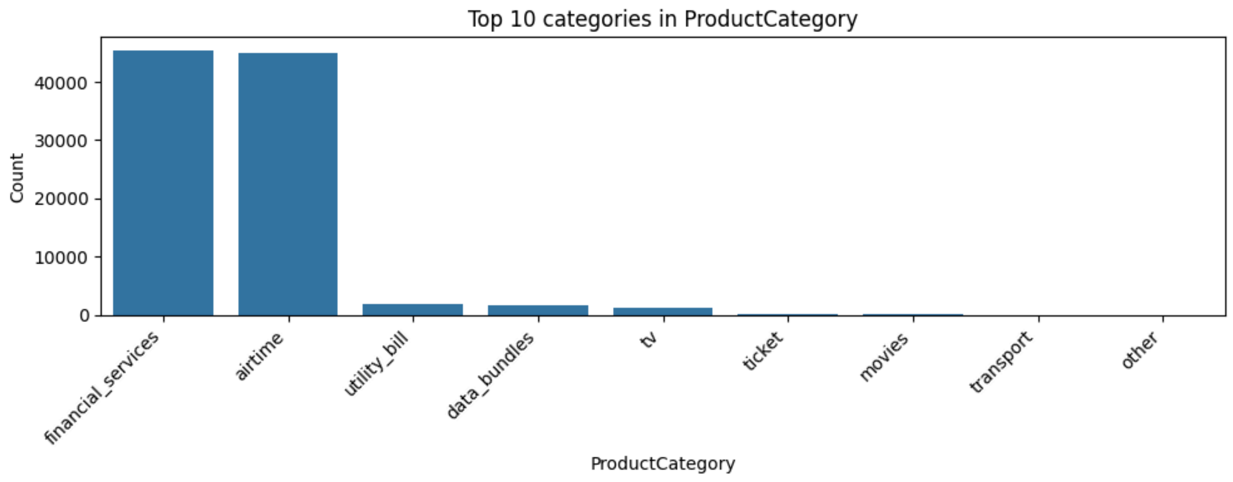

Top 10 categories in AccountId
Count
30000
25000
20000
15000
10000
5000
0
AccountId_4841
AccountId_4249
AccountId_4840
AccountId_3206
AccountId_318
AccountId_10
AccountId_3595
AccountId_751
AccountId_2648
AccountId_2314
AccountId

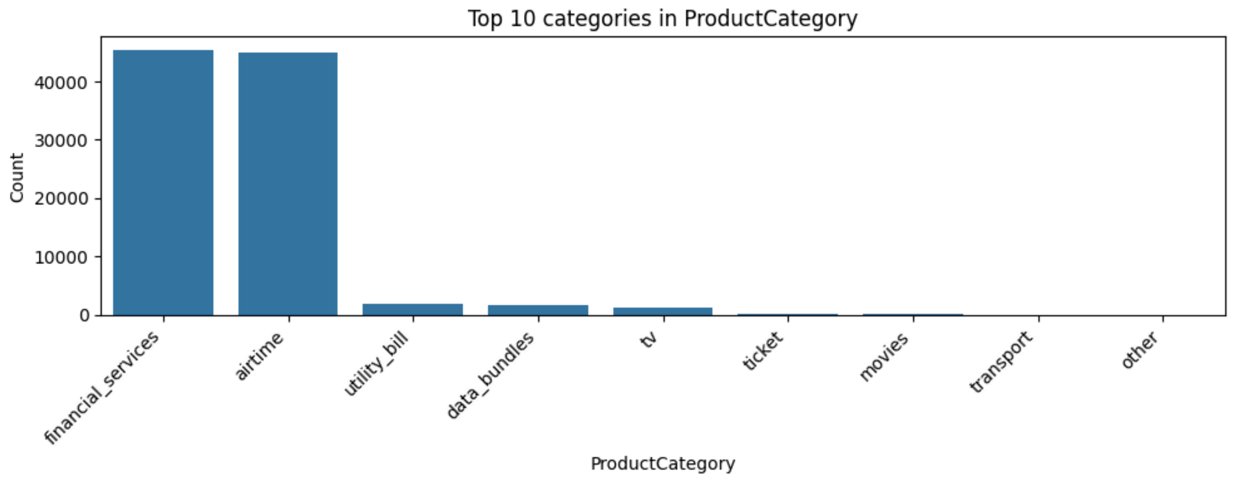

Top 10 categories in CustomerId
Count
4000
3000
2000
1000
0
CustomerId_7343
CustomerId_3634
CustomerId_647
CustomerId_1096
CustomerId_4033
CustomerId_3066
CustomerId_2528
CustomerId_2728
CustomerId_3078
CustomerId_806
CustomerId

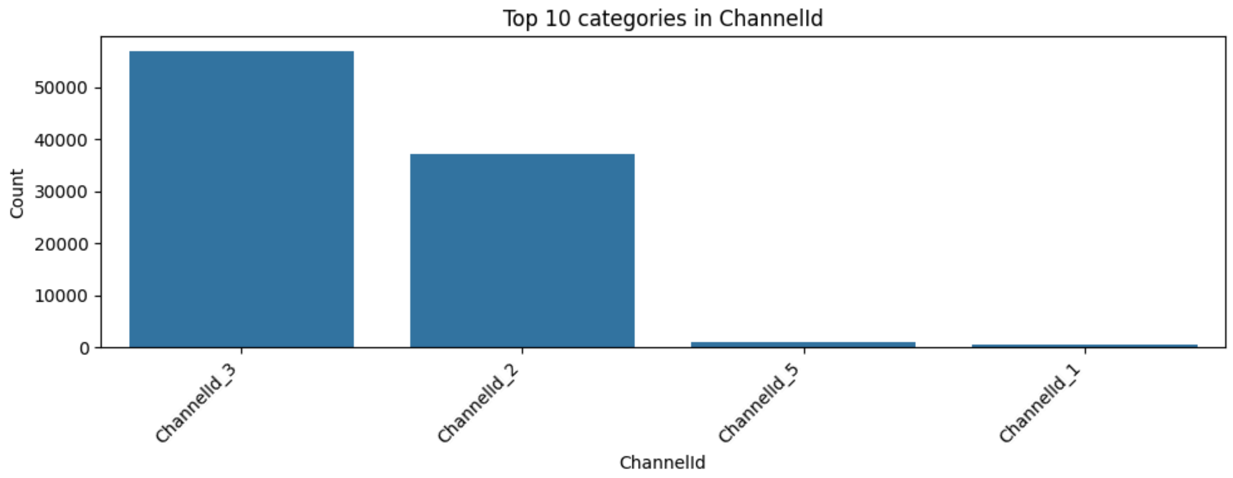

Top 10 categories in CurrencyCode
Count
100000
80000
60000
40000
20000
0
UGX
CurrencyCode

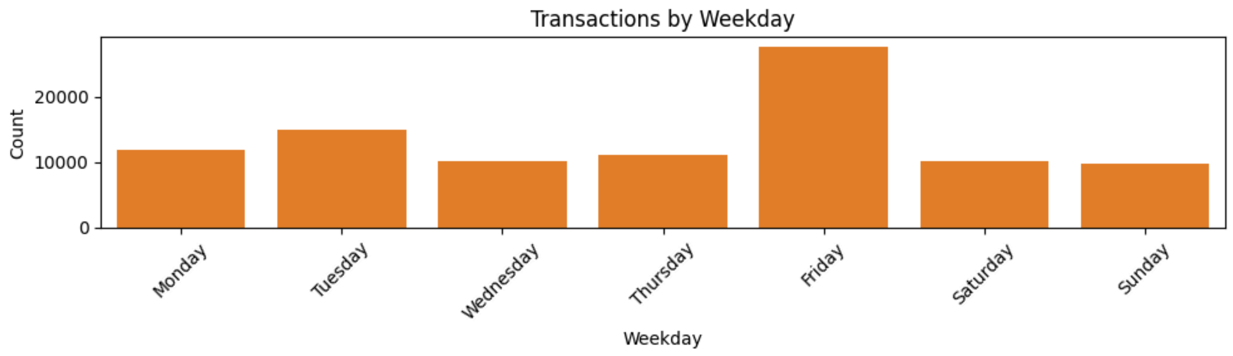

Top 10 categories in ProviderId
Count
40000
30000
20000
10000
0
ProviderId_4
ProviderId_6
ProviderId_5
ProviderId_1
ProviderId_3
ProviderId_2
ProviderId

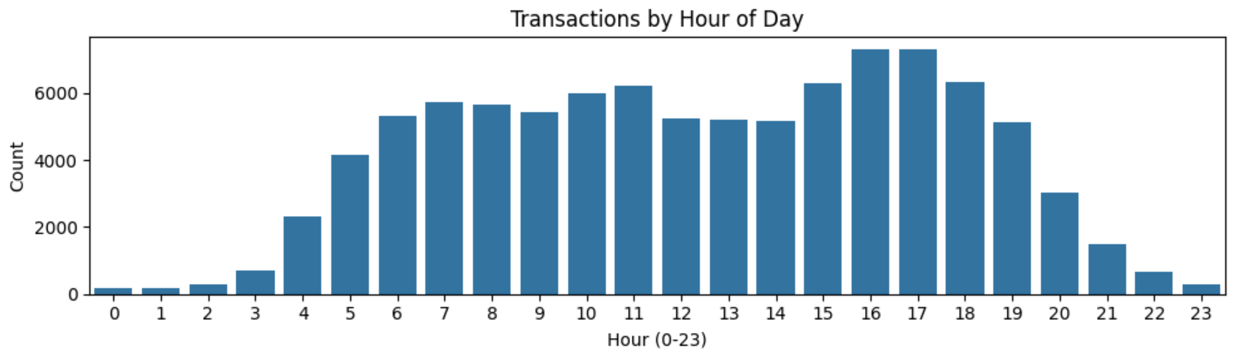

Top 10 categories in ProductId
Count
30000
25000
20000
15000
10000
5000
0
ProductId_6
ProductId_3
ProductId_10
ProductId_15
ProductId_1
ProductId_4
ProductId_21
ProductId_11
ProductId_19
ProductId_14
ProductId


Top 10 categories in ProductCategory
Count
0
10000
20000
30000
40000
financial_services
airtime
utility_bill
data_bundles
tv
ticket
movies
transport
other
ProductCategory


Top 10 categories in ProductCategory
Count
0
10000
20000
30000
40000
financial_services
airtime
utility_bill
data_bundles
tv
ticket
movies
transport
other
ProductCategory


Top 10 categories in ChannelId
Count
0
10000
20000
30000
40000
50000
ChannelId_3
ChannelId_2
ChannelId_5
ChannelId_1
ChannelId



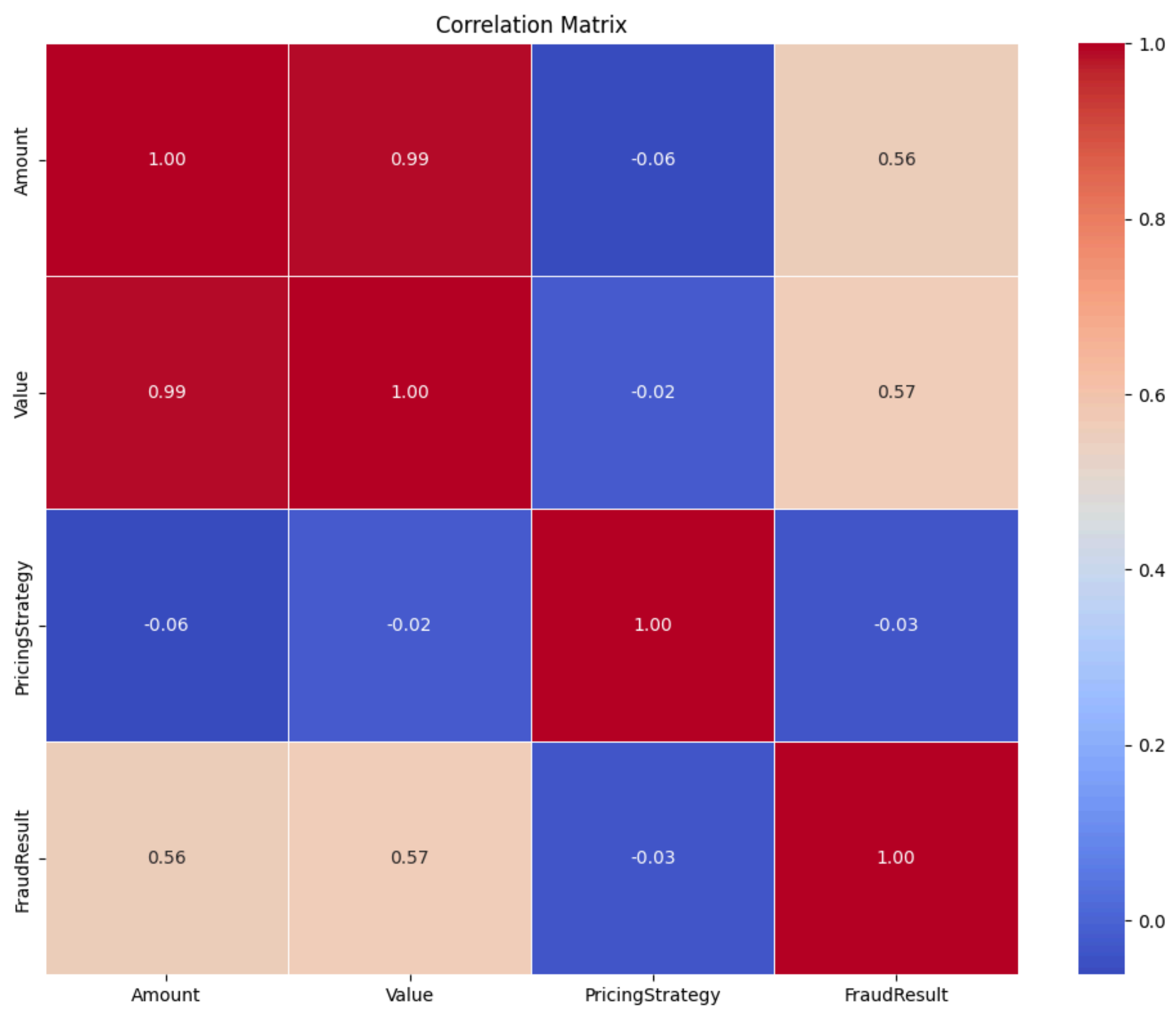


## 5. Correlation and Outliers

### Correlation

Correlation analysis was performed to understand relationships between numerical features. (Note: High correlation between Amount and Value is expected).

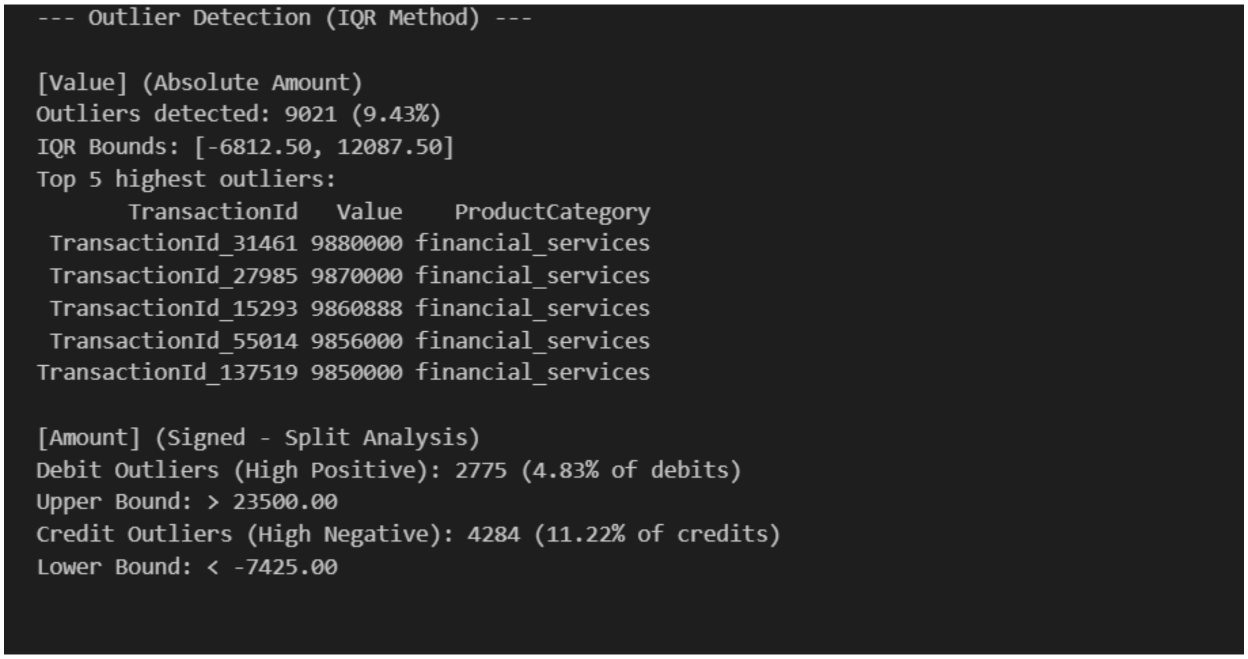
The matrix highlights that Amount and Value move almost in lockstep (Value is the absolute Amount), so either one is redundant for modeling unless you derive asymmetric features (e.g., separate debit/credit). Other numerical features show mostly weak correlations, which is good for multicollinearity but means you need engineered interactions (e.g., ProductCategory * Amount). If you spot any unexpected strong correlations (e.g., CountryCode vs. ChannelId), that may signal proxies or data leakage that need review before training.



## 5.Outlier Detection

Significant outliers were detected using the IQR method. These "whales" represent a mix of high-value VIP customers and potential fraud risks.

- Action: These values must be capped or log-transformed for the Logistic Regression (WoE) model to prevent coefficient distortion.



```
--- Outlier Detection (IQR Method) ---

[Value] (Absolute Amount)
Outliers detected: 9021 (9.43%)
IQR Bounds: [-6812.50, 12087.50]
Top 5 highest outliers:
       TransactionId   Value    ProductCategory
 TransactionId_31461 9880000 financial_services
 TransactionId_27985 9870000 financial_services
 TransactionId_15293 9860888 financial_services
 TransactionId_55014 9856000 financial_services
TransactionId_137519 9850000 financial_services

[Amount] (Signed - Split Analysis)
Debit Outliers (High Positive): 2775 (4.83% of debits)
Upper Bound: > 23500.00
Credit Outliers (High Negative): 4284 (11.22% of credits)
Lower Bound: < -7425.00
```

## 6. EDA Key Insights and Business Implications

Based on the EDA, the following strategic insights align with the end-to-end modeling goals:

**A. Viability of RFM for Proxy Target Definition**
The heavy skew and clear separation between casual and power users confirm the strategy to use RFM (Recency, Frequency, Monetary) clustering. We can confidently label the "least engaged" cluster as high-risk (ishighrisk=1) for supervised training.

**B. Segmentation is Critical**
A "one-size-fits-all" model may underperform due to the dominance of specific product categories. Feature engineering should include interaction terms (e.g., Product x Amount), and separate scorecards may be considered for high-value vs. low-value segments.

**C. Outlier Management for "PD-Like" Scoring**
Insight: Significant outliers exist in `Amount` and `Value`.
Business Impact: For the Logistic Regression (WoE) model mentioned in the README (favored for interpretability/Basel II), these outliers must be capped or log-transformed. Raw values would distort the coefficients. Additionally, distinguishing between Credits(inflows/refunds) and Debits(outflows) is vital—consistent inflows should improve a customer's credit score (lower PD).

**D. Entity Resolution for Credit Limits**

Discrepancies between top Customer and Account counts confirm that users operate multiple accounts. Risk probability (PD) and Exposure (EAD) must be calculated at the CustomerId level to prevent risky users from bypassing limits by opening new accounts.

**E. Channel Bias and Deployment Risk**
The dominance of web-based transactions poses a deployment risk if the future product targets mobile users. Post-deployment monitoring for Data Drift in ChannelId is essential.

## 7. Feature Engineering

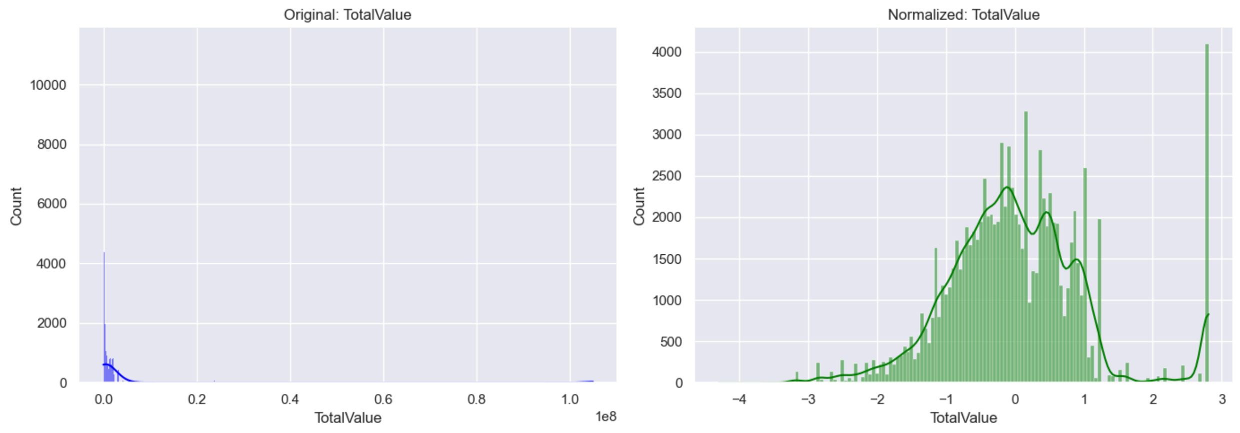
The feature engineering phase focused on transforming raw transactional data into a robust set of customer-centric features suitable for credit risk modeling. The process began with **Temporal Feature Extraction**, where the `TransactionStartTime` was decomposed into granular components such as Hour, Day, Month, Year, and Weekday. Additionally, binary flags for `IsWeekend` and `IsBusinessHour` were created to capture behavioral patterns related to transaction timing. Following this, the data was aggregated at the customer level to create a comprehensive profile for each user. This aggregation generated key metrics including `TotalValue`, `TransactionCount`, `AvgValue`, `MaxDebit`, `MaxCredit`, and `NetFlow`, effectively summarizing a customer's financial behavior and liquidity.

To ensure data integrity, a rigorous **Missing Value Imputation** strategy was applied. Columns such as `MaxCredit`, `MaxDebit`, `StdValue`, and `CV_Value` contained missing values for customers with limited transaction history (e.g., no debits or only single transactions). These were logically imputed with 0, reflecting the absence of credit/debit activity or variance. Unnecessary columns were subsequently dropped, resulting in a clean feature set of 26 numerical and binary variables for 95,662 unique customers.

Finally, the dataset underwent **Feature Transformation** to address skewness and scale differences, which are critical for model performance. A hybrid strategy was employed: "Money and Count" variables (e.g., `TotalValue`, `TransactionCount`) which followed a Power Law distribution were transformed using a Log-Standardization approach (`log1p` followed by `StandardScaler`

`). This successfully compressed extreme outliers ("whales") and normalized the distributions. Flow variables like` NetFlow ` were Standardized directly to center the mean at 0. Visual comparisons and statistical summaries confirmed that these transformations significantly reduced skewness, creating a normalized dataset ready for machine learning algorithms.`

Plot showing the effectiveness of standardization before and after:



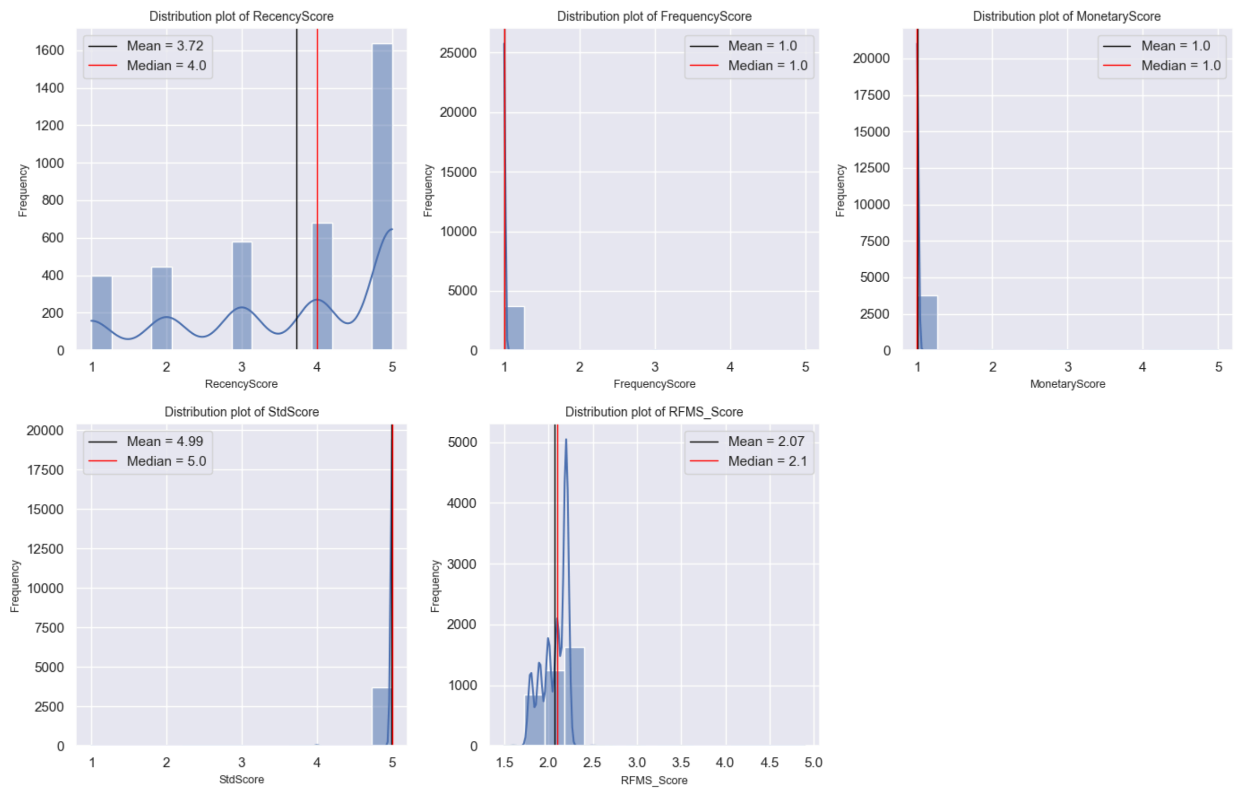
## 8. Target Variable Engineering and Risk Analysis

To establish a reliable target variable for credit risk modeling, a proxy target was engineered using RFMS (Recency, Frequency, Monetary, Standard Deviation) analysis. Since the dataset lacked explicit "Default" labels, this unsupervised approach allowed for the segmentation of users based on their transaction behavior and stability.

**RFMS Segmentation and Scoring**
A custom CreditScoreEngine was developed to calculate RFMS metrics for each customer:

Recency: Days since the last transaction.
Frequency: Total number of transactions.
Monetary: Total value of transactions.
Stability (Std): Standard deviation of transaction amounts (imputed with 0 for single-transaction users).
These metrics were scored on a scale of 1 to 5 using quantile binning. A weighted composite RFMS_Score was then calculated, prioritizing Monetary value (50%) and Stability (20%) to reflect creditworthiness.

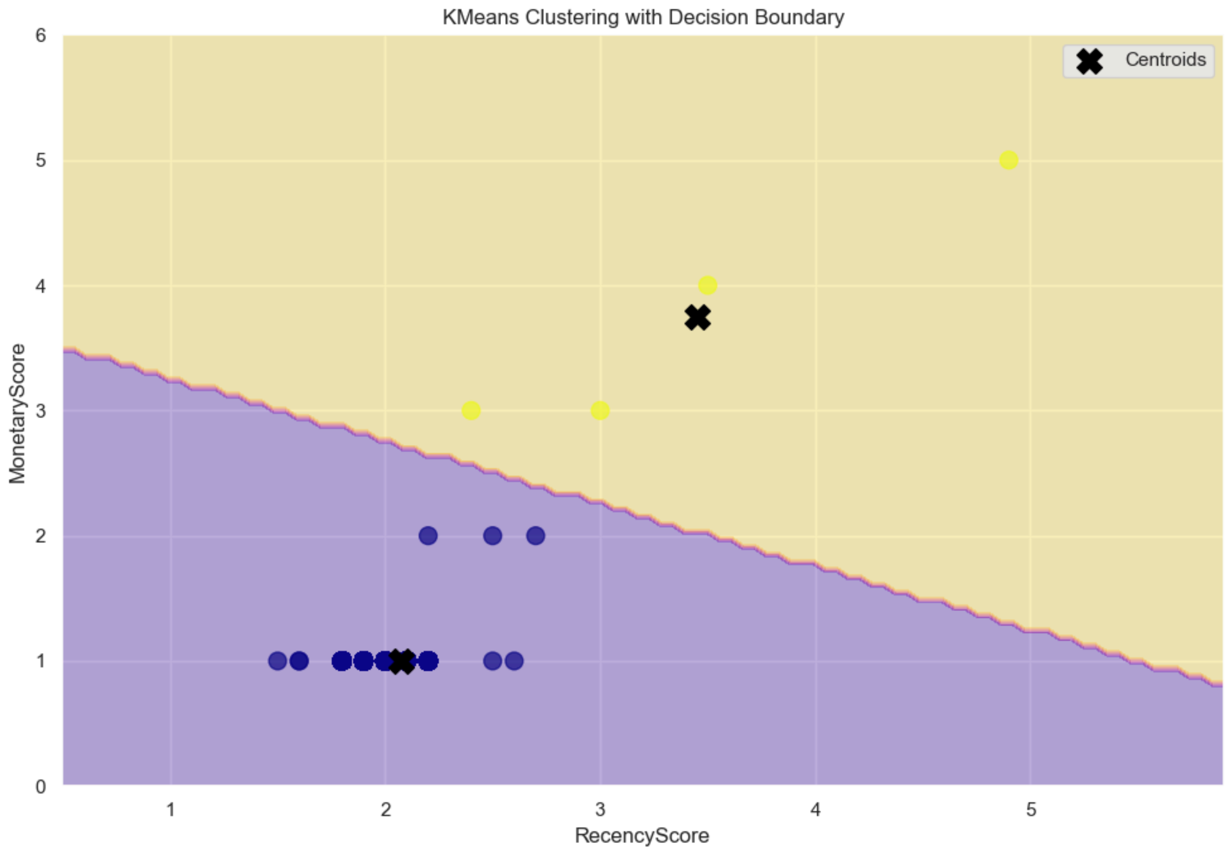
**Figure 1: Distribution of calculated RFMS scores across the customer base.**

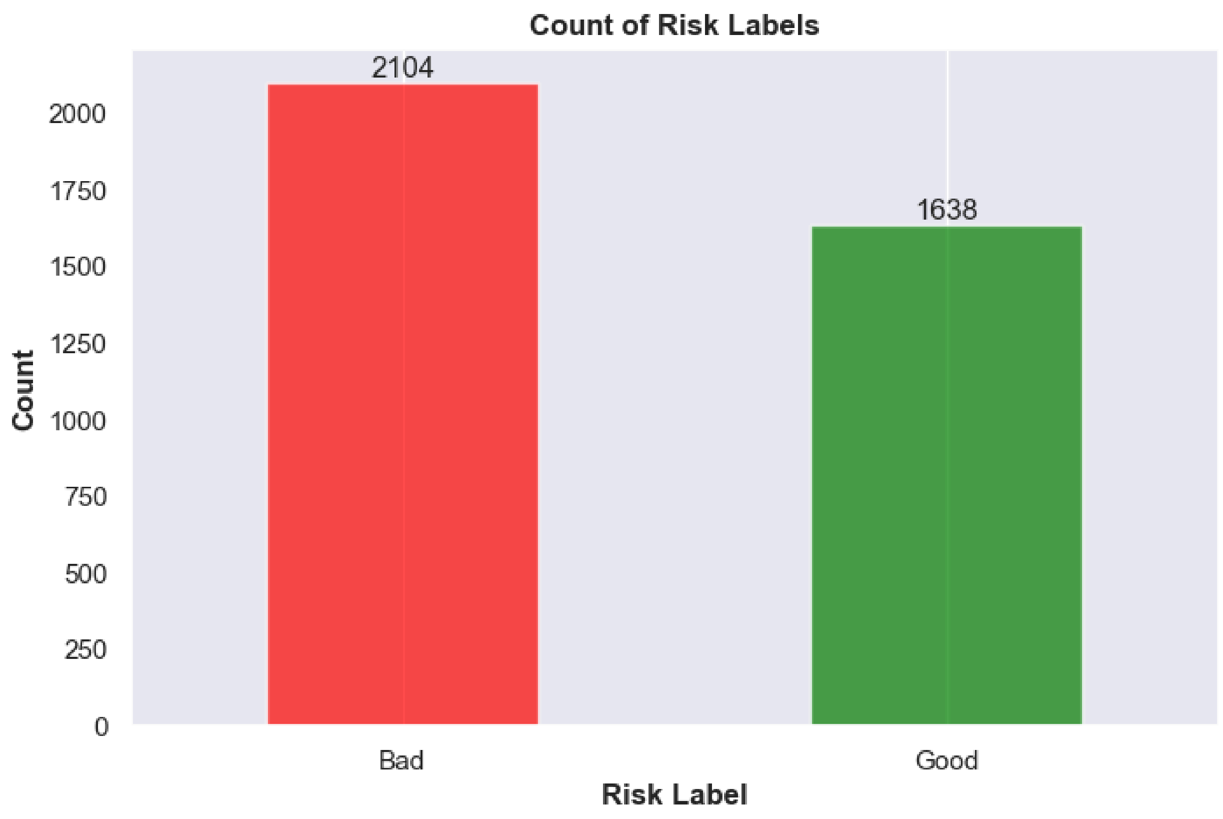


**Risk Labeling (The Proxy Target)**
To classify users as "Good" (Low Risk) or "Bad" (High Risk), K-Means clustering was initially tested but found to be too harsh, classifying 99% of users as "Bad." A more balanced approach was adopted using the 55th percentile of the RFMS_Score as the decision boundary.

Good (1): Users with an RFMS score above the boundary (High value, consistent behavior).
Bad (0): Users with an RFMS score below the boundary.

**Figure 2: Visualization of the decision boundary separating High Risk and Low Risk customers.**


KMeans Clustering with Decision Boundary
Centroids
MonetaryScore
RecencyScore
0
1
2
3
4
5
6
1
2
3
4
5



**Weight of Evidence (WoE) and Information Value (IV) Analysis**
To validate the predictive power of features against this new target, a Weight of Evidence (WoE) analysis was conducted. This step transformed independent variables into a measure of "risk strength."

**Key Insight:** The RFMS_Score and RecencyScore showed the highest Information Value (IV), confirming their strong relationship with the target.
Categorical Insight: ProductCategory proved to be a significant driver of risk differentiation, whereas ChannelId showed weaker predictive power.

**Figure 3: Inverse relationship between WoE and Bad Probability for RFMS Score bins.**

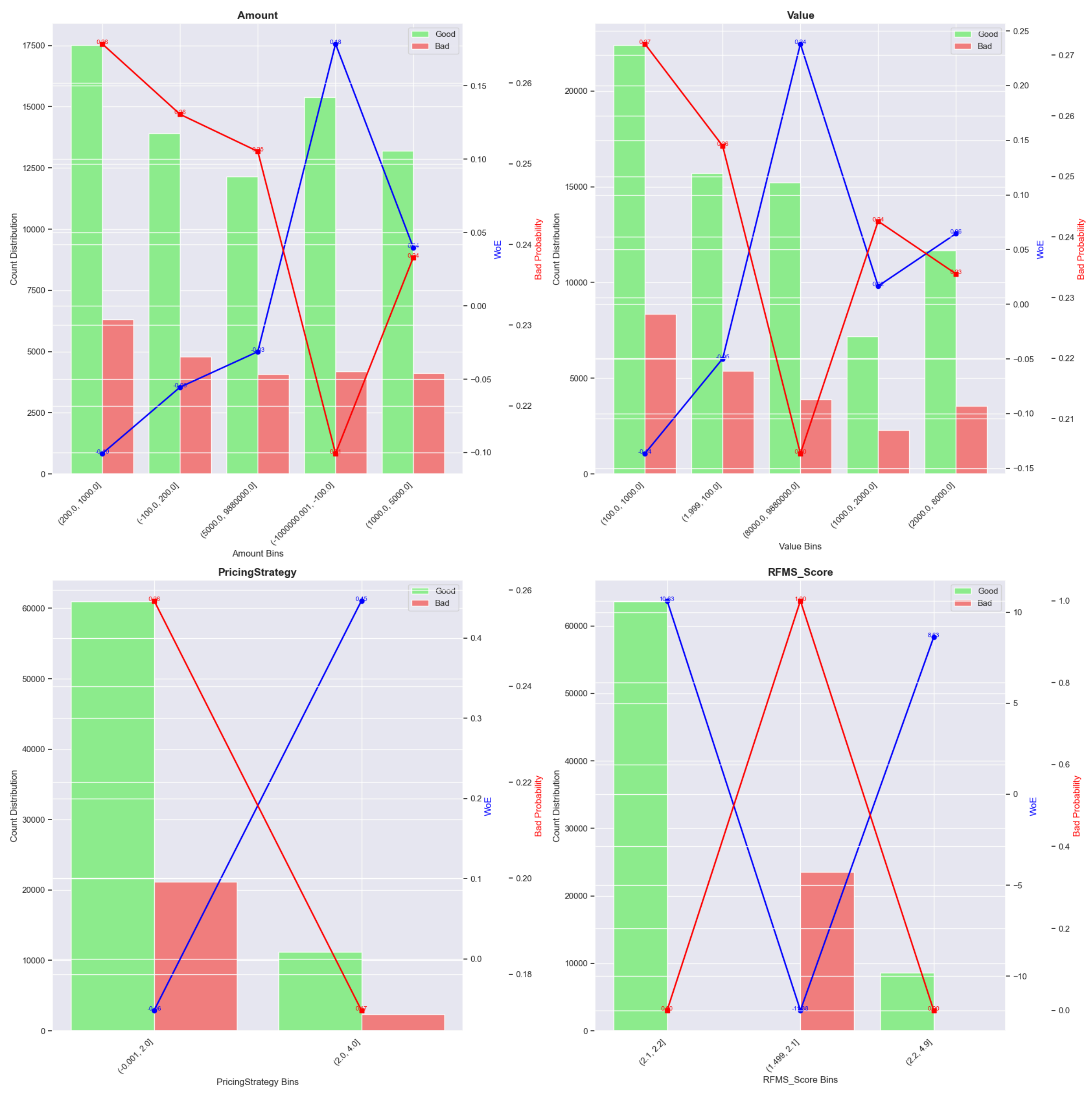
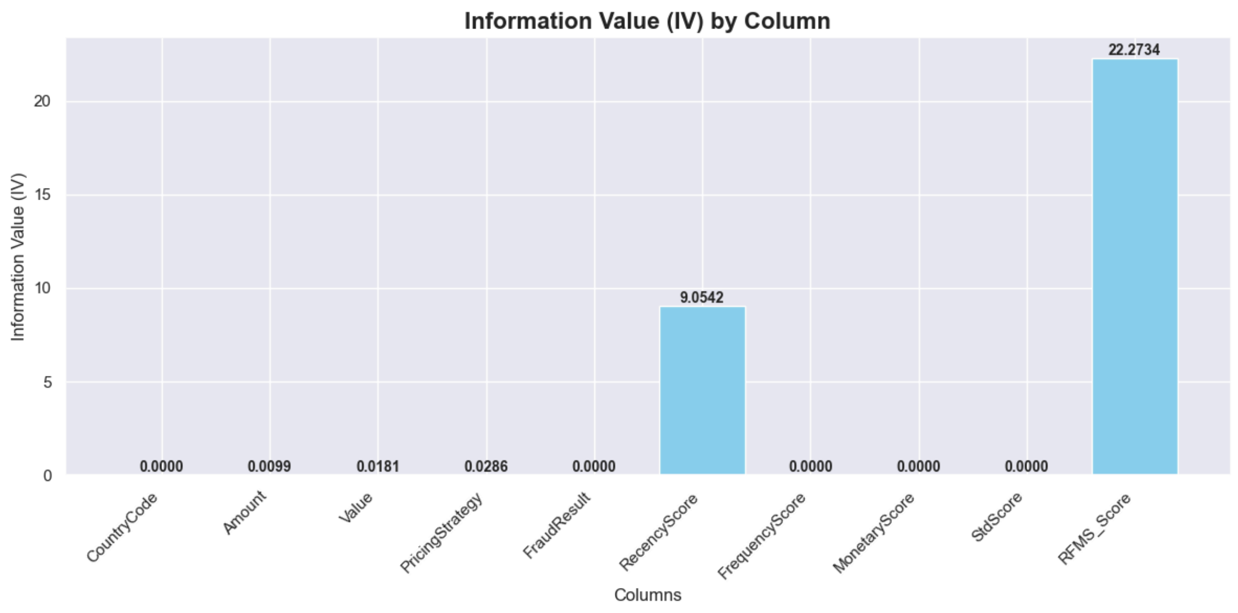


**Figure 4: Information Value (IV) ranking of features.**



**9. Model Training and Evaluation**
With the features engineered and the target variable defined (RiskLabel), the focus shifted to training machine learning models to predict credit risk.

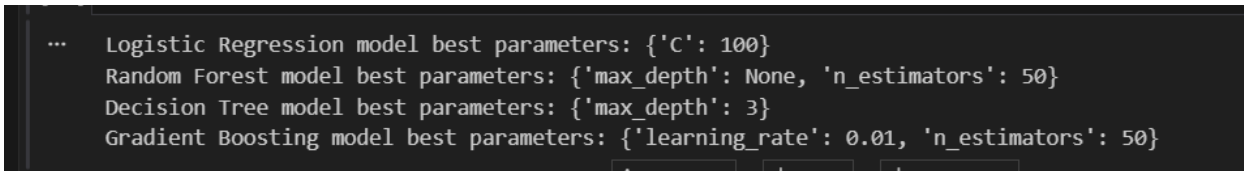
**Data Preparation**
The dataset was split into training (70%) and testing (30%) sets. To address class imbalance (where "Bad" users might outnumber "Good" users or vice versa), SMOTE (Synthetic Minority Over-sampling Technique) was applied to the training data. This ensured the models learned to identify both classes effectively without bias.

**Model Selection and Hyperparameter Tuning**
Four distinct algorithms were selected for evaluation:

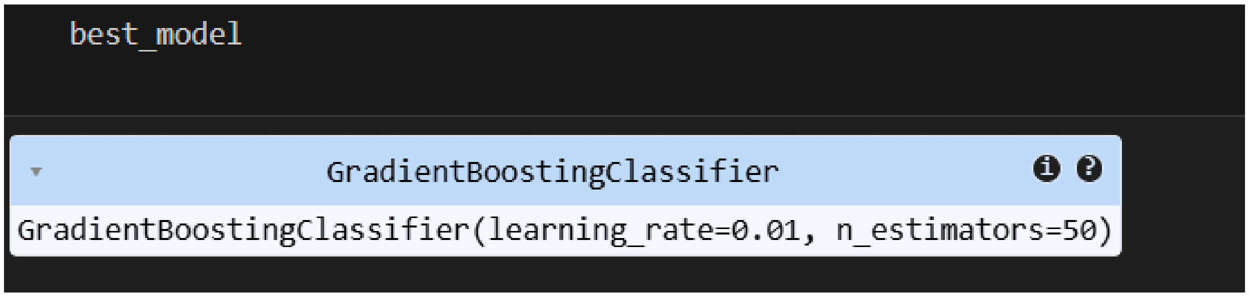
Logistic Regression: For baseline performance and interpretability.
Random Forest: For handling non-linear relationships and feature interactions.
Decision Tree: For simple rule-based classification.
Gradient Boosting: For high predictive accuracy through iterative correction.
A Grid Search with 5-fold Cross-Validation was performed to optimize hyperparameters (e.g., n_estimators, max_depth, learning_rate) for each model.

**Performance Evaluation**
The models were evaluated on the unseen test set using Accuracy, Precision, Recall, F1-Score, and ROC-AUC.
SSS

**Figure 5: Comparative performance metrics of the trained models.**



```
...   Logistic Regression model best parameters: {'C': 100}
      Random Forest model best parameters: {'max_depth': None, 'n_estimators': 50}
      Decision Tree model best parameters: {'max_depth': 3}
      Gradient Boosting model best parameters: {'learning_rate': 0.01, 'n_estimators': 50}
```



```
best_model
```

```
GradientBoostingClassifier
GradientBoostingClassifier(learning_rate=0.01, n_estimators=50)
```

Conclusion: The GradientBoostingClassifier  emerged as the best-performing model, achieving the highest balance of Precision and Recall. This model was serialized (.pkl) for deployment, along with the feature scalers and encoders, ensuring a fully reproducible inference pipeline.