Bank Transaction Analysis: Fraud Detection

Anomaly Detection with Isolation Forest

Widya Catur Utami Putri
Data Portfolio

Complete Code at:
github.com/widyacatur/personalproject

Project Overview

Data Preparation: Loading & Feature Engineering

This section focuses on getting the raw bank transaction data into a usable format This step ensures the data is clean, well-structured, and rich enough for effective analysis and modeling.

Exploratory Data Analysis (EDA): Patterns & Insights

Understand the data's inherent characteristics through visualizations and summary statistics. The goal is to uncover typical spending behaviors, customer demographics, and channel usage patterns.

Anomaly Detection: Methodology & Outlier Characteristics

Using machine learning technique (Isolation Forest Model) to pinpoint unusual transactions. Crucially, we'll then analyze the specific traits of the detected outliers.

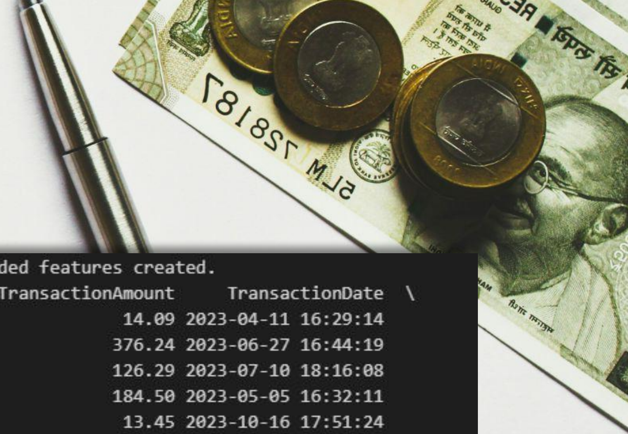
Recommendations & Next Steps

Translate analytical findings into actionable strategies. This includes concrete recommendations based on the outlier characteristics.

Data Loading & Initial Inspection

```
Data columns (total 16 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   TransactionID            2512 non-null   object
 1   AccountID                2512 non-null   object
 2   TransactionAmount        2512 non-null   float64
 3   TransactionDate          2512 non-null   object
 4   TransactionType          2512 non-null   object
 5   Location                 2512 non-null   object
 6   DeviceID                 2512 non-null   object
 7   IP Address               2512 non-null   object
 8   MerchantID               2512 non-null   object
 9   Channel                  2512 non-null   object
 10  CustomerAge              2512 non-null   int64
 11  CustomerOccupation       2512 non-null   object
 12  TransactionDuration      2512 non-null   int64
 13  LoginAttempts            2512 non-null   int64
 14  AccountBalance           2512 non-null   float64
 15  PreviousTransactionDate  2512 non-null   object
dtypes: float64(2), int64(3), object(11)
memory usage: 314.1+ KB
None
(2512, 16)
```

- Dataset: Kaggle Bank Transaction Dataset for Fraud Detection.
- Initial Inspection:
  - We have 2,512 transactions and 16 columns.
  - The Transaction Date and Previous Transaction Date columns were not in the correct datetime format, which will be corrected in the next step.
- Next Steps: Clean and prepare the data for analysis.



Feature Engineering

Date/Time Features:

- Converted date columns to the proper datetime format.
- Extracted the TransactionHour and TransactionDay of the week.
- Calculated DaysSinceLastTransaction to identify unusual gaps between transactions.

Encoding:

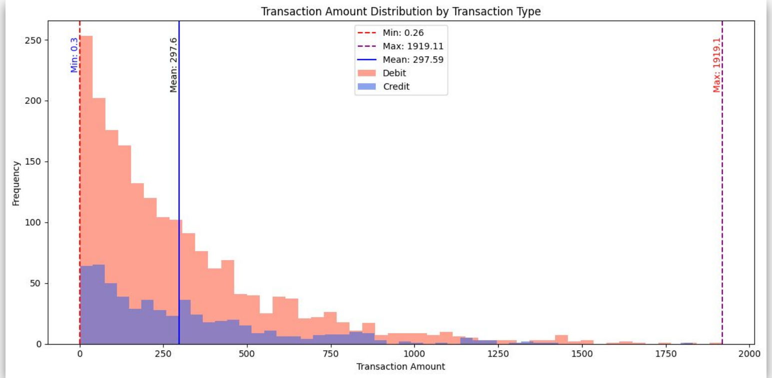
- Used Label Encoding for ordinal data like TransactionDay and CustomerOccupation.
- Used One-Hot Encoding for nominal data like TransactionType and Channel.

New Ratios & Brackets:

- Created BalanceChangeRatio by dividing the TransactionAmount by the AccountBalance.
- Categorized CustomerAge into AgeBracket groups.

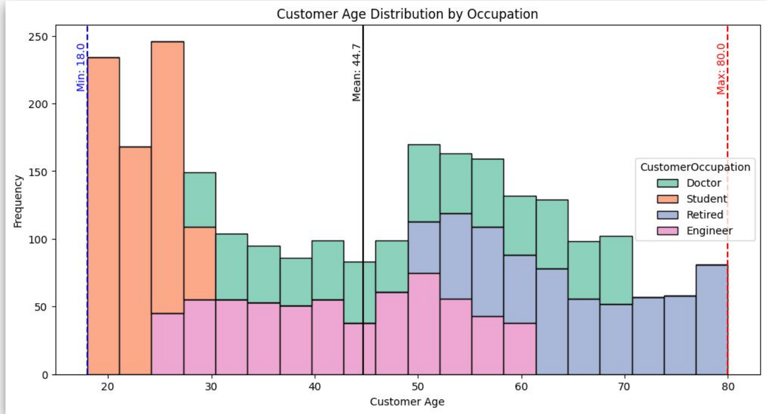
```
DataFrame with One-Hot Encoded features created.
  TransactionID AccountID  TransactionAmount      TransactionDate  \
0      TX000001   AC00128              14.09 2023-04-11 16:29:14
1      TX000002   AC00455             376.24 2023-06-27 16:44:19
2      TX000003   AC00019             126.29 2023-07-10 18:16:08
3      TX000004   AC00070             184.50 2023-05-05 16:32:11
4      TX000005   AC00411              13.45 2023-10-16 17:51:24

  TransactionType   Location DeviceID      IP Address MerchantID Channel
0           Debit  San Diego  D000380  162.198.218.92       M015     ATM
1           Debit    Houston  D000051     13.149.61.4       M052     ATM
2           Debit       Mesa  D000235  215.97.143.157       M009  Online
3           Debit    Raleigh  D000187  200.13.225.150       M002  Online
4          Credit    Atlanta  D000308    65.164.3.100       M091  Online

   CustomerAge CustomerOccupation  TransactionDuration  LoginAttempts  \
0           70             Doctor                   81              1
1           68             Doctor                  141              1
2           19            Student                   56              1
3           26            Student                   25              1
4           26            Student                  198              1

   AccountBalance PreviousTransactionDate  DaysSinceLastTransaction  \
...
1          1.0              0.0             0.0
2          0.0              0.0             1.0
3          0.0              0.0             1.0
4          0.0              0.0             1.0
```

Key Findings from EDA

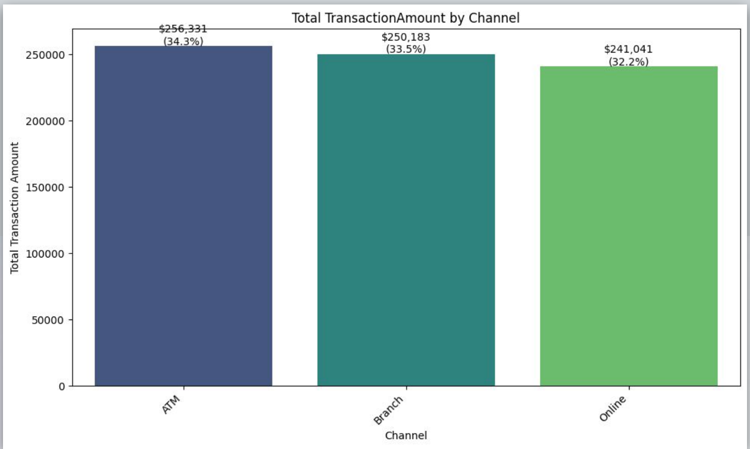


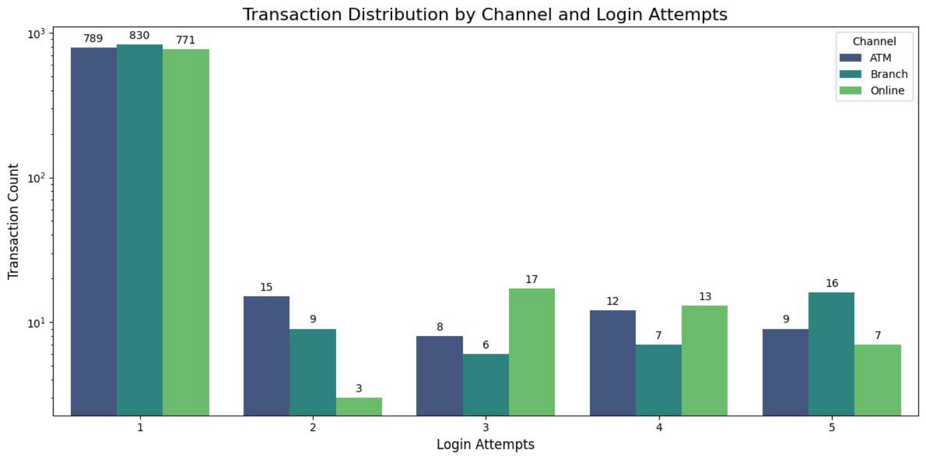
- Transaction Amount: On average, customers spend around $298 per transaction. The amounts range from as low as $0.30 to a high of $1,919.
- Transaction Types: Debit transactions are more frequent and generally have higher values than credit transactions.

- Customer Ages: Range from 18 to 80 years old, with an average age of 44.
- Customer Occupation: Dominated by Students with 657 transactions, followed closely by Doctors and Engineers with 631 and 625 transactions, respectively.



Key Findings from EDA





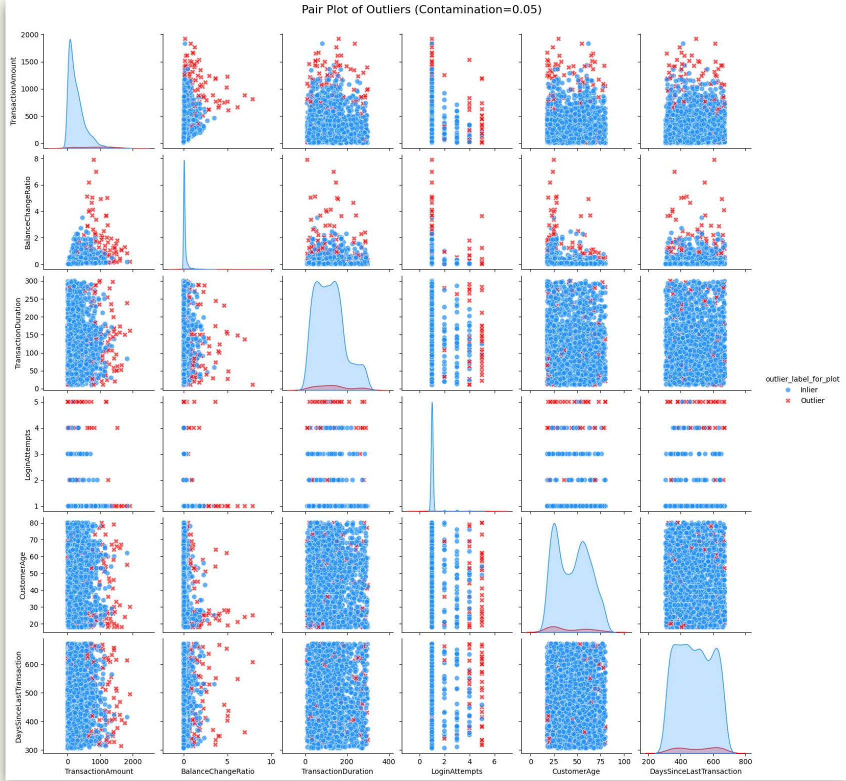
Most customers successfully logged in on their first try. However, a small but notable number of transactions, specifically 32 out of 2,512, required five login attempts, representing approximately 1.3% of the total transactions. A deeper look showed that the Branch channel was the most frequently used channel among these multiple-attempt transactions.

Anomaly Detection: Methodology

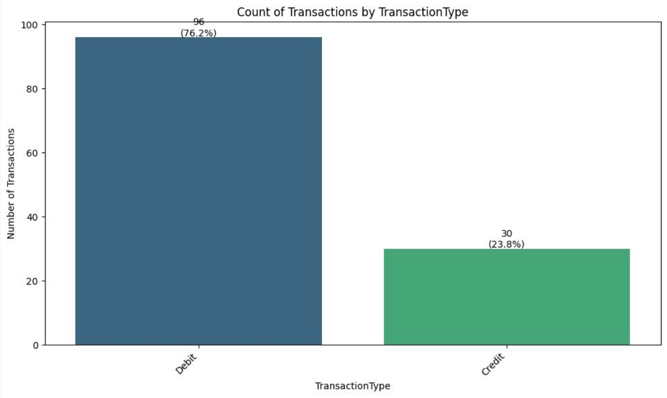
```
Contamination=0.01: Detected 26 outliers.
Contamination=0.02: Detected 51 outliers.
Contamination=0.05: Detected 126 outliers.
Contamination=0.1: Detected 252 outliers.

--- Final Anomaly Detection with chosen contamination ---
Total number of outliers detected: 126
```

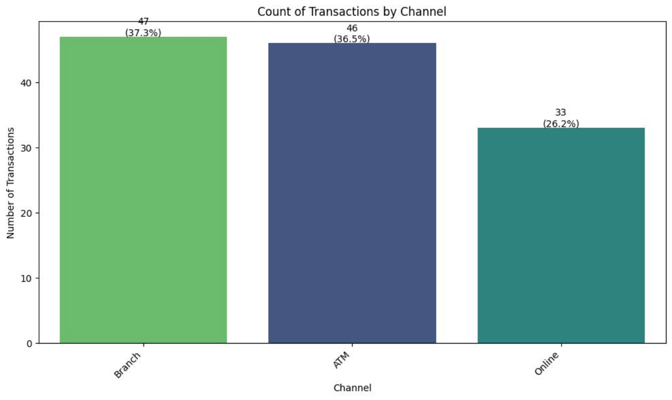
The Isolation Forest model is used to detect unusual transactions by breaking the data into smaller segments, making it easier to isolate entries that differ from the norm.

It includes a parameter called contamination, which defines the expected proportion of outliers. After testing several values, a contamination level of 5% was chosen, resulting in 126 transactions being flagged as potentially suspicious.



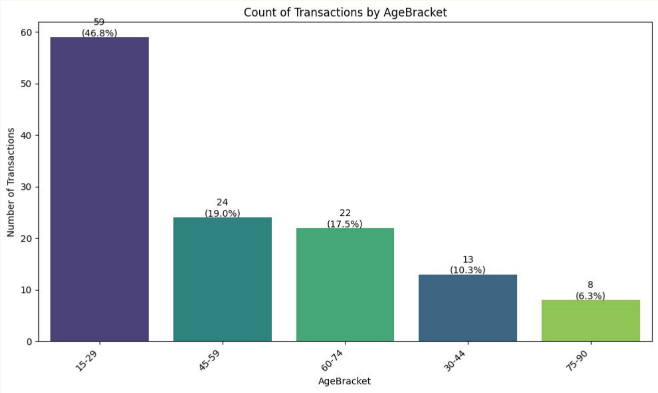
Anomaly Detection: Outlier Characteristics

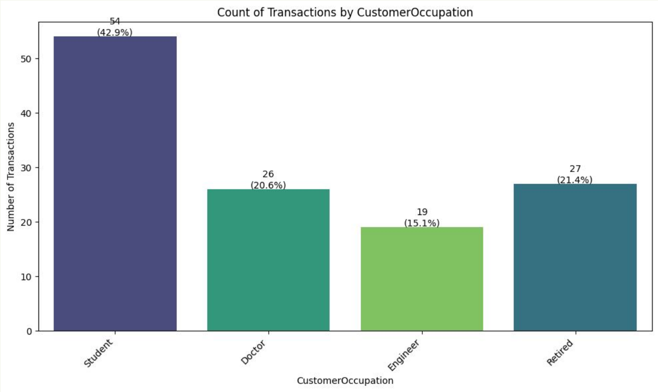




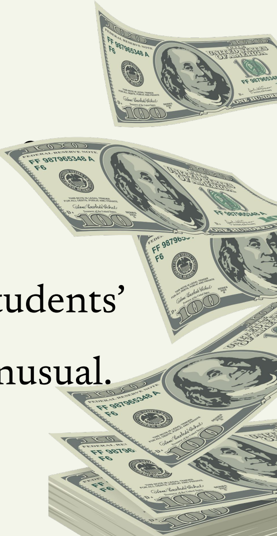
- Transaction Type:
  - 76.2% Debit vs. 23.8% Credit.
  - Debit transactions are disproportionately represented among outliers, aligning with their higher frequency and value in the overall dataset.
- Channel:
  - 37.3% Branch, 36.5% ATM, 26.2% Online.
  - In-person channels (Branch, ATM) show a higher concentration of anomalies, suggesting potential vulnerabilities or unique patterns.

Anomaly Detection: Outlier Characteristics

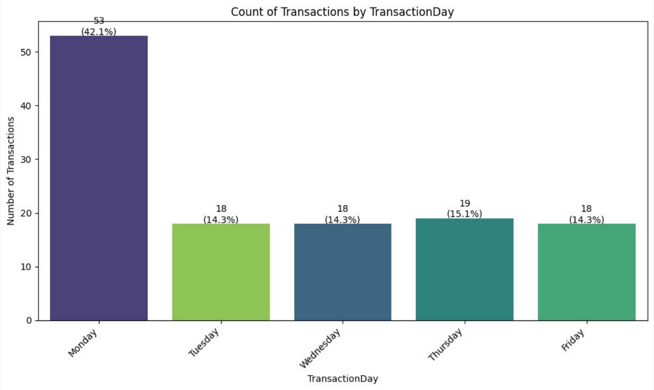


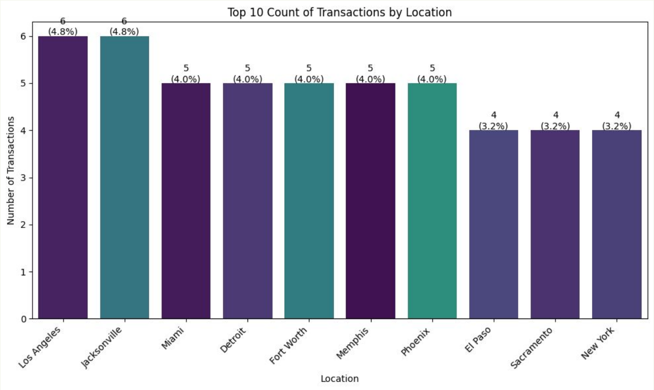


- Age Bracket:
  - 46.8% from 15-29 age group.
  - Younger customers (and students) are significantly overrepresented in the outlier group.
- Customer Occupation:
  - 42.9% Students, 21.4% Retired, Doctors.
  - Consistent with age bracket findings, students' transactions are frequently flagged as unusual.

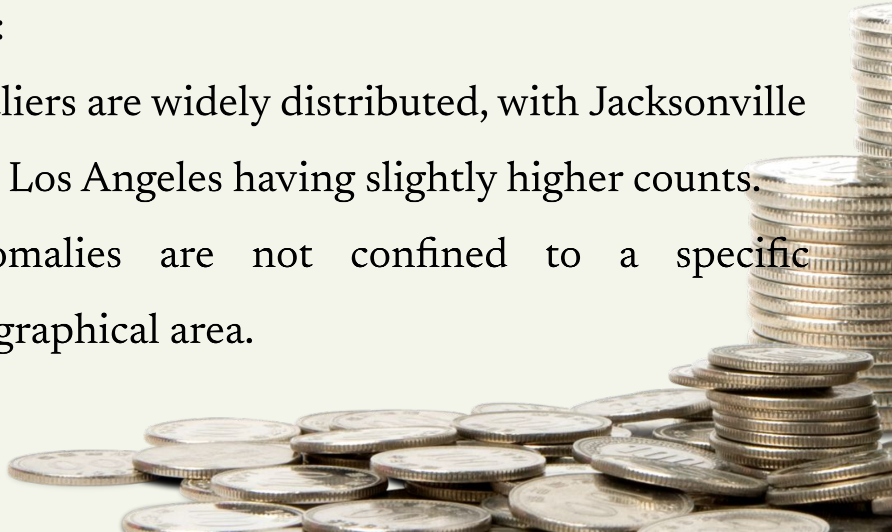


Anomaly Detection: Outlier Characteristics





- Transaction Day:
  - 42.1% occurred on Monday.
  - A strong temporal pattern, with a high spike in anomalies at the start of the week.
- Location:
  - Outliers are widely distributed, with Jacksonville and Los Angeles having slightly higher counts.
  - Anomalies are not confined to a specific geographical area.



Recommendations & Next Steps

Based on the analysis, here are actionable recommendations to secure transaction processes:

- **Prioritize Debit Transaction Monitoring**: Implement stricter real-time monitoring and anomaly detection rules specifically for debit transactions, given their high outlier rate.
- **Strengthen ATM & Branch Security Protocols**: Review and enhance security measures for in-person transactions, particularly at ATMs and branches, considering their higher susceptibility to unusual activity and multiple login attempts.
- **Implement Targeted Scrutiny for Younger Demographics**: Develop specific alert thresholds or review processes for transactions originating from the 15-29 age bracket and Students, as their transactions are frequently flagged as unusual.
- **Investigate Monday Activity Spikes**: Conduct deeper forensic analysis into transactions occurring on Mondays to understand the root causes of the elevated anomaly rate on this specific day.
- **Develop a Tiered Alert System**: Create an automated system that generates different alert levels based on a combination of high-risk factors (e.g., a high-value debit transaction from an ATM by a young student on a Monday).

Thank you

Widya Catur Utami Putri
Data Portfolio

Complete Code at:
github.com/widyacatur/personalproject