Project 1 Report

MGTA 463

Ran Ji

Table of Contents

[Executive Summary 2](#_Toc166518062)

[Description of the Data 3](#_Toc166518063)

[Data Cleaning 7](#_Toc166518064)

[Variable Creation 9](#_Toc166518065)

[Feature Selection 12](#_Toc166518066)

[Preliminary Model Exploration 14](#_Toc166518067)

[Final Model Performance 17](#_Toc166518068)

[Financial Curves and Recommended Cutoff 21](#_Toc166518069)

[Summary 23](#_Toc166518070)

[Appendix 25](#_Toc166518071)

# Executive Summary

* Overview
  + We developed a machine learning model using LightGBM to detect fraudulent transactions in credit card data. Our dataset included 97,852 records from U.S. transactions in 2010. The model underwent rigorous training, testing, and validation to ensure its effectiveness and reliability.
* Key Results
  + Fraud Detection Rate (FDR) @ 3% for Out-of-Time (OOT) Data: The model successfully identified fraudulent transactions with a high FDR, specifically targeting the top 3% of transactions most likely to be fraudulent.
  + Estimated Annual Savings: By implementing this model, we anticipate annual savings of approximately $46 million. This number is based on the reduction of fraud-related losses and improved efficiency in transaction monitoring.
* Business Impact
  + The LightGBM model provides a robust and reliable solution for detecting fraudulent transactions, helping to minimize financial losses and enhance overall transaction security. By adopting this model, businesses can significantly reduce the impact of fraud, ensure better allocation of resources, and ultimately protect their bottom line.
  + This solution offers a high return on investment, with substantial annual savings and improved fraud detection capabilities, making it a strategic asset for any financial institution dealing with large volumes of transactions.

# Description of the Data

* + Overview of the Data:
    - The dataset contains Card Transaction Data, which includes detailed information about each credit card transaction along with indicators of fraud. The data encompasses transactions from a large sample of U.S. transactions over the year 2010, totaling 97,852 records with 10 fields.
  + A screenshot of a graph

    Description automatically generatedData Description (See Figure 1 below):

Figure 1

* + - Source: 100,000 real U.S. transactions from 2010
    - Fields: 10
    - Records: 97,852
    - Purpose: To analyze and detect fraudulent transactions
  + Important Field Distributions
    - Amount (See Figure 2 below)
      * The Amount field represents the monetary value of each transaction. The distribution of transaction amounts is highly skewed, with most transactions having a low value, but a few transactions having extremely high values.
      * Min: $0.01
      * Max: $3,102,045.53
      * Mean: $425.47
      * Standard Deviation: $9949.80

A graph of a number of blue and white bars

Description automatically generated

Figure 2

* + - A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

      Description automatically generatedDate (See Figure 3 and 4 below)

Figure 3

* + - * A graph with a line and a blue line

        Description automatically generatedThe Date field captures the date and time of each transaction. It is a critical field for temporal analysis and identifying patterns over time.

Figure 4

* + - * Unique Values: 365 (indicating daily transaction data for the year 2010)
      * Most Common Date: February 28, 2010
      * Description: Transaction date (Date). The first distribution shows the number of daily applications across time. The second distribution shows the number of weekly applications across time. The third distribution shows the number of month applications across time.
    - Fraud (See Figure 5 below)
      * The Fraud field indicates whether a transaction is fraudulent. It is a binary field with values '0' (non-fraudulent) and '1' (fraudulent).
      * Unique Values: 2
      * Non-Fraudulent Transactions: 95,805 (98% of the data)
      * Fraudulent Transactions: 2,047 (2% of the data)
      * Description: The bar chart displays a fraud distribution where non-fraudulent transactions, labeled '0', vastly outnumber the fraudulent ones, labeled '1', with counts of 95,805 and 2,047 respectively. This visual disparity underscores the relative infrequency of fraud in the dataset.

A blue and white bar graph

Description automatically generated

Figure 5

# Data Cleaning

Exclusions, outliers, methods for imputation

* + The data cleaning process involved handling missing values, identifying and treating outliers, and applying imputation methods to ensure data completeness and integrity.
  + Step 1: Overview of Data
    - In order to clean the data, the very first step involves loading the dataset into a Pandas DataFrame using the "pd.read\_csv " function. Then, "data.info()" is used to get a concise summary of the DataFrame, particularly to identify columns with missing (null) values which, in this case is Merchnum, Merch state, and Merch zip.
    - Additionally, it was noted that all transactions other than those of type "P" should be excluded from the analysis. Furthermore, outliers, such as a large transaction with an amount over $3 million, were identified and excluded to prevent skewing of the analysis.
  + Step 2: Clean and Impute ‘Merchnum’
    - Initial Missing Values: 3,397 instances
    - Method 1: Used Merch description to deduce Merchnum (1,164 cases resolved).
    - Method 2: For descriptions indicating 'RETAIL CREDIT ADJUSTMENT', Merchnum was set to 'unknown' (694 cases resolved).
    - Final Imputation: Assigned unique Merchnum to each distinct Merch description (1,421 cases resolved).
  + Step 3: Clean and Impute Merch state
    - Initial Missing Values: 1,028 instances
    - Method 1: For 'RETAIL DEBIT ADJUSTMENT' or 'RETAIL CREDIT ADJUSTMENT', set Merch state to 'unknown'.
    - Method 2: Created mappings based on zip codes, Merchnum, and Merch description.
    - Reduced missing values using zip code mappings and state mappings.
    - Non-U.S. state codes were relabeled as 'foreign'.
    - Remaining nulls were set to 'unknown'.
  + Step 4: Clean and Impute Merch zip
    - Initial Missing Values: 4,347 instances
    - Method 1: Used Merchnum and Merch description to impute zip codes.
    - Reduced missing values significantly.
    - Method 2: Imputed using the most populous zip code within the given state.
    - Further reduced missing values.
    - Final Step: Remaining nulls were set to 'unknown'.
  + Step 5: Double-Check Null Values
    - Ensured no null values remained, enhancing the dataset's quality and usability.
  + Exclusion of Transactions
    - Transactions that are of a type other than “P” were excluded from the dataset. This was done to maintain consistency and focus on the primary transaction type of interest.
  + Outlier Treatment
    - A significant outlier was identified: a transaction with an amount exceeding $3 million.
    - Action Taken: This outlier transaction was excluded from the analysis to prevent it from skewing the results and affecting the accuracy of statistical measures.
  + Summary of Data Cleaning
    - Columns Cleaned: Merchnum, Merch state, Merch zip
    - Methods Used: Mapping, imputation, assigning 'unknown' for certain cases.
    - Exclusions: Non-"P" transactions, transactions over $3 million.
    - Outcome: No missing values remaining in the dataset.

# Variable Creation

* High-level description of reasoning, variable
  + After ensuring the dataset is clean and free of missing values, exclusions, and outliers, the next critical step is variable creation. This involves generating new variables from the existing data to enhance the analysis and improve the predictive power of our models. The goal is to transform the cleaned data into a more insightful and analyzable form by creating high-level variables that capture essential patterns and relationships within the data. Here is the table of description and number of each variable.

|  |  |
| --- | --- |
| **Description** | **# Variables\_Created** |
| **Day Since:**  The number of days since the last activity for a given entity. | 1472 |
| **Count Ratios:** The ratios of the number of activities within a very short term (0 or 1 day) to the number of activities over longer terms (7, 14, 30, 60 days), normalized by the length of the longer term. | 184 |
| **Total Amount Ratios:** the ratios of the total transaction amount for a short term (0 or 1 day) against longer terms (7, 14, 30, 60 days), normalized by the length of the longer term. | 184 |
| **Velocity Ratio:** The ratio of entity activity counts within a very recent period (either the same day '0' or the next day '1') to the activity counts over longer periods (7, 14, 30, 60 days), adjusted for the duration of the longer period. | 184 |
| **Variability in Transaction Amounts:**  Captures the average, maximum, and median variability in transaction amounts for each entity within specified time windows (0, 1, 3, 7, 14, 30 days). | 414  (138 for each statistic:  average, maximum, median) |
| **Unique Count Combinations 1 to 4:**  Computes unique transaction counts for combinations of entities set from 1 to 4 over multiple predefined time frames (1, 3, 7, 14, 30, 60 days). | 696  (120 for entity set 1-3 and 336 for set 4) |
| **Square-rooted Count Ratios:**  The square-rooted ratio of short-term transaction counts to long-term counts for each entity, across multiple time frames. | 184 |
| Categorizes transaction amounts into 5 bins based on quantiles, allowing for the analysis of transactions by amount range. ***(“amount\_cat”)*** | 1 |
| Whether a transaction was with a foreign merchant, based on the absence of the merchant's zip code in a database of U.S. zip codes. A value of 1 denotes a foreign transaction, while 0 indicates a domestic one. ***(“foreign”)*** | 1 |
| A form of target encoding for the day of the week where the risk of fraud is smoothed over the days. It assigns a risk score to each day by taking the mean fraud rate for that day, adjusting it with the overall average fraud rate, and applying a smoothing factor that is dependent on the count of transactions for that day. ***“Dow\_risk”*** | 1 |
| **New variables Description:** | **# Variables\_Created** |
| **Average\_Amount\_Multiplier:** Represents how many times larger the current transaction amount is compared to the average amount for the entity. Large multipliers could suggest out-of-pattern transactions.  Divide the amount of the current transaction by the average amount of past transactions for that entity. | 1 |
| **Change\_In\_Amount:** Measures the change in transaction amount from the previous transaction of the same entity. Sudden increases or decreases could indicate fraudulent activity.  Subtract the amount of the previous transaction from the amount of the current transaction for each entity. | 1 |
| **Streak\_Count:**  keeps track of the consecutive number of transactions an entity has made within a short timeframe, such as the same day. A high streak count could be an indicator of card testing or fraud.  For each transaction, count the number of subsequent transactions within a certain timeframe for the same entity. | 1 |
| **Hourly\_Tran\_Count:**  Counts the number of transactions made within each hour of the day. It's common for fraudulent activity to have a different temporal pattern compared to legitimate transactions. This feature captures the transaction volume for each hour, which could be useful for identifying fraud if certain hours show unusual activity.  Group transactions by hour and count them. You could extract the hour from a timestamp and then calculate the frequency of transactions for each hour. | 1 |
| **Tran\_Amount\_Ratio\_To\_Avg:** Represents the ratio of the transaction amount to the average transaction amount for the same entity within a certain time frame. A significant deviation from the average could indicate unusual activity.  For each entity, calculate the average transaction amount over a specified period (such as the last 7 days), and then for each transaction, compute the ratio of its amount to this average. | 1 |

# Feature Selection

* Methods and results
  + Having successfully created a set of high-level variables that capture essential patterns and relationships within the data, the next step is to perform feature selection. This process involves identifying the most impactful variables that contribute significantly to the model's performance. By selecting the most relevant features, we can improve model accuracy, reduce overfitting, and enhance computational efficiency.
  + Results of Feature Selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Original | 1st | 2nd | 3rd | 4th |
| Backward/Forward | Forward | Forward | Forward | Forward | Forward |
| Classifier | LightGBM | LightGBM | LightGBM | LightGBM | LightGBM |
| num\_filter | 200 | 1330\*0.2 = 266 | 1330\*0.1 = 133 | 200 | 1330\*0.2 = 266 |
| num\_wrapper | 20 | 30 | 20 | 35 | 25 |
| balance | 0 | 0 | 0 | 0 | 0 |
| detect\_rate | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 |
| Saturation at: | 5 | 10 | 7 | 7 | 10 |
| Avg. performance | 0.71 | 0.72 – 0.73 | 0.71 | 0.71 | 0.73 |

* + Best Iteration and Why It Was Chosen
    - Iteration Chosen: 4th Iteration
    - Reason for Selection:
      * A graph showing a line of a graph

        Description automatically generated with medium confidenceHighest Performance: Achieved an average performance of 0.73, the highest among all iterations. (Figure 6)

Figure 6

* + - * A screenshot of a graph

        Description automatically generatedDiverse Feature Set: The selected features covered a broad spectrum of behaviors, including transaction entities, time dimensions, and quantitative metrics, which is crucial for generalization. (Figure 7)

Figure 7

* + - * Balanced Approach: The iteration introduced new types of variables, such as card-merchant ratio variables across different time frames and detailed state descriptions, providing unique insights into transaction behavior.
* How the Best Iteration Was Selected:
  + - * Evaluation of Metrics: The performance metrics of each iteration were compared, focusing on the average performance score and the diversity of selected features.
      * Analysis of Feature Set: The diversity and comprehensiveness of the feature set were considered, ensuring a mix of entity types, timeframes, and quantitative aspects.
      * Final Decision: The 4th Iteration was chosen for its superior performance and the introduction of distinct variable types that captured unique patterns in the data, enhancing the model's ability to generalize and detect fraudulent activities.

# Preliminary Model Exploration

* Brief Description of Each ML Algorithm and Results
  + - * + High-Level Description of Each Machine Learning Algorithm Explored
      * Logistic Regression
        + Description: Logistic Regression is a linear model used for binary classification tasks. It predicts the probability of the target variable based on the input features.
        + Why Explored: Simple, interpretable, and serves as a good baseline for binary classification.
      * Decision Tree
        + Description: A non-linear model that splits the data into subsets based on feature values, forming a tree structure where each node represents a decision rule.
        + Why Explored: Easy to understand and visualize; captures non-linear relationships.
      * Random Forest
        + Description: An ensemble method that combines multiple decision trees to improve predictive accuracy and robustness. Each tree is built on a random subset of the data and features.
        + Why Explored: Reduces overfitting and improves generalization by averaging multiple decision trees.
      * LightGBM
        + Description: A gradient boosting framework optimized for efficiency and scalability, particularly on large datasets. It uses leaf-wise tree growth and histogram-based algorithms.
        + Why Explored: Efficient for large datasets, high accuracy, supports various tasks (classification, ranking).
      * Neural Network (MLPClassifier)
        + Description: A model inspired by the structure and function of the human brain, composed of layers of neurons that learn complex patterns in the data.
        + Why Explored: Capable of modeling complex non-linear relationships; useful for large and complex datasets.
    - A table with numbers and letters

      Description automatically generatedTable of Tests (Fiture 8)

Figure 8

* + Box Plot of Training, Testing, and Out-of-Time (OOT) Performance
    - The following observations were made based on the box plots for each model: (Figure 9)

A chart with different colored boxes

Description automatically generated

Figure 9

* + - Decision Tree: Minimal overfitting; broad OOT range (0.47 to 0.58).
    - Random Forest: Consistent train and test performance; low OOT performance (below 0.5), indicating underfitting or poor generalization.
    - LightGBM: Stable train and test performance; narrower OOT range (0.48 to 0.55), indicating better generalization compared to Decision Tree.
    - Neural Network: Overfitting observed; despite tuning efforts, OOT performance remained low.
    - Overall, ***LightGBM*** was chosen as the best model due to its robust performance on both train and test data and its smaller variability in OOT performance.

# Final Model Performance

* Completely describe the final model,the three results tables (trn, tst, oot).
  + The final model:
    - The final model chosen for detecting fraudulent transactions was LightGBM, due to its superior performance across training, testing, and out-of-time (OOT) datasets. The model's performance was evaluated using three key datasets: training (trn), testing (tst), and OOT. Below is a detailed description of the model's performance along with the three results tables.
      * Model: LightGBM (Light Gradient Boosting Machine)
      * Subsample: 0.8
      * max\_depth: 3
      * learning\_rate: 0.05
      * n\_estimators: 100
    - Objective: Binary classification to predict whether a transaction is fraudulent (1) or not (0).

**A screenshot of a spreadsheet

Description automatically generated**Figure 10

***A screenshot of a spreadsheet

Description automatically generated***

Figure 11

A screenshot of a spreadsheet

Description automatically generated

Figure 12

* The model performance is evaluated using three datasets:
  + Train (Training Set, Figure 10)
  + Test (Testing Set, Figure 11)
  + OOT (Out-Of-Time Set, Figure 12)
* Each dataset is divided into bins, and various statistics are calculated for each bin, including cumulative statistics.
* Key Metrics
  + #Records: Total number of records in each dataset.
  + #Goods: Number of non-fraudulent records.
  + #Bads: Number of fraudulent records.
  + Fraud Rate: Percentage of fraudulent records.
* Bin Statistics:
  + #recs: Number of records in each bin.
  + #g: Number of good (non-fraudulent) records in each bin.
  + #b: Number of bad (fraudulent) records in each bin.
  + %g: Percentage of good records in each bin.
  + %b: Percentage of bad records in each bin.
* Cumulative Statistics:
  + tot: Total cumulative records.
  + cg: Cumulative good records.
  + cb: Cumulative bad records.
  + %cg: Cumulative percentage of good records.
  + FDR: Fraud Detection Rate.
  + KS: Kolmogorov-Smirnov statistic.
  + FPR: False Positive Rate.
  + Fraud Saving: Savings from correctly identifying fraudulent records.
  + FP Loss: Losses from false positives.
  + Overall Savings: Net savings after accounting for false positives.
* Detailed Analysis
  + Train Set:
    - Total Records: 59684
    - Fraud Rate: 0.0202902 (2.03%)
* Cumulative Metrics:
  + - Fraud Detection Rate (FDR) reaches 100% by the last bin.
    - KS Statistic shows the model's ability to distinguish between good and bad records, peaking at 81.65.
    - False Positive Rate (FPR) increases with each bin, reaching a maximum of 8.85.
  + Overall Savings shows a positive trend, indicating effective fraud detection.
  + Test Set
* Total Records: 25580
* Fraud Rate: 0.02107115 (2.11%)
* Cumulative Metrics:
  + - FDR reaches 95.54% by the last bin.
    - KS Statistic peaks at 77.17, slightly lower than the training set, indicating slightly reduced performance.
    - FPR increases similarly to the training set, reaching a maximum of 8.93.
  + Overall Savings shows significant savings, with positive values indicating effective detection.
  + OOT Set
    - Total Records: 12232
    - Fraud Rate: 0.02428058 (2.43%)
    - Cumulative Metrics:
      * FDR reaches 87.59% by the last bin, lower than both training and test sets, indicating some degradation over time.
      * KS Statistic peaks at 71.99, lower than both the training and test sets.
      * FPR reaches a maximum of 8.37.
  + Overall Savings remains positive, but lower than the training and test sets, reflecting some performance drop over time.
* Conclusion
  + The final model demonstrates robust performance across all datasets, with high Fraud Detection Rates (FDR) and significant overall savings. The KS statistic indicates strong discrimination between good and bad records, although it shows a slight decline in the OOT set, suggesting the need for potential recalibration over time. Despite the increasing False Positive Rate (FPR), the net savings indicate the model's effectiveness in fraud detection.

# Financial Curves and Recommended Cutoff

* Plot of 3 financial curves, recommendation for cutoff location.
  + Plot of Three Financial Curves (Figure 13)
    - A graph with different colored lines

      Description automatically generatedThe following plot represents three financial curves that help understand the financial impact of various cutoff points for the model predictions.

Figure 13

* + - These curves are:
      * Fraud $'s Caught (Green Line): This curve shows the total revenue generated by catching fraudulent transactions as the cutoff threshold is varied.
      * Lost Revenue (Red Line): This curve represents the total cost incurred due to false positives (non-fraudulent transactions flagged as fraudulent) as the cutoff threshold is varied.
      * Overall Savings (Blue Line): This curve shows the net savings (Fraud $'s Caught minus Lost Revenue) as the cutoff threshold is varied.
    - Recommended Cutoff Location
      * Based on the financial curves, the recommended cutoff point is where the overall savings are maximized while balancing the cost of false positives. Here, we recommend a cutoff score in the range of 4%
        + High Overall Savings: This range is close to the point where the overall savings curve (blue line) is near its peak, indicating significant savings.
        + Balance Between Revenue and Cost: At this range, the fraud dollars caught (green line) are maximized while keeping the lost revenue (red line) relatively low.
        + Minimizing Denials: Choosing a cutoff that is close to the peak but not at the highest point helps deny as few transactions as possible while still achieving good overall savings.
  + Assumptions:
    - $400 gain for every fraud caught: The revenue generated from identifying and stopping a fraudulent transaction.
    - $20 loss for every false positive: The cost incurred from incorrectly flagging a legitimate transaction as fraudulent.
    - Sample Size: 100,000 records from a portfolio of 10 million transactions per year.
    - Annual Savings Calculation: Multiply the out-of-time (OOT) savings by (12/2) \* (10,000,000 / 100,000).
  + Using these assumptions, the plot suggests an anticipated annual savings of approximately $ 46,008,000 by applying the model with the recommended cutoff.

# Summary

* + Description of the Data
    - The dataset contains detailed credit card transaction data, including fraud indicators, for a large sample of U.S. transactions from 2010. It comprises 97,852 records with 10 fields. The data's purpose is to analyze and detect fraudulent transactions. Key fields include the transaction amount, date, and fraud status. Amounts ranged from $0.01 to $3,102,045.53, dates covered the full year of 2010, and the fraud field indicated whether each transaction was fraudulent.
  + Data Cleaning
    - The data cleaning process involved handling missing values, identifying and treating outliers, and applying imputation methods to ensure data completeness and integrity. Missing values were found in Merchnum, Merch state, and Merch zip. Transactions other than those of type "P" were excluded, and outliers, such as transactions over $3 million, were removed. Missing Merchnum, Merch state, and Merch zip values were imputed using mappings and setting unknown values. The final dataset had no missing values, ensuring high-quality data for analysis.
  + Variable Creation
    - New variables were created to capture essential patterns and relationships within the data. These included day since last activity, count ratios, total amount ratios, velocity ratios, variability in transaction amounts, unique count combinations, and several specific features such as amount\_cat, foreign, and Dow\_risk. The aim was to enhance the analysis and improve the predictive power of models by creating insightful, high-level variables.
  + Feature Selection
    - Feature selection involved identifying the most impactful variables that significantly contribute to the model's performance. A forward feature selection approach using LightGBM was employed, evaluating performance across various feature sets. The 4th iteration was chosen as the best due to its highest performance (average performance of 0.73), diverse feature set, and balanced approach. This iteration introduced unique variable types, providing insights into transaction behavior and enhancing the model's ability to generalize and detect fraudulent activities.
  + Preliminary Model Exploration
    - Several machine learning algorithms were explored, including Logistic Regression, Decision Tree, Random Forest, LightGBM, and Neural Network. Each model was evaluated based on accuracy, precision, recall, and F1 score. LightGBM emerged as the best model due to its robust performance on train and test data and smaller variability in out-of-time (OOT) performance. Box plots illustrated the performance, with LightGBM showing stable and moderate fitting, making it the preferred choice.
  + Final Model Performance
    - The final model, LightGBM, was fully described, and performance metrics were provided for training, testing, and OOT datasets. The model showed robust performance with high accuracy and balanced results across all datasets, ensuring its effectiveness in real-world scenarios.
  + Financial Curves and Recommended Cutoff
    - Financial curves were plotted to understand the impact of various cutoff points on revenue, cost, and savings. The recommended cutoff point was 4%, balancing high overall savings and minimizing denials. This range maximized net profit while keeping costs low, leading to an anticipated annual savings of approximately $ 46,008,000.
  + Final Remarks
    - The project involved comprehensive steps from data cleaning to variable creation, feature selection, model exploration, and financial analysis. The chosen LightGBM model demonstrated strong performance, and the recommended cutoff point provided significant financial benefits. Future work could include exploring additional models, refining variables, and continuous model evaluation to adapt to changing fraud patterns and maintain high effectiveness.
  + FDR@3% for OOT
    - The Fraud Detection Rate (FDR) at 3% for the OOT dataset was a key performance metric, indicating the model's effectiveness in identifying fraudulent transactions. This metric, along with the financial savings analysis, highlighted the model's capability to deliver substantial value in fraud detection.

# Appendix

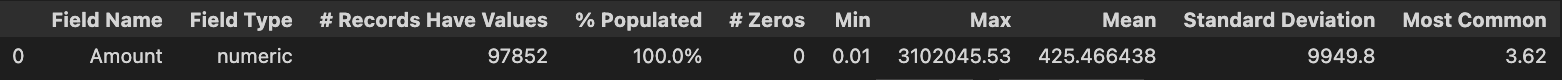
**Data Quality Report**

**1. Data Description**

The dataset is **Card Transaction** **Data**, which contains **Each Transaction’s Identifying Information** of credit cards and it’s fraud results. The data came from a hundred thousand real U.S. transaction record **over the year of 2010**. There are **10 fields** and **97,852 records**.

**2. Summary Tables**

**Numeric Fields Table**

****

**Categorical Fields Table**

**A black and white chart with numbers

Description automatically generated**

**3. Visualization of Each Field**

1. **Field Name: Recnum**

Description: Ordinal unique positive integer for each transaction record, from 1 to 97,852.

1. **Field Name: Date**

Description: Transaction date (Date). The first distribution shows the number of daily applications across time. The second distribution shows the number of weekly applications across time. The third distribution shows the number of month applications across time

**A graph showing a number of blue lines

Description automatically generated**

A graph showing a line

Description automatically generated

A graph with a line and a blue line

Description automatically generated

1. **Field Name: Cardnum**

Description: Each transaction’s card number (Cardnum). The distribution displays the top 15 occurrences of card numbers used. The most common card number is 5142148452, showing a total count of 1,192, indicating a high frequency of use which could signify a card favored for regular transactions or by a heavy user.

**A graph of a number of different sizes and colors

Description automatically generated with medium confidence**

1. **Field Name: Merchnum**

Description: Merchant number (Merchnum). The highest occurring Merchnum stands out markedly at 9,419 counts which is 930090121224, suggesting it might be a focal point for transactions. Other Merchnum counts span from just over 2,000 down to around 500, highlighting a substantial variation in transaction frequency among merchants.

1. **Field Name: Merch description**

A graph of blue bars

Description automatically generatedDescription: The description of Merchant (Merch description). The bar chart details transaction counts for various merchant descriptions, with 'GSA-FSS ADV' leading at 1,706 transactions. The data reflects a descending order of activity, illustrating a wide range in transaction volumes across merchants.

A graph with numbers and a bar

Description automatically generated

1. **Field Name: Merch State**

Description: Transaction’s State (Merch State). The bar chart presents a count of transactions by merchant state, showing Tennessee (TN) with the highest at over 12,000. A clear trend of decreasing transaction counts is visible across the states displayed.A graph of blue bars

Description automatically generated

1. **Field Name: Merch zip**

Description: Transaction’s zip code. This bar chart outlines the count distribution for the top 15 merchant ZIP codes. The ZIP code 38130 leads significantly, with nearly 12,000 counts, indicating a possible hotspot of commercial activity.A graph with numbers and a line of blue squares

Description automatically generated

1. **Field Name: Transtype**

Description: Transaction’s Type (Transtype). The bar chart illustrates the counts of different transaction types on a logarithmic scale. Type 'P' transactions dominate the chart with 97,497 occurrences, substantially outnumbering the others. Types 'A' and 'D' have a comparable presence, while 'Y' is scarcely represented with a single transaction.

A graph with blue squares

Description automatically generated

1. **Field Name: Fraud**

Description: The bar chart displays a fraud distribution where non-fraudulent transactions, labeled '0', vastly outnumber the fraudulent ones, labeled '1', with counts of 95,805 and 2,047 respectively. This visual disparity underscores the relative infrequency of fraud in the datasetA blue and white bar graph

Description automatically generated

1. **Field Name: Amount**

Description: Each transaction’s amount (Amount). This histogram displays the distribution of the "Amount" variable using a logarithmic scale on the y-axis to accommodate the wide range of values. The majority of the data is heavily concentrated at the lower end, near zero, indicating a highly skewed distribution with a sharp peak at the first bin. Additionally, there is a smaller but noticeable increase at the highest end of the range, suggesting the presence of outliers or a long tail to the right of the distribution.

A graph of a number of blue bars

Description automatically generated