MULTIVARIATE ANALYSIS PROJECT REPORT
Sl-505

Submitted by:

Sonu Gupta 22N0062
Deepak Kumar 22N0058
Kamesh Dubey 22N0088

Supervised by:
Prof. Siuli Mukhopadhyay
Associate Professor
Department of Mathematics
11T Bombay

Bank Customer Segmentation Report

Data Information

Dataset Name:

Bank Customer Segmentation Data

Overview:

The dataset contains information about bank customers' behavior and transactions.

Column Details

- **BALANCE:** Total account balance of the customer.
- **BALANCE_FREQUENCY:** Frequency of updating the balance.
- **PURCHASES:** Total amount of purchases made by the customer.
- **ONEOFF_PURCHASES:** Amount of purchases for a single payment.
- **INSTALLMENTS_PURCHASES:** Amount of purchases paid in installments.
- **CASH_ADVANCE:** Total cash advance taken by the customer.
- **PURCHASES_FREQUENCY:** Frequency of purchases.
- **ONEOFF_PURCHASES_FREQUENCY:** Frequency of one-off purchases.
- **PURCHASES_INSTALLMENTS_FREQUENCY:** Frequency of installment purchases.
- **CASH_ADVANCE_FREQUENCY:** Frequency of cash advances.
- **CASH_ADVANCE_TRX:** Number of transactions for cash advances.
- **PURCHASES_TRX:** Number of purchase transactions.
- **CREDIT_LIMIT:** Credit limit of the customer.
- **PAYMENTS:** Total payments done by the customer.
- **MINIMUM_PAYMENTS:** Minimum payments made by the customer.
- **PRC_FULL_PAYMENT:** Percentage of full payment made by the customer.
- **TENURE:** Number of months as a customer.

Objective of the Project

The main aim of the ”Bank Customer Segmentation” project is to use smart ways like PCA, Factor Analysis, and grouping to sort out bank customers based on how they handle their money. This helps in making the data simpler, finding important patterns, and then splitting customers into different groups. The end goal is to really understand and separate different types of customers, so the bank can create special plans and services just for them. This way, the bank can make its customers happier and more engaged by offering things that suit them better.

Data Exploration

In this project, we're dealing with a dataset containing information about bank customers. There are 17 different aspects we're studying about each customer. Overall, our dataset consists of 8950 rows and 18 columns.

CUST_ID	C10001	C10002	C10003	C10004	C10005
BALANCE	40.900749	3202.467416	2495.148862	1666.670542	817.714335
BALANCE_FREQUENCY	0.818182	0.909091	1.000000	0.636364	1.000000
PURCHASES	95.400000	0.000000	773.170000	1499.000000	16.000000
ONEOFF_PURCHASES	0.000000	0.000000	773.170000	1499.000000	16.000000
INSTALLMENTS_PURCHASES	95.400000	0.000000	0.000000	0.000000	0.000000
CASH_ADVANCE	0.000000	6442.945483	0.000000	205.788017	0.000000
PURCHASES_FREQUENCY	0.166667	0.000000	1.000000	0.083333	0.083333
ONEOFF_PURCHASES_FREQUENCY	0.000000	0.000000	1.000000	0.083333	0.083333
PURCHASES_INSTALLMENTS_FREQUENCY	0.083333	0.000000	0.000000	0.000000	0.000000
CASH_ADVANCE_FREQUENCY	0.000000	0.250000	0.000000	0.083333	0.000000
CASH_ADVANCE_TRX	0.000000	4.000000	0.000000	1.000000	0.000000
PURCHASES_TRX	2.000000	0.000000	12.000000	1.000000	1.000000
CREDIT_LIMIT	1000.000000	7000.000000	7500.000000	7500.000000	1200.000000
PAYMENTS	201.802084	4103.032597	622.066742	0.000000	678.334763
MINIMUM_PAYMENTS	139.509787	1072.340217	627.284787	0.019163	244.791237
PRC_FULL_PAYMENT	0.000000	0.222222	0.000000	0.000000	0.000000
TENURE	12.000000	12.000000	12.000000	12.000000	12.000000

Figure 1: Few Rows of Data

Data Cleaning

In the process of working with our bank customer data, it's essential to ensure that the information is accurate and tidy. Data cleaning involves various steps such as:

- Handling missing values: Checking for any missing or incomplete information and deciding how to deal with it, whether by filling in the missing data or removing those entries.
- Removing duplicates: Identifying and eliminating any duplicated records in the dataset.
- Standardizing data formats: Ensuring consistency in data formats across different columns, such as date formats or numerical representations.
- Correcting inconsistencies: Checking for and rectifying any inconsistencies or errors in the data entries.

Handling Missing Values

In our dataset, we've observed that there are some missing values in two specific columns:

- There are 313 missing values out of 8950 observations in the **MINIMUM_PAYMENTS** column.
- Additionally, there's only 1 missing value in the **CREDIT_LIMIT** column.

Here are the steps I propose to handle these missing values:

For the MINIMUM_PAYMENTS column: The data in the MINIMUM_PAYMENTS column is skewed towards lower values, and we have some missing values. One way to address this is by considering the minimum non-zero value in the column and filling the missing values with this minimum value.

For the CREDIT_LIMIT column: There's only one missing value in the CREDIT_LIMIT column out of 8950 observations. We can easily remove this particular observation from the dataset since it's just a single entry.

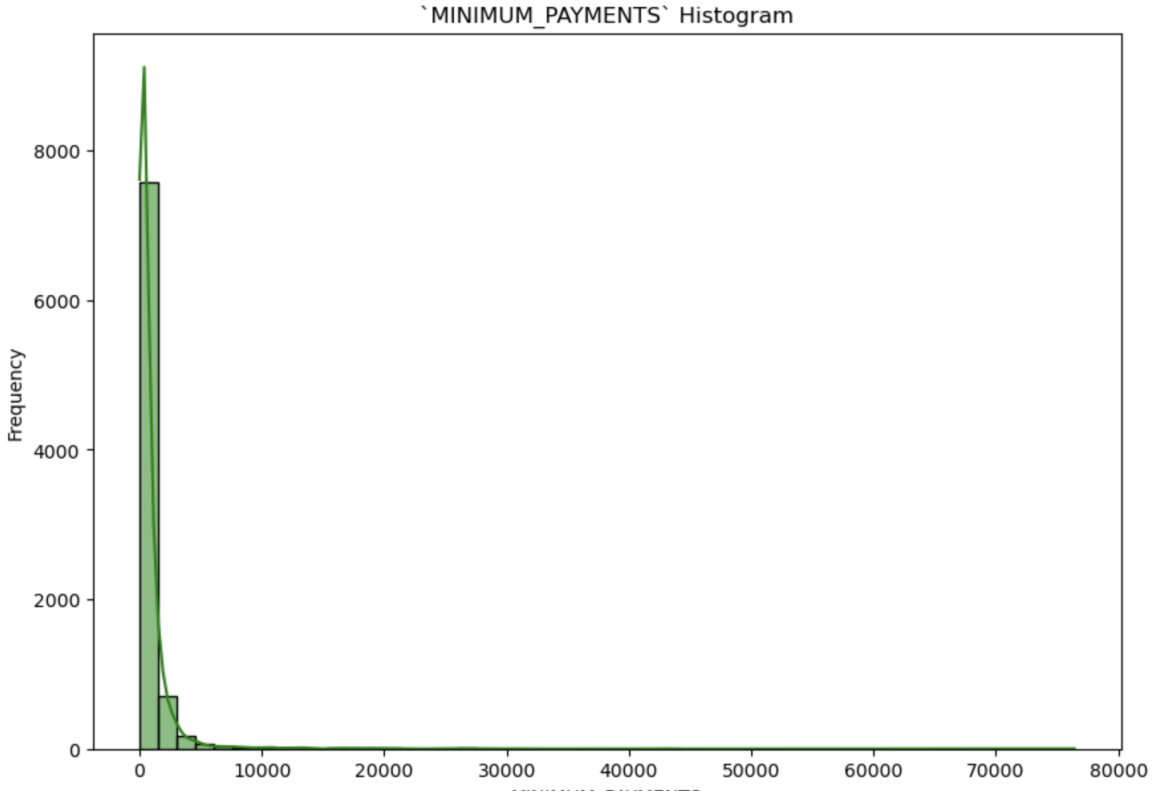

Figure 2: MINIMUM_PAYMENTS Histogram

Removing duplicates:

There are no duplicate values present in the dataset. Each observation is unique, ensuring that our data doesn't contain any repeated information.

Handling Typos:

All columns in our dataset contain numerical values, indicating a consistent data type throughout. Moreover, after thorough examination, there are no apparent typos or inconsistencies in the data entries, ensuring its accuracy and reliability.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) unveils patterns, outliers, and insights within datasets, aiding in understanding data structure, identifying relationships between variables, and informing modeling decisions. It helps detect errors, comprehend data distributions, and guides feature selection, improving the efficiency and accuracy of subsequent analyses or machine learning models. Ultimately, EDA empowers data-driven decisions by revealing crucial aspects and nuances hidden within the data.

During the Exploratory Data Analysis (EDA) phase, we delved into the dataset to understand the relationships between different columns.

Correlation Analysis: Upon drawing the correlation matrix, we identified some columns that show a strong relationship with each other:

- **PURCHASES** and **ONEOFF_PURCHASES** have a high correlation of 0.92, suggesting a strong positive relationship between these two columns. This indicates that customers who make more general purchases (PURCHASES) also tend to make substantial single purchases (ONEOFF_PURCHASES).
- **PURCHASES_INSTALLMENTS_FREQUENCY** and **PURCHASES_FREQUENCY** exhibit a correlation coefficient of 0.86. This strong positive correlation implies that customers who frequently make purchases in installments also tend to have a high frequency of overall purchases.
- Lastly, **CASH_ADVANCE_TRX** and **CASH_ADVANCE_FREQUENCY** show a correlation of 0.8, indicating a substantial positive relationship between the number of cash advance transactions and the frequency of cash advances taken by customers.

Understanding these strong correlations can provide insights into how different aspects of customer behavior are interconnected within the dataset.

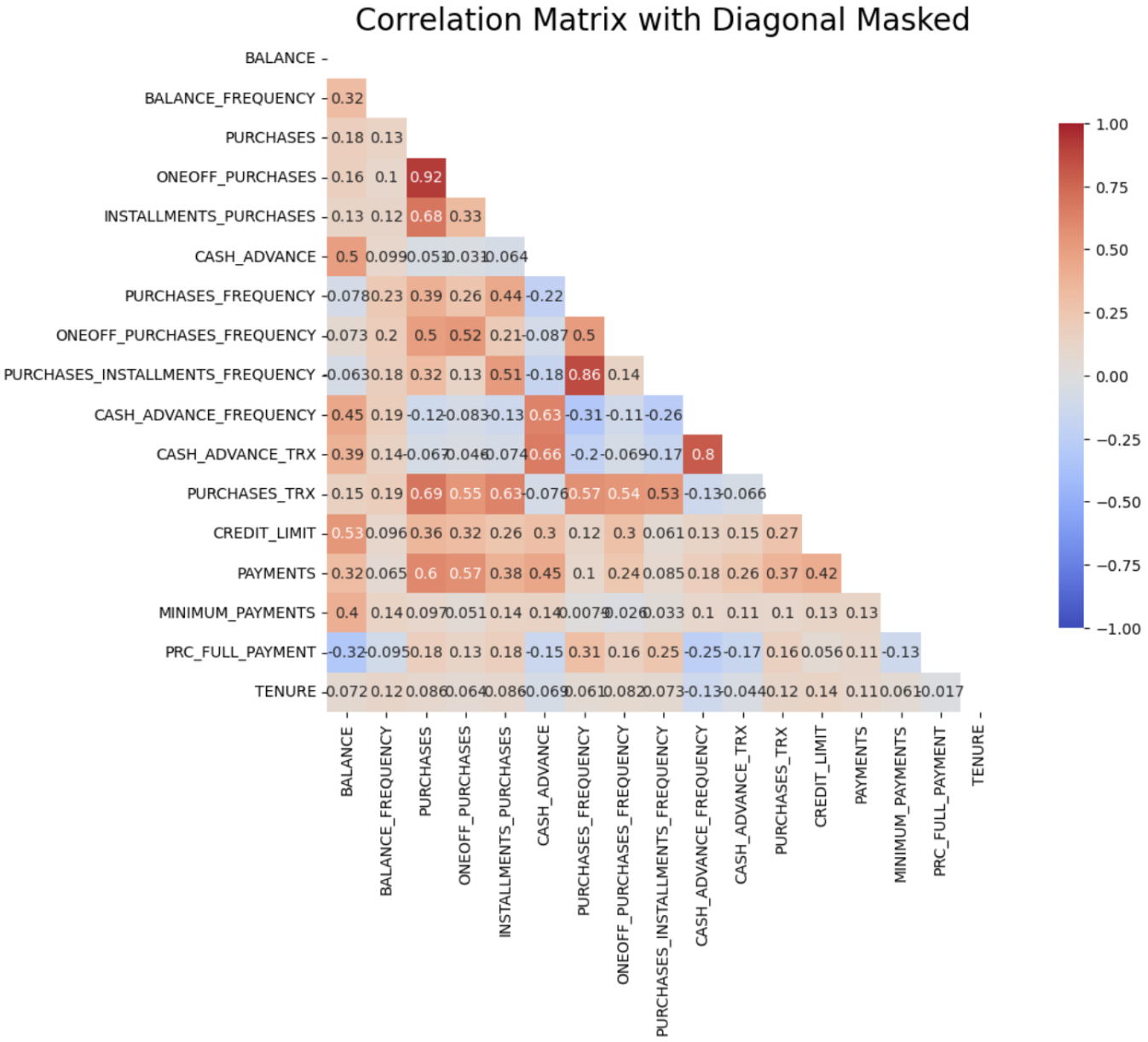

Figure 3: Correlation Matrix

Analysis

In this section, we'll dive deeper into analyzing the data to gain valuable insights and understanding.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique used to simplify complex datasets. It helps in identifying patterns and reducing the number of variables while preserving the key information within the data.

In our context, applying PCA to our bank customer dataset can assist in:

- Reducing the dimensionality of our dataset, making it easier to visualize and interpret.
- Identifying the most significant features that contribute to the variance in customer behavior.
- Uncovering underlying patterns or structures within the data that might not be apparent in the original dataset.

By implementing PCA, we aim to streamline our dataset and extract essential information, enabling us to better understand the key factors influencing customer behavior and segmentation.

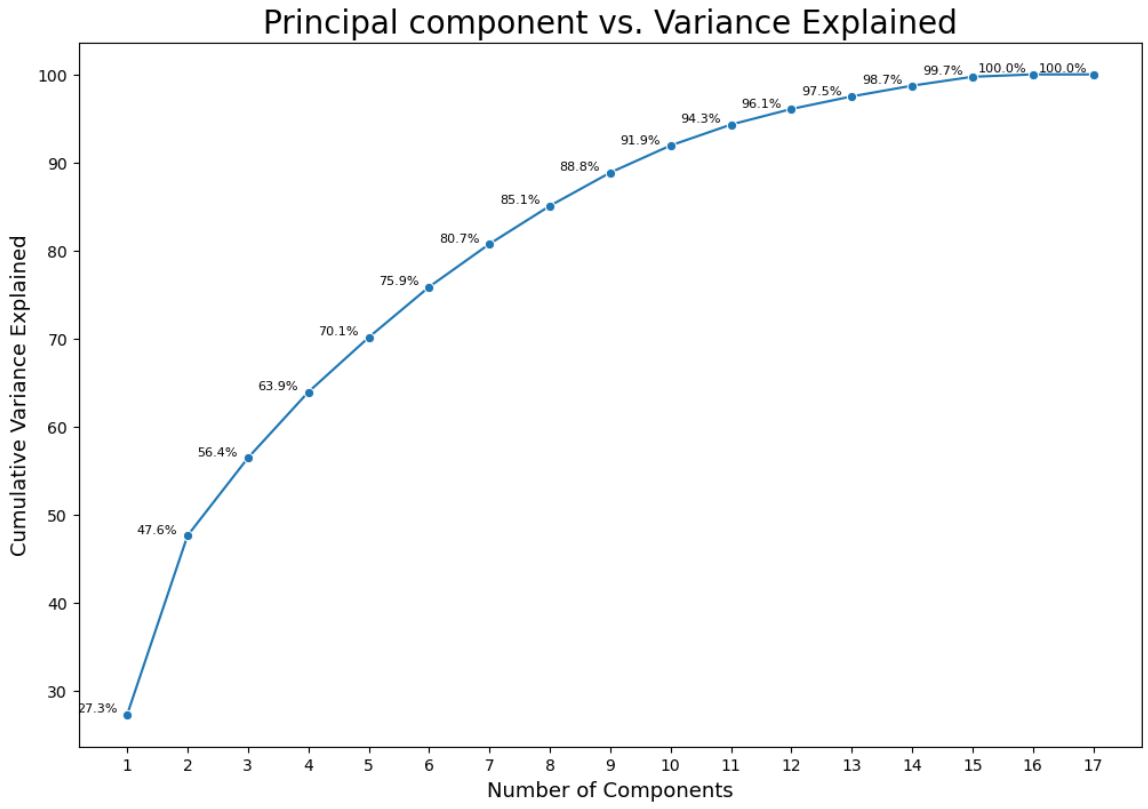

Figure 4: Principal component vs. Variance Explained

Insight:

This plot represents the relationship between the cumulative variance explained (right-hand side of the plot) and the number of principal components used (top-left corner of the plot). The y-axis represents the percentage of variance explained, and the x-axis represents the number of principal components.

As the number of components increases, the percentage of variance explained by each additional component also increases. This means that each subsequent component is able to capture a greater amount of variance in the data, potentially explaining even more variance than the preceding components.

In the context of Principal Component Analysis (PCA), this plot provides insights into how many components are necessary to capture a certain percentage of the total variance in the data. The intersection of the blue line and the diagonal line at (8, 85.1%) represents a commonly used threshold for selecting the number of components to retain.

Feature Contribution to Principal Components

This subsection deals with understanding how each feature or column in our dataset contributes to the principal components obtained through PCA.

In PCA, the original features are transformed into a new set of variables called principal components. These components are combinations of the original features and are ordered by their importance in explaining the variance in the data.

Here, we aim to explore and interpret which original features have a higher impact or contribute more significantly to these principal components. Understanding the feature contributions helps us grasp the key factors driving the variations observed in the dataset.

By analyzing the contribution of features to principal components, we gain insights into which aspects of customer behavior or attributes are more influential in shaping the overall patterns and variability in our dataset.

Component Loadings of PCA Variables

The component loadings in PCA represent the correlation between the original variables and the principal components. These loadings help in understanding how much each original variable contributes to each principal component. A plot of component loadings visually displays these relationships.

Here's an explanation of a plot showing component loadings:

Imagine a scatter plot where each point represents a variable (feature) from your dataset. The x-axis displays the principal components (PC1, PC2, etc.), while the y-axis shows the correlation or loading value of each variable with that principal component.

- Direction: The direction of the points concerning the components shows how positively or negatively they correlate with that principal component.

- Distance from the Origin: The farther the points are from the origin, the stronger their correlation or contribution to that principal component. Variables closer to the origin have less influence on that particular principal component.

- Clusters or Patterns: Groups of variables closer together might indicate similar behavior or relationships in the data. For example, if a variable points strongly in the direction of PC1 and is far from the origin, it has a high correlation with PC1 and contributes significantly to explaining the variance captured by PC1.

Analyzing this plot helps in identifying which variables contribute the most to each principal component and how these variables are related to each other in terms of their influence on the overall dataset variability.

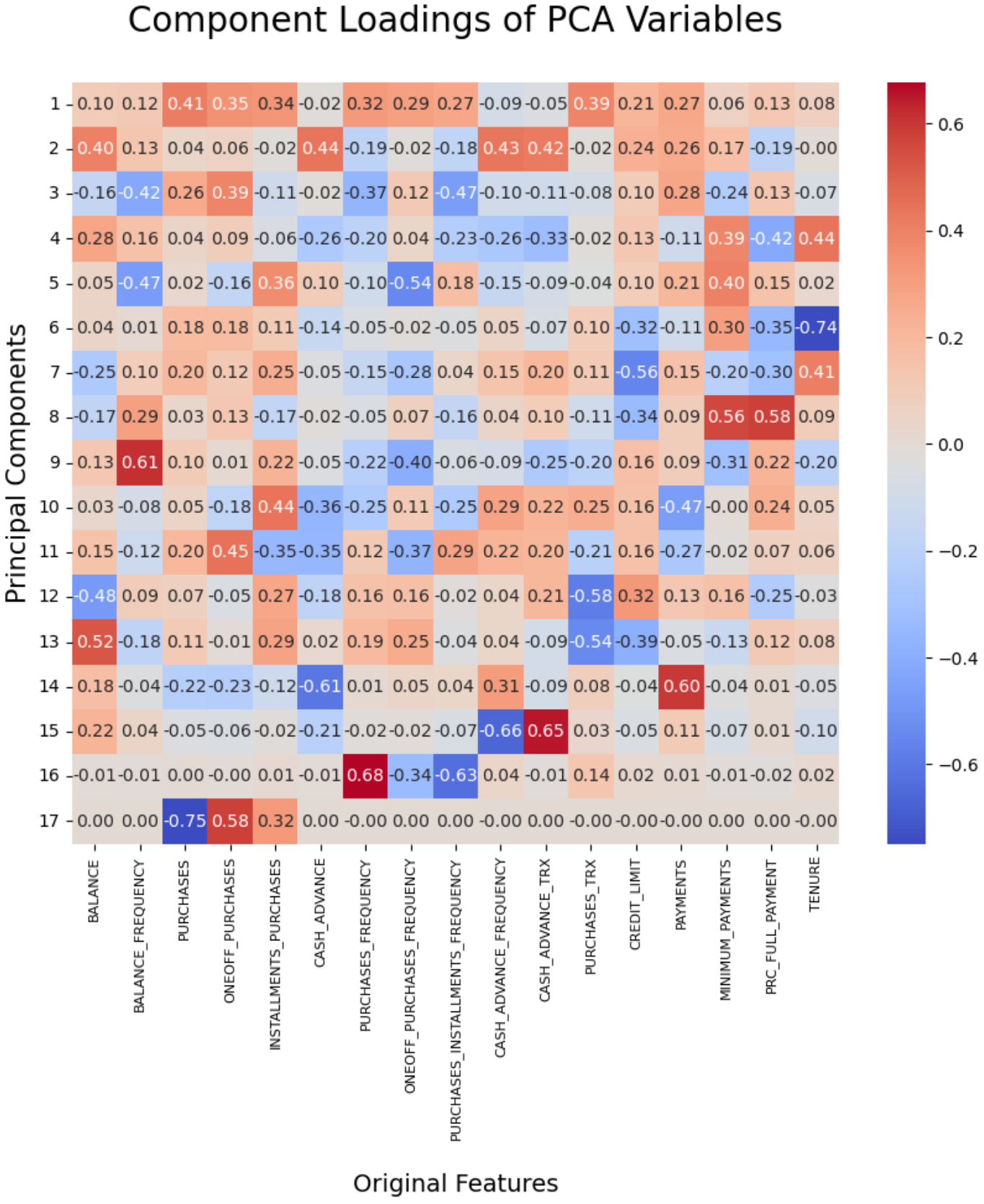

Figure 5: Component Loadings of PCA Variables

PCA Visualization in 2 Dimensions

In the given PCA2 plot, each point represents a sample from the original dataset. The two axes (PCA1 and PCA2) represent two linear combinations of the original features, weighted by their explained variance. The axes are sorted by their explained variance, with the highest variance explained by the first principal component.

In the case of PCA2, the second principal component (PCA2) explains a higher amount of variance compared to the first principal component (PCA1). This suggests that there is a linear relationship between the features that the second principal component captures.

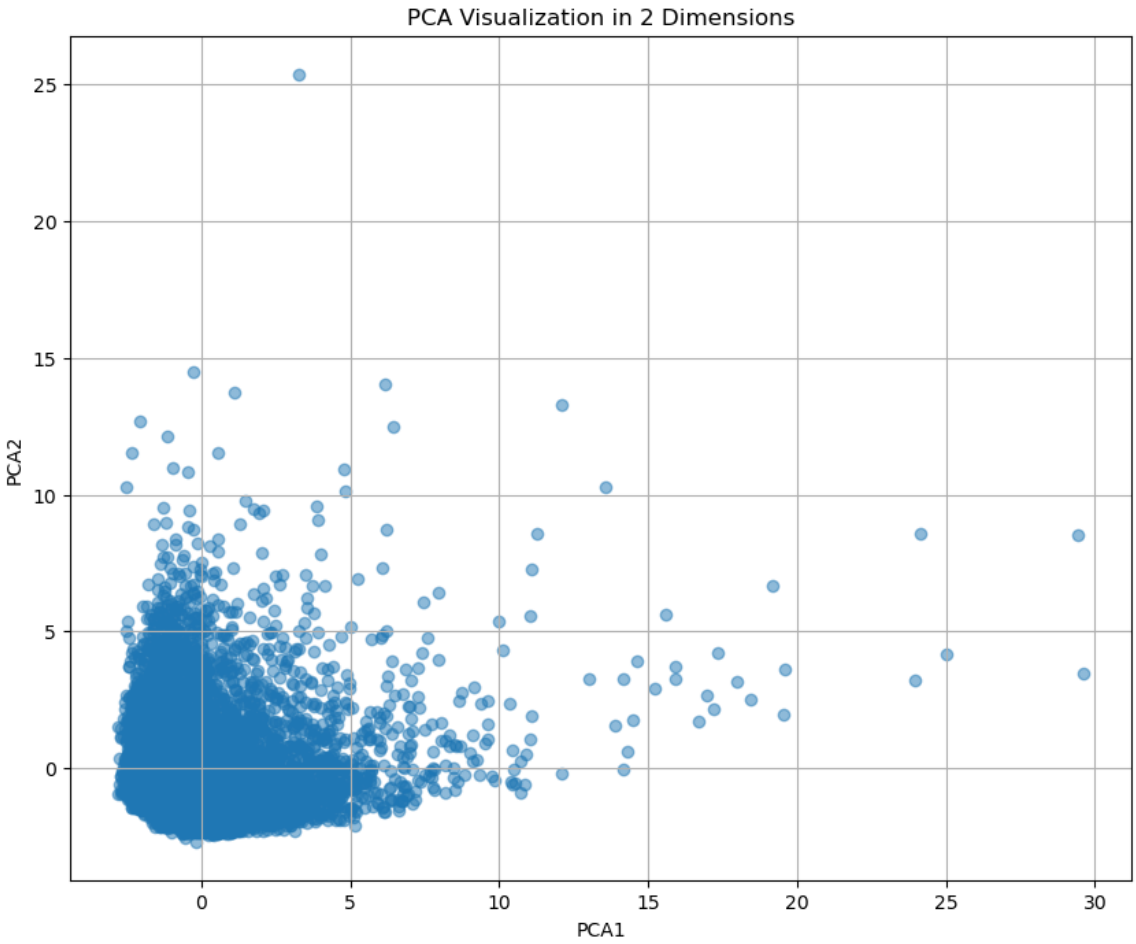

Figure 6: PCA Visualization in 2 Dimensions

Factor Analysis

Factor Analysis is a statistical method used to uncover underlying latent variables (factors) from observed variables. It aims to identify patterns in the relationships among observed variables and explain the variance by reducing the dimensions into a smaller set of unobserved variables.

In this section, we'll delve into Factor Analysis, exploring its application in our dataset. Factor Analysis helps in understanding the interdependencies among variables, discovering hidden patterns, and simplifying the complexity of the data.

$$X_{ij} = \mu_i + \sum_{k=1}^{p} l_{ik} F_{kj} + \varepsilon_{ij}$$

Where:

- X_{ij} represents the observed value for variable X at the ith row and jth column.
- μ_i denotes the mean or intercept for the ith variable.
- l_{ik} denotes the loading of the ith variable on the kth factor.
- F_{kj} represents the value of the kth factor for the jth observation.
- ε_{ij} represents the error term for the ith variable at the jth observation.

Factor Loading Heatmap

A Factor Loading Heatmap visualizes the factor loadings obtained from Factor Analysis. Factor loadings represent the correlations between the observed variables and the underlying factors extracted through Factor Analysis.

In this heatmap, each cell displays the strength and direction of the correlation between an observed variable and a factor. Higher values indicate stronger correlations, indicating that the variable contributes more to that particular factor.

By examining the vertical slices along the Y-axis, we can identify influential latent variables (factors) for specific features. For instance, Factor 1 significantly impacts BALANCE and BALANCE_FREQUENCY, while Factor 2 influences PAYMENTS and PRC_FULL_PAYMENT.

The color gradient denotes the magnitude of loading, with darker blue cells signifying higher absolute values of influence. Notably, Factor 1 exerts a strong influence on PURCHASES FREQUENCY and CREDIT_LIMIT.

Positive or negative loadings indicate the direction and strength of correlation between features and latent factors. For instance, Factor 2 positively correlates with ONEOFF_PURCHASES FREQUENCY, while Factor 6 exhibits a negative correlation with TENURE.

Examining the correlation matrix in the heatmap's bottom-right unveils interrelationships between factors. For example, factors 1 and 2 exhibit a robust correlation of 0.91, suggesting a high degree of interrelation between these two factors.

Ultimately, this heatmap offers a nuanced understanding of the dataset's underlying structure, unraveling intricate patterns within the relationships among factors and features. Deciphering this matrix enables us to grasp the complex web of connections within the data.

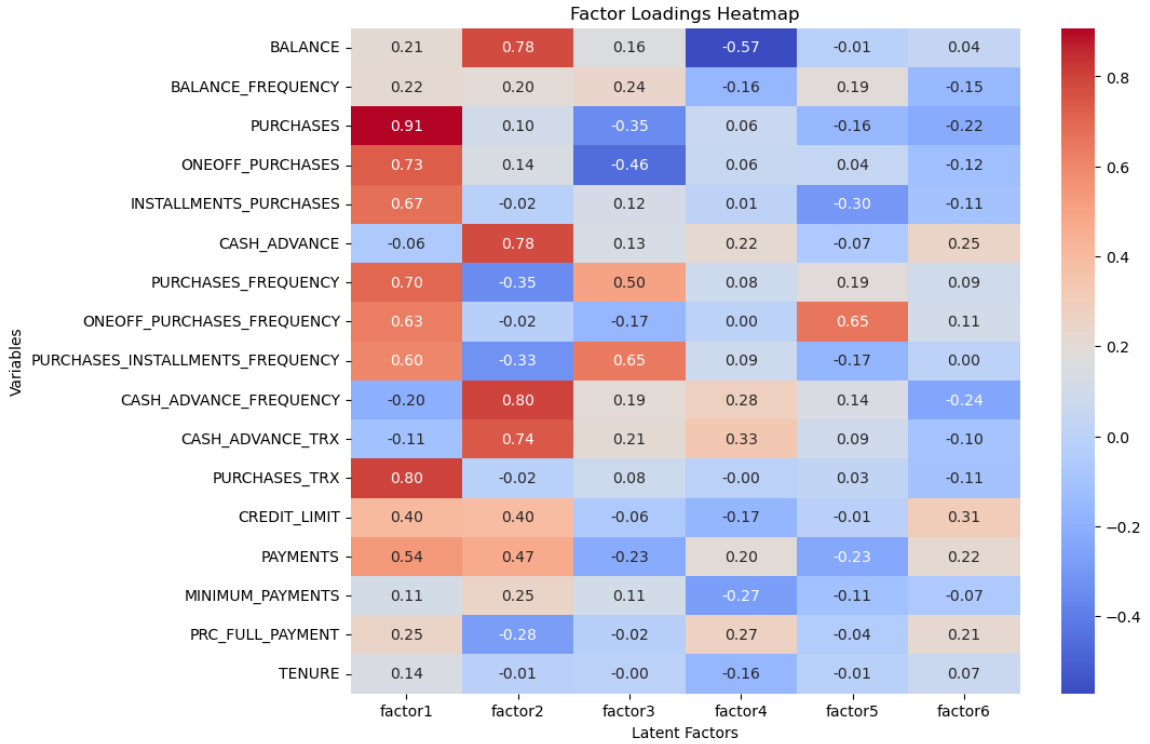

Figure 7: Factor Loading Heatmap

Factor Rotation and Factor Loadings Heatmap

Factor Rotation is a technique used in Factor Analysis to enhance the interpretability of factors by rotating them to a simpler or more interpretable structure. It aims to achieve clearer patterns by altering the orientation of factors without changing their correlations with observed variables.

In this section, we'll explore the concept of Factor Rotation and its impact on the interpretation of factors obtained from Factor Analysis.

Additionally, we will present a Factor Loadings Heatmap, which visualizes the correlations between observed variables and the rotated factors. This heatmap provides insights into the relationships between variables and the rotated factors, aiding in the identification of key variables influencing each factor.

The factor loading heatmap showcases correlations between factors and original variables. However, when two factors heavily load onto one variable, interpretation becomes challenging. This complication emphasizes the need for factor rotation. By emphasizing inter-factor correlations over their individual links with original variables, factor rotation minimizes such complexities, ensuring a clearer and more robust interpretation, especially in cases where multiple factors strongly impact the same variable.

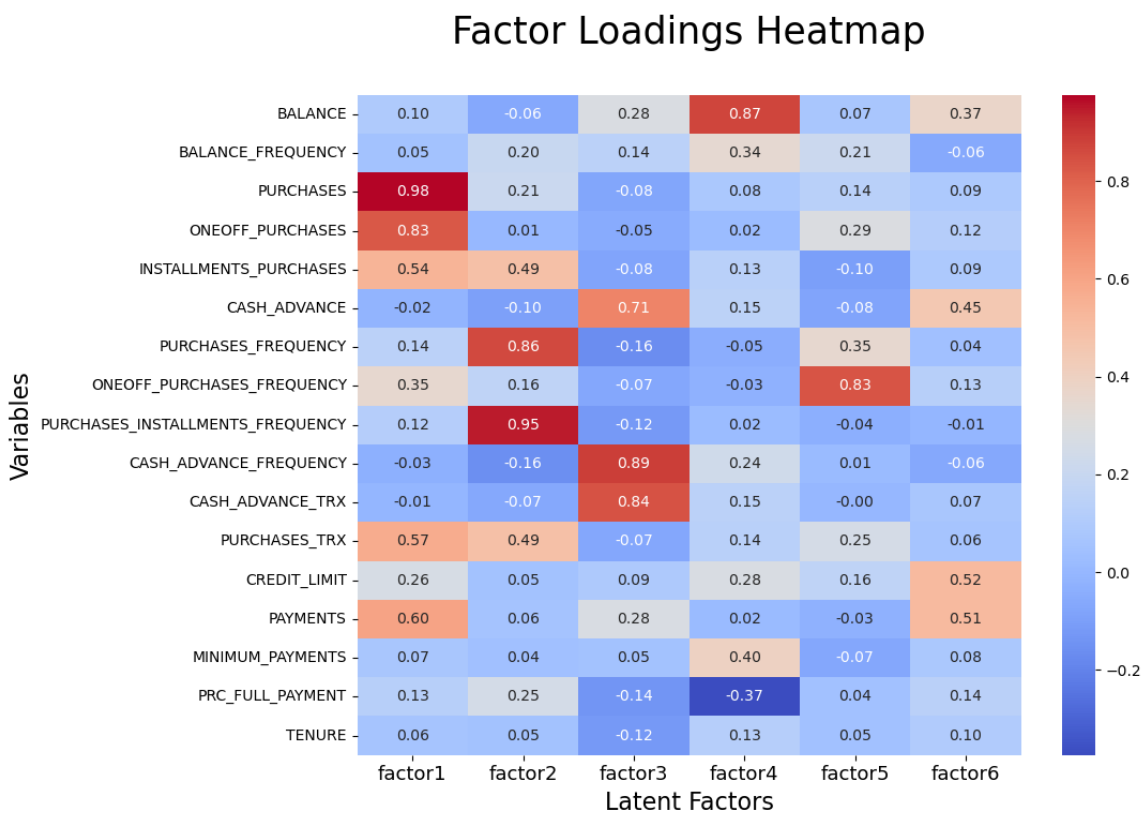

Figure 8: Factor Loadings Heatmap

Interpretation of Rotated Factors

Factor 1: Captures diverse purchasing behaviors across different channels and methods

- **Variables:** 'PURCHASE,' 'ONLINE PURCHASES,' 'INSTALLMENTS PURCHASES'
- **Interpretation:** Factor 1 represents the diverse purchasing behaviors exhibited by customers across various channels and methods. It encompasses purchases made online, through installments, and other related modes, indicating the varied purchasing patterns among customers.

Factor 2: Types of Purchases

- **Variables:** 'PURCHASES,' 'ONOFF PURCHASES,' 'PURCHASES TRX'
- **Interpretation:** Factor 2 signifies distinct categories or patterns in customers' purchasing habits. It encapsulates various types of purchases made by customers, reflecting specific categories or modes in which they engage while making transactions.

Factor 3: Cash Advance

- **Variables:** 'CASH ADVANCE,' 'CASH ADVANCE FREQUENCY,' 'CASH ADVANCE TRX'
- **Interpretation:** Factor 3 represents preferences in payment methods, particularly related to cash advance transactions. It signifies how frequently customers engage in cash advance transactions and their behaviors concerning this payment method.

Clustering

Clustering is a machine learning technique used to partition a dataset into groups or clusters of similar data points based on certain characteristics or features.

KMeans

KMeans is a popular clustering algorithm that aims to partition data into K clusters. It works by iteratively assigning data points to the nearest cluster centroid and updating the centroids to minimize the total within-cluster variance.

In this subsection, we'll delve into the KMeans clustering algorithm, exploring its working principles, application in clustering data, and the process of determining the optimal number of clusters (K) using techniques such as the Elbow Method or Silhouette Score.

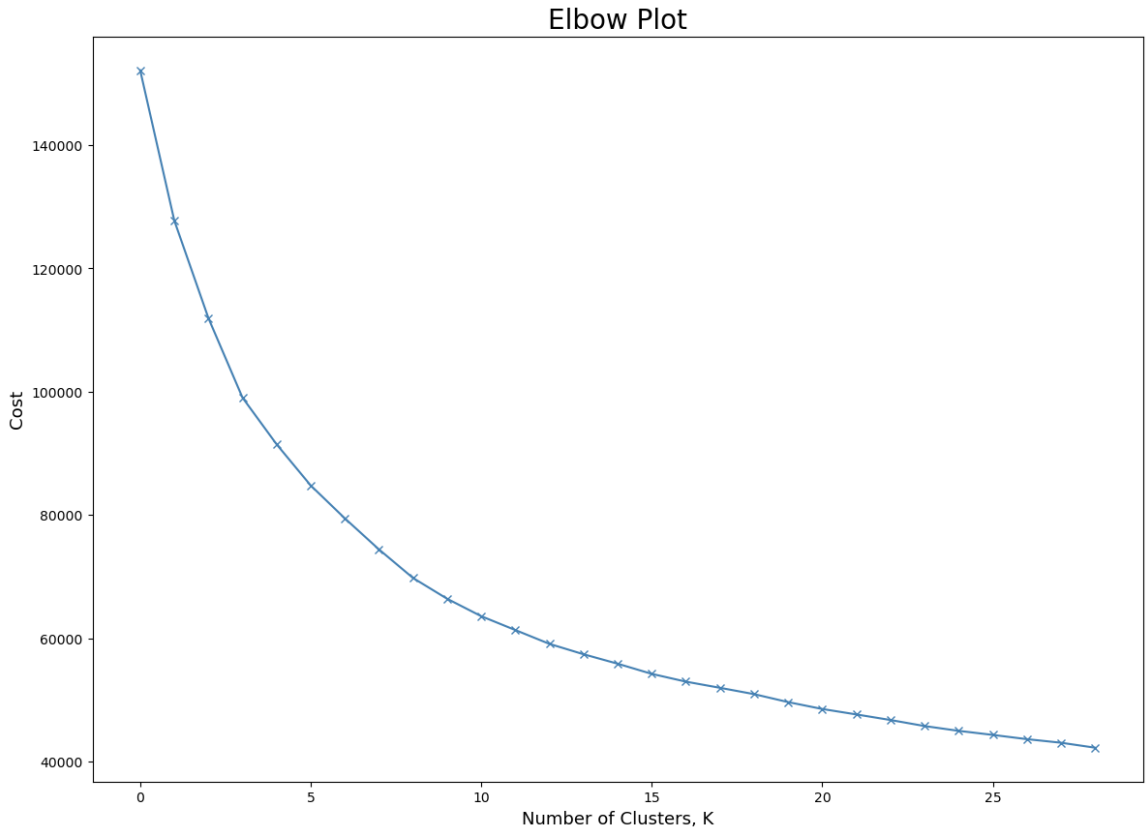

Figure 9: Elbow Plot

KMeans Clustering Results

The KMeans clustering algorithm was applied to the dataset with the choice of 4 clusters (k=4). Here are the obtained results:

- Sum of Squared Distances to Closest Cluster Center (Inertia): 99048.0
- Number of Iterations Run: 16

- Number of Features Seen During Fit: 17

The ”Sum of Squared Distances to Closest Cluster Center,” also known as inertia, is a measure indicating how internally coherent the clusters are. A lower inertia generally signifies that the clusters are more compact and well-separated.

The KMeans algorithm required a total of 16 iterations to converge and determine the optimal cluster centroids. Additionally, during the fitting process, it observed 17 features in the dataset.

These results provide insights into the performance and characteristics of the KMeans clustering applied with 4 clusters to the scaled dataset.

Scatterplot of Clusters with K=4

A scatterplot was generated to visualize the clusters resulting from the KMeans algorithm with K=4 applied to the principal components PCA1 and PCA2.

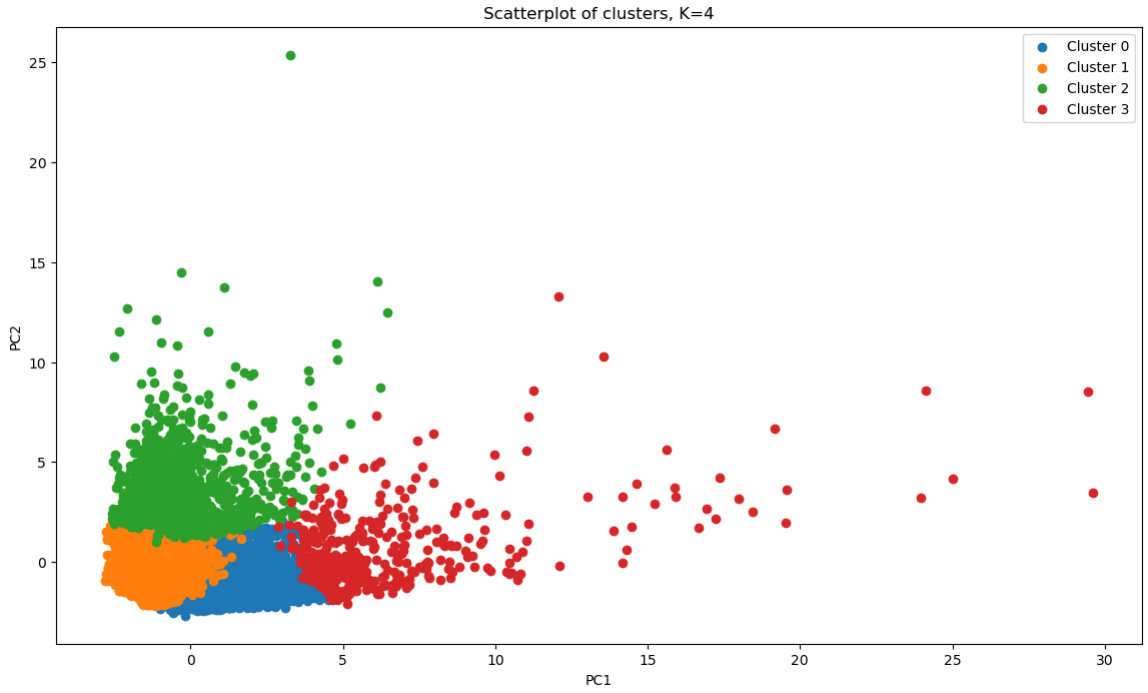

Figure 10: Scatterplot of clusters, K=4

The scatterplot displays the distribution of data points across the principal components PCA1 and PCA2, highlighting the identified clusters. Each data point is color-coded according to its assigned cluster label (Cluster 0, Cluster 1, Cluster 2, and Cluster 3).

The visualization aims to provide insights into the separation or overlap of clusters in the reduced dimensional space of PCA1 and PCA2, facilitating the examination of distinct groupings or patterns among the data points.

Hierarchical Clustering

Hierarchical Clustering is a method used to cluster data by recursively merging or dividing clusters based on the similarity between data points. It creates a tree-like structure known as a dendrogram, illustrating the

arrangement of clusters at different levels of similarity.

Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering is a bottom-up approach where each data point starts in its own cluster, and pairs of clusters are iteratively merged based on their similarity until only one cluster remains.

The process involves calculating distances or similarities between clusters or data points and merging the closest ones until the desired number of clusters is achieved or a specific criterion, such as a distance threshold, is met.

Complete Linkage

Complete Linkage is a method used in hierarchical clustering to measure the distance or dissimilarity between clusters. It computes the distance between two clusters based on the maximum distance between their individual data points.

The distance between two clusters is defined as the maximum distance between any pair of data points, one from each cluster. This method considers the most dissimilar or farthest points between clusters when determining their distance.

Mathematically, the distance $d(C_i, C_j)$ between clusters C_i and C_j using Complete Linkage can be expressed as:

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} \text{distance}(x, y)$$

where x and y are individual data points in clusters C_i and C_j respectively, and distance(x, y) represents the distance metric used to measure the dissimilarity between data points.

Complete Linkage tends to create compact clusters with approximately equal diameters and may be sensitive to outliers or noise due to its consideration of the maximum pairwise distances between clusters.

Cluster Label	No of Observation
0	4931
1	315
2	3680
3	23

Figure 11: This table shows how many observations are assigned to clusters 01, 2, and 3, respectively.

CUST_ID	C10001	C10002	C10003	C10004	C10005	C10006	C10007	C10008
BALANCE	40.900749	3202.467416	2495.148862	1666.670542	817.714335	1809.828751	627.260806	1823.652743
BALANCE_FREQUENCY	0.818182	0.909091	1.000000	0.636364	1.000000	1.000000	1.000000	1.000000
PURCHASES	95.400000	0.000000	773.170000	1499.000000	16.000000	1333.280000	7091.010000	436.200000
ONEOFF_PURCHASES	0.000000	0.000000	773.170000	1499.000000	16.000000	0.000000	6402.630000	0.000000
INSTALLMENTS_PURCHASES	95.400000	0.000000	0.000000	0.000000	0.000000	1333.280000	688.380000	436.200000
CASH_ADVANCE	0.000000	6442.945483	0.000000	205.788017	0.000000	0.000000	0.000000	0.000000
PURCHASES_FREQUENCY	0.166667	0.000000	1.000000	0.083333	0.083333	0.666667	1.000000	1.000000
ONEOFF_PURCHASES_FREQUENCY	0.000000	0.000000	1.000000	0.083333	0.083333	0.000000	1.000000	0.000000
PURCHASES_INSTALLMENTS_FREQUENCY	0.083333	0.000000	0.000000	0.000000	0.000000	0.583333	1.000000	1.000000
CASH_ADVANCE_FREQUENCY	0.000000	0.250000	0.000000	0.083333	0.000000	0.000000	0.000000	0.000000
CASH_ADVANCE_TRX	0.000000	4.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
PURCHASES_TRX	2.000000	0.000000	12.000000	1.000000	1.000000	8.000000	64.000000	12.000000
CREDIT_LIMIT	1000.000000	7000.000000	7500.000000	7500.000000	1200.000000	1800.000000	13500.000000	2300.000000
PAYMENTS	201.802084	4103.032597	622.066742	0.000000	678.334763	1400.057770	6354.314328	679.065082
MINIMUM_PAYMENTS	139.509787	1072.340217	627.284787	0.019163	244.791237	2407.246035	198.065894	532.033990
PRC_FULL_PAYMENT	0.000000	0.222222	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
TENURE	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000
cluster_label	0.000000	2.000000	2.000000	2.000000	0.000000	0.000000	1.000000	0.000000

Figure 12: Scatterplot of clusters, K=4

Insight

This dataset includes an additional column named 'Cluster Label' which indicates the cluster assignment for each customer. This label specifies the particular cluster to which each customer has been assigned based on the clustering analysis performed on the dataset.

Understanding these cluster labels is crucial as they enable segmentation of customers into distinct groups based on their behavioral patterns. This segmentation allows for tailored marketing strategies, personalized services, and targeted approaches to meet the specific needs and preferences of customers in each cluster, ultimately enhancing customer satisfaction and engagement.

Dendrogram Visualization

The result of Hierarchical Clustering is often visualized using a dendrogram, a tree-like diagram that demonstrates the merging process and displays the hierarchical relationships among data points or clusters.

The vertical lines in the dendrogram represent clusters, and the height at which two clusters merge indicates their dissimilarity. By observing the dendrogram, one can determine the number of clusters by identifying significant jumps in height.

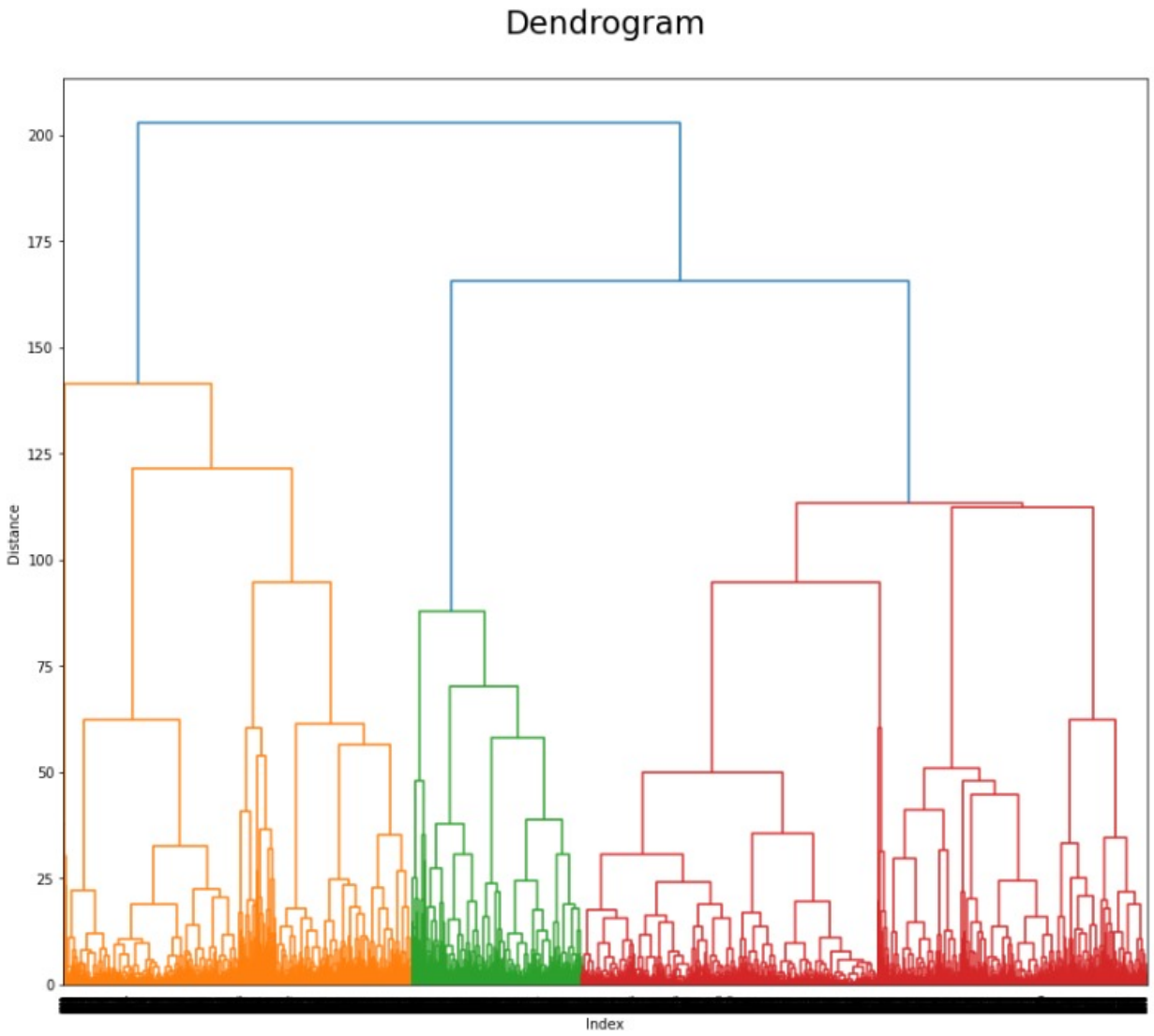

Figure 13: Complete Linkage