Report on Customer Segmentation through Clustering

Introduction

Customer segmentation is an important step in understanding different customer groups and developing individualized marketing strategies. This report utilizes clustering techniques to segment customers based on their transaction patterns and demographic characteristics using two datasets: Customers.csv and Transactions.csv. Clustering allows us to find meaningful segments for making data-driven business decisions.

It applied three clustering algorithms, namely K-Means, Agglomerative Clustering, and Gaussian Mixture Models, which have the latter identified to be more suitable based on ease of implementation, computationally efficient, and produces clear well-defined clusters. The work analyses different metrics: DB Index, silhouette score, to determine the quality and validation of clustering. In summary, six segments are developed for distinct customers that have implications for actionable engagement with them.

Data Preprocessing

Data Cleaning and Transformation:

To ensure that the datasets were ready for clustering, comprehensive preprocessing steps were performed. There were no missing values, and no duplicate values existed in either dataset. Key date fields such as TransactionDate and SignupDate were converted to datetime objects for feature engineering purposes. The two datasets were merged using the CustomerID column to create a combined dataset of 1,000 rows.

Feature Engineering:

Several features were engineered to enhance the dataset:

Recency: This is calculated as the number of days since the last transaction for every customer. The range was between 124 days and 281 days, and this gave some information about customer activity.

Tenure: It is measured in terms of the number of days since a customer signed up, and the range was between 115 days and 1,019 days.

Frequency: This is defined as the total number of transactions per customer, and the customers made as many as 7 transactions.

TotalValue and Quantity: Indicating currency value and number of items bought, respectively.

	Quantity	TotalValue	Price	Recency	Tenure	Frequency	Region_Encoded
0	1	300.68	300.68	124	756	4	1
1	1	300.68	300.68	214	115	4	0
2	1	300.68	300.68	247	268	6	1
3	2	601.36	300.68	276	261	7	3
4	3	902.04	300.68	281	1019	4	1

Scaling and Encoding:

All features must contribute the same to clustering. Thus, the numerical features TotalValue, Quantity, Recency, Tenure, and Frequency have all been scaled from 0 to 1 using MinMaxScaler. Also, categorical feature Region is encoded with Label Encoder for coherent implementation in the clustering algorithm.

Clustering Approach

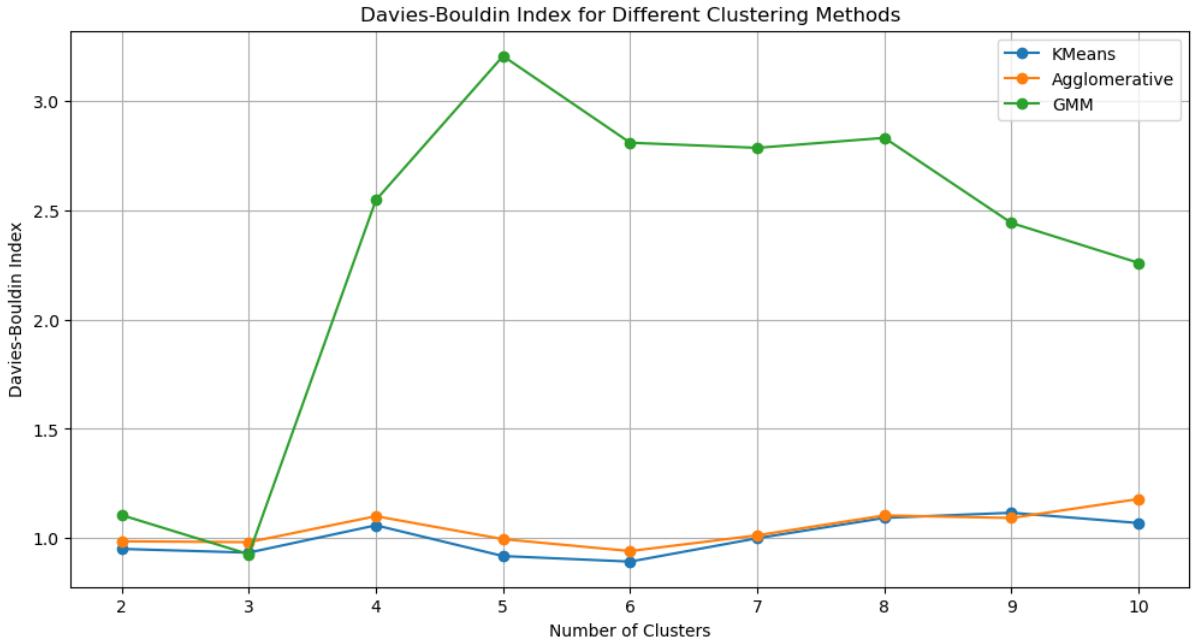
Algorithm Details:

Three algorithms were explored in the clustering process:

K-Means Clustering: This is a centroid-based algorithm that categorizes data into distinct groups. It was run with a number of clusters ranging from 2 to 10, with 6 clusters indicated as optimal.

Agglomerative Clustering: This is a hierarchical method that groups data points progressively by mergers. The optimal number of clusters for this method was also 6.

Gaussian Mixture Models (GMM): This is a probabilistic clustering approach that relies on the assumption that data points are generated from a mixture of Gaussian distributions that overlap. The approach yielded 3 clusters, but the resulting structure of the clusters showed greater overlap.

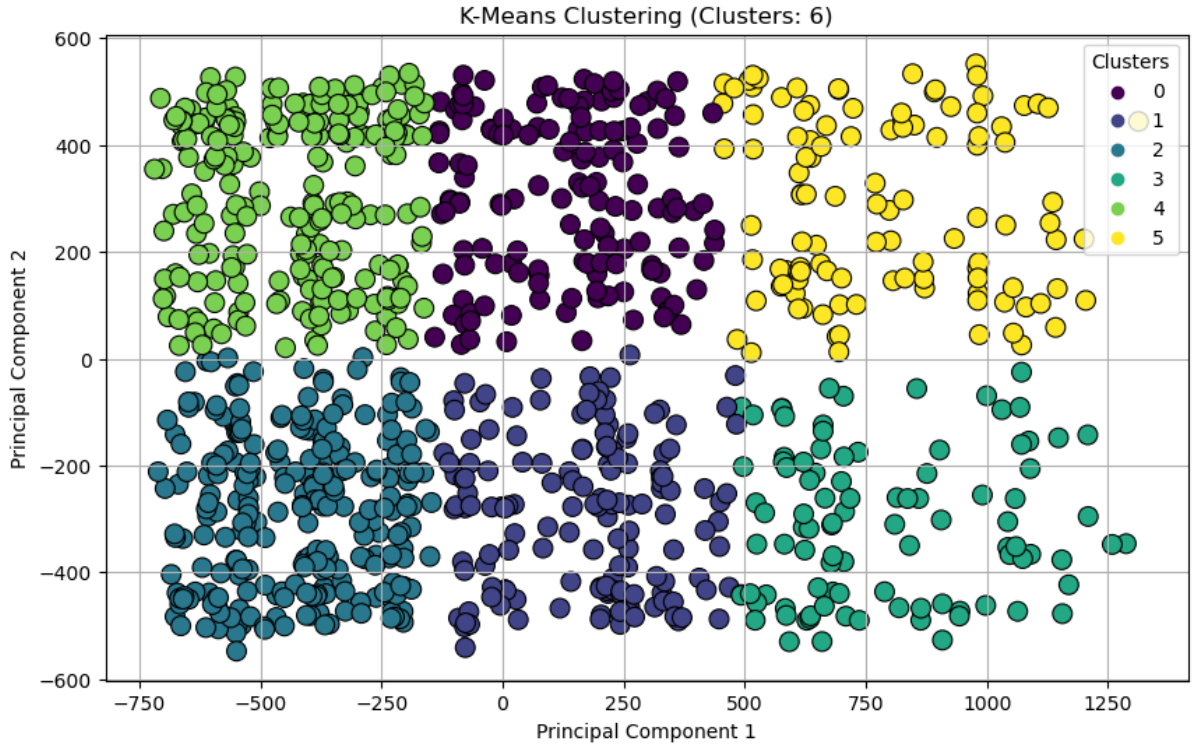


Optimal Number of Clusters

After verifying that K-Means performed better:

Silhouette Score: K-Means was found to have the highest silhouette score with 6 clusters, indicating that the clusters being formed are well-separated and compact.

Davies-Bouldin Index (DB Index): The DB Index for K-Means was 0.8919. It scored much better than Agglomerative Clustering, which had a DB Index of 0.9402 and GMM with a DB index of 0.9256.



Evaluation Metrics

The DB Index measures the compactness of clusters and their separation, with lower values indicating better clustering. K-Means achieved the best DB Index of 0.8919 with 6 clusters, outperforming Agglomerative Clustering (0.9402) and GMM (0.9256). Across all methods, increasing the number of clusters beyond 6 led to diminishing returns.