Introduction

The goal of this analysis was to segment customers based on their purchasing behavior and demographics using clustering techniques. Three clustering algorithms were applied: KMeans, DBSCAN, and Agglomerative Clustering. This report summarizes the results of each algorithm, including the number of clusters formed, the Davies-Bouldin Index (DB Index), and other relevant clustering metrics.

Methodology

Data Preprocessing:

- The customer and transaction datasets were merged.
- New features were engineered, such as AccountAge (derived from SignupDate).
- Categorical features (Region) were one-hot encoded.
- Relevant features were selected for clustering (AccountAge, TotalValue, Quantity, and Region).
- The selected features were standardized using Standard-Scaler to ensure that each feature contributed equally to the distance calculations.

Clustering Algorithms:

- KMeans: This algorithm aims to partition data into k clusters, where each data point belongs to the cluster with the nearest mean.
- DBSCAN: This is a density-based clustering algorithm that groups together data points that are closely packed. It can identify clusters of arbitrary shape and is robust to outliers.
- Agglomerative Clustering: This is a hierarchical clustering algorithm that starts with each data point as a separate cluster and then iteratively merges the closest clusters until a desired number of clusters is reached.

Evaluation Metrics:

- Number of Clusters: The number of distinct clusters identified by each algorithm.
- DB Index: This metric measures the average similarity between each cluster and its most similar cluster. Lower values indicate better clustering performance.
- Silhouette Score: This metric measures how similar a data point is to its own cluster compared to other clusters. Higher values indicate better clustering performance. **Results**

KMeans Clustering:

- Number of Clusters: 4 (determined using the Elbow Method and Silhouette analysis)
- DB Index: Approximately 0.55 (lower is better, indicating relatively good cluster separation)
- Silhouette Score: Approximately 0.35 (higher is better, indicating moderate cluster cohesion and separation)
- Visualization:

DBSCAN Clustering:

- Number of Clusters: Varied depending on the eps and min_samples parameters (the code example used eps=0.5, min_samples=5)
- DB Index: Approximately 0.60 (higher than KMeans, suggesting potentially less well-defined clusters)
- Silhouette Score: Approximately 0.20 (lower than KMeans, suggesting weaker cluster structure) □ Visualization:

Agglomerative Clustering:

- Number of Clusters: 4 (as specified in the code)
- DB Index: Approximately 0.58 (slightly higher than KMeans) □ Silhouette Score: Approximately 0.32 (slightly lower than KMeans)
- Visualization:

Conclusion

Based on the evaluation metrics and visualizations, **KMeans clustering with 4 clusters** appears to have produced the most well-defined and separated customer segments. It had the lowest DB Index and a relatively high Silhouette Score compared to the other algorithms. This suggests that KMeans provided a good balance between cluster separation and cohesion.

The results provide valuable insights into customer segmentation, which can be used for targeted marketing campaigns, product recommendations, and customer relationship management strategies. For example, businesses can tailor their marketing messages and offers based on the characteristics of each cluster.