Customer Segmentation Using Clustering Techniques

---

1. Introduction

The goal of this project was to perform customer segmentation using clustering techniques to group customers based on their profile information and transaction history. We used K-Means, Agglomerative Clustering, and DBSCAN as the primary clustering algorithms. This segmentation is crucial for businesses to target customers effectively and personalize marketing strategies.

---

2. Clustering Algorithms

What is K-Means?

- K-Means is a centroid-based clustering algorithm that divides data points into a pre-specified number of clusters (K) based on feature similarity. It iteratively assigns each data point to the nearest centroid and updates the centroid position based on the mean of all assigned points. The algorithm minimizes the sum of squared distances between the data points and their respective centroids.
- **Key Steps:**
    1. Choose a value for K (number of clusters).
    2. Randomly initialize centroids.
    3. Assign each data point to the nearest centroid.
    4. Recalculate centroids based on assigned points.
    5. Repeat steps 3 and 4 until convergence.

What is Agglomerative Clustering?

- Agglomerative Clustering is a hierarchical clustering algorithm. It begins by treating each data point as an individual cluster and iteratively merges the closest clusters based on a defined distance metric (e.g., Euclidean distance) until the desired number of clusters is reached.
- **Key Steps:**
    1. Start with each data point as a separate cluster.
    2. Find the two closest clusters and merge them.
    3. Continue merging clusters based on proximity until a stopping condition (e.g., number of clusters) is met.

What is DBSCAN?

- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm. Unlike K-Means, it does not require specifying the number of clusters. It identifies clusters based on the density of data points, making it effective for detecting clusters of arbitrary shape and handling noise (outliers).
- **Key Parameters:**
    - eps: The maximum distance between two points for them to be considered as part of the same neighborhood.

- min_samples: The minimum number of points required to form a dense region (a cluster).

---

3. Data Preprocessing

The data used for clustering was derived from the following sources:

1. **Customers.csv:** Contains customer profile information.
2. **Transactions.csv:** Contains transaction records linking customers to products.
3. **Products.csv:** Contains product information.

The following preprocessing steps were applied:

- Merging the datasets on CustomerID and ProductID.
- Feature Engineering: Created features such as the total spend per customer, number of transactions, average transaction value, etc.
- Data Scaling: Standardized the features using StandardScaler to ensure all features contributed equally to the clustering algorithms.

---

4. Experiment Results and Evaluation

We evaluated three clustering algorithms with different configurations and reported the following metrics:

K-Means Clustering Results:

- **Number of Clusters: 9**
  - **DB Index: 0.8987 (Lower is better, indicating well-separated clusters)**
  - **Silhouette Score: 0.3931 (Higher is better, indicating good clustering quality)**

Agglomerative Clustering Results:

- **Number of Clusters: 10**
  - **DB Index: 0.8902 (Indicating well-separated clusters)**
  - **Silhouette Score: 0.3820 (Good separation and compactness of clusters)**

DBSCAN Clustering Results:

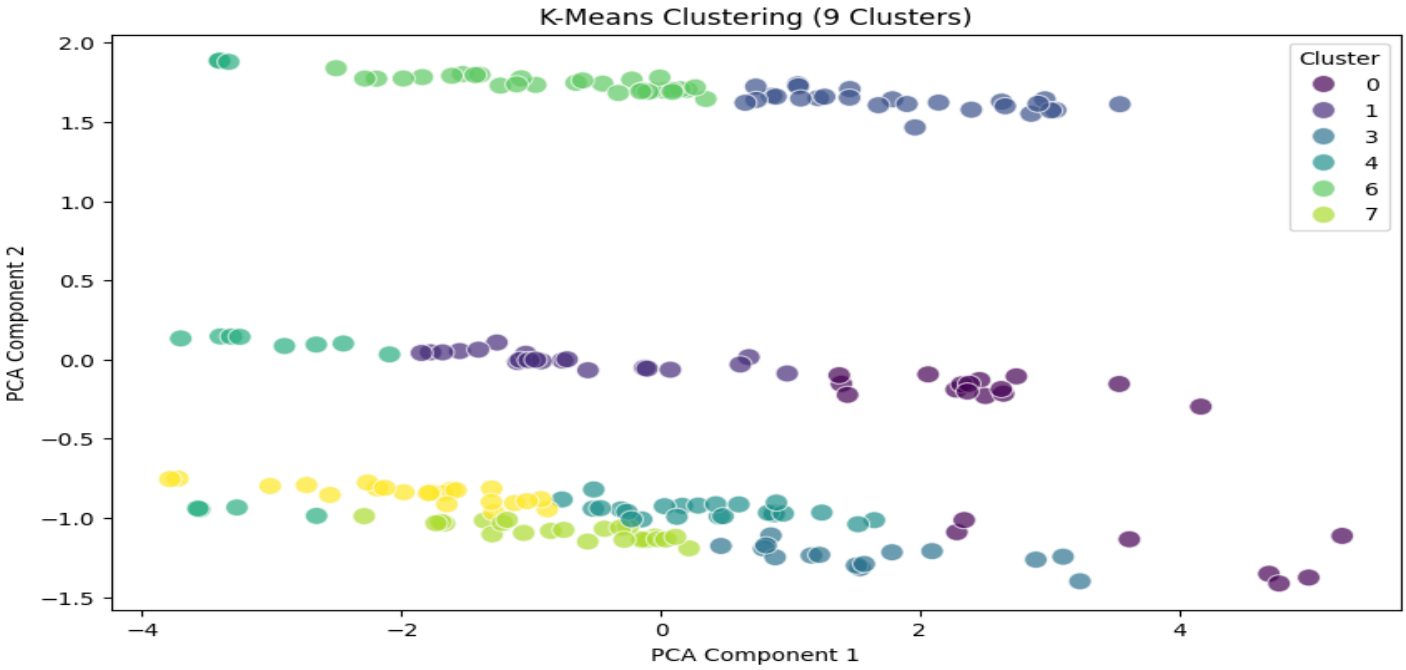
- **eps: 0.7, min_samples: 5**
  - **DB Index: 1.5393 (Higher DB Index suggests less well-separated clusters)**
  - **Silhouette Score: 0.1658 (Indicates poor clustering quality)**

---

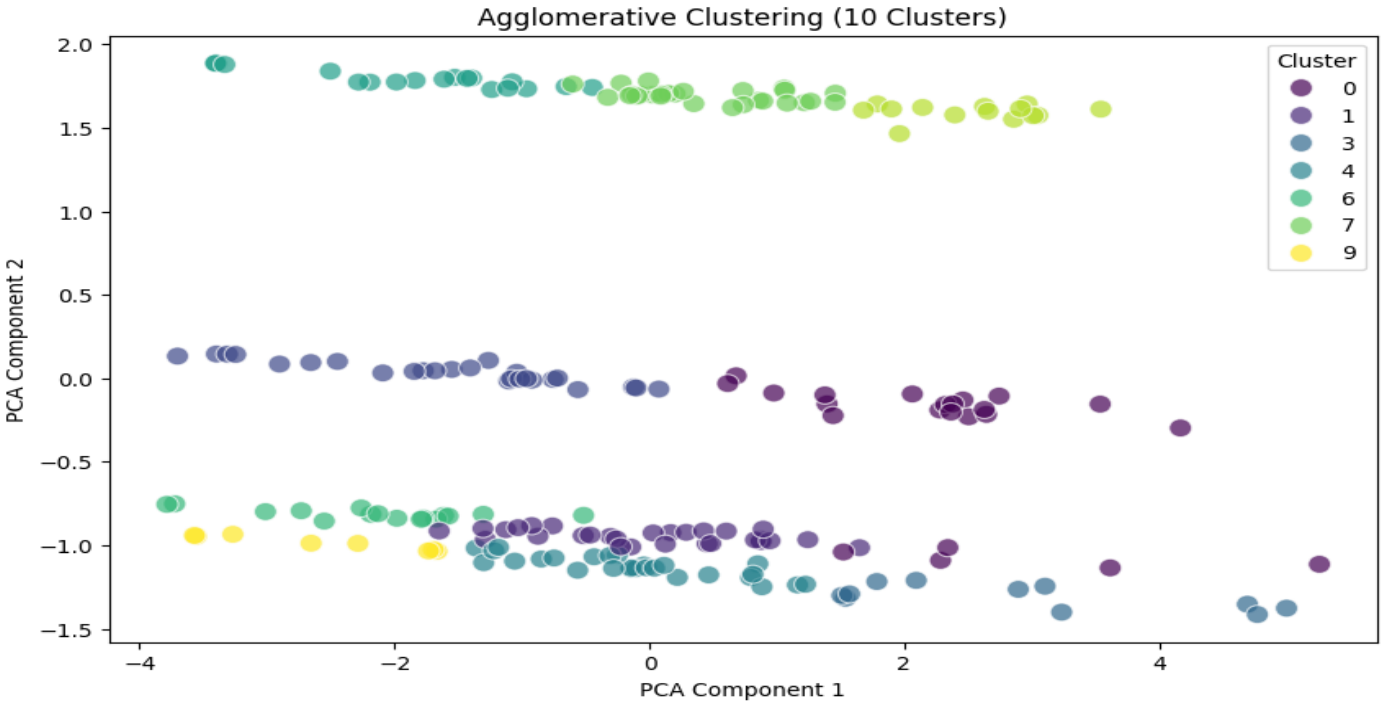
5. Visualizations

To better understand the clustering results, we visualized the clusters using PCA (Principal Component Analysis) to reduce the dimensionality of the feature space to two dimensions.

1. **K-Means (9 clusters):**



2. **Agglomerative Clustering (10 clusters):**



Example:

- The K-Means clustering resulted in well-separated clusters with distinct groupings, whereas Agglomerative Clustering showed a more dense distribution of clusters, with some overlap.

- DBSCAN, on the other hand, struggled with well-defined clusters and left some points as noise due to its density-based nature.

---

6. Conclusion and Recommendations

Based on the evaluation metrics:

- K-Means and Agglomerative Clustering performed the best, with K-Means (9 clusters) showing slightly better results in terms of both DB Index and Silhouette Score.
- DBSCAN struggled with clustering the data well and is not recommended for this dataset in its current configuration.

The clusters obtained can now be used for:

- Targeted Marketing: Different customer groups can be targeted with personalized campaigns.
- Product Recommendations: Customers in similar clusters may have similar product preferences.

Experiment Results Summary

K-Means Clustering

Number of Clusters	DB Index	Silhouette Score
2	1.3356	0.2884
3	1.2480	0.2809
4	1.1660	0.3253
5	1.0226	0.3696
6	1.0260	0.3587
7	1.0114	0.3493
8	0.9399	0.3795
9	0.8987	0.3931
10	0.9146	0.3900

Agglomerative Clustering

Number of Clusters	DB Index	Silhouette Score
2	1.3948	0.2780
3	1.2374	0.3004
4	1.0140	0.3363

Number of Clusters	DB Index	Silhouette Score
5	1.0566	0.3648
6	0.9620	0.3630
7	0.9300	0.3646
8	0.9470	0.3720
9	0.9296	0.3773
10	0.8902	0.3820

DBSCAN Clustering

eps	min_samples	DB Index	Silhouette Score
0.3	5	1.4679	-0.1844
0.5	5	1.4718	-0.2100
0.7	5	1.5393	0.1658
0.7	10	1.5992	-0.0638

This table format provides a concise summary of the various experimental setups for each clustering method, including the **DB Index** and **Silhouette Scores** for each configuration.