Clustering

Part 5

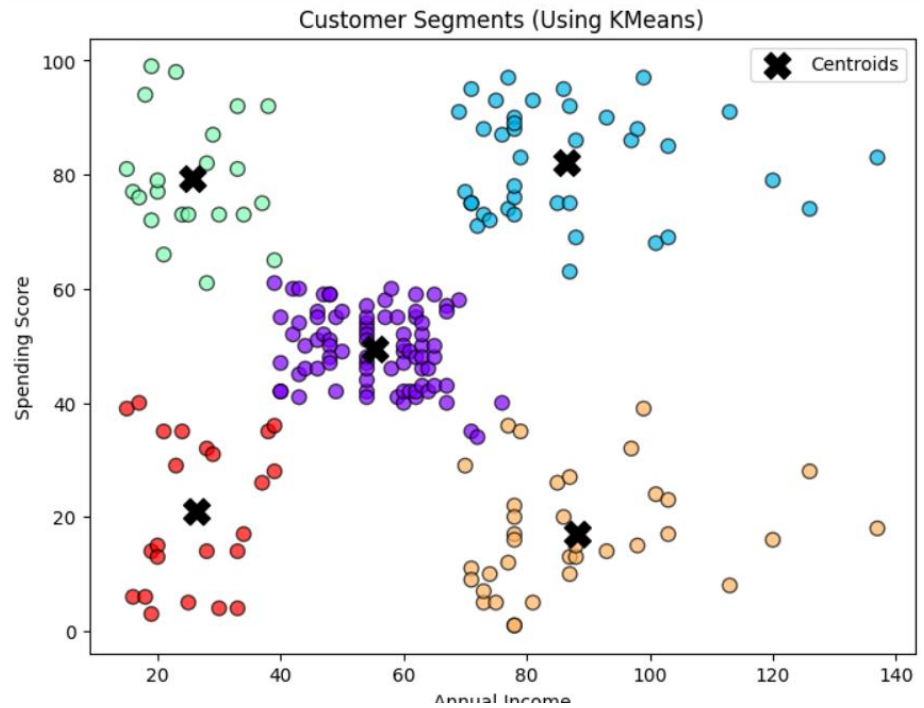
1. Optimal Clusters

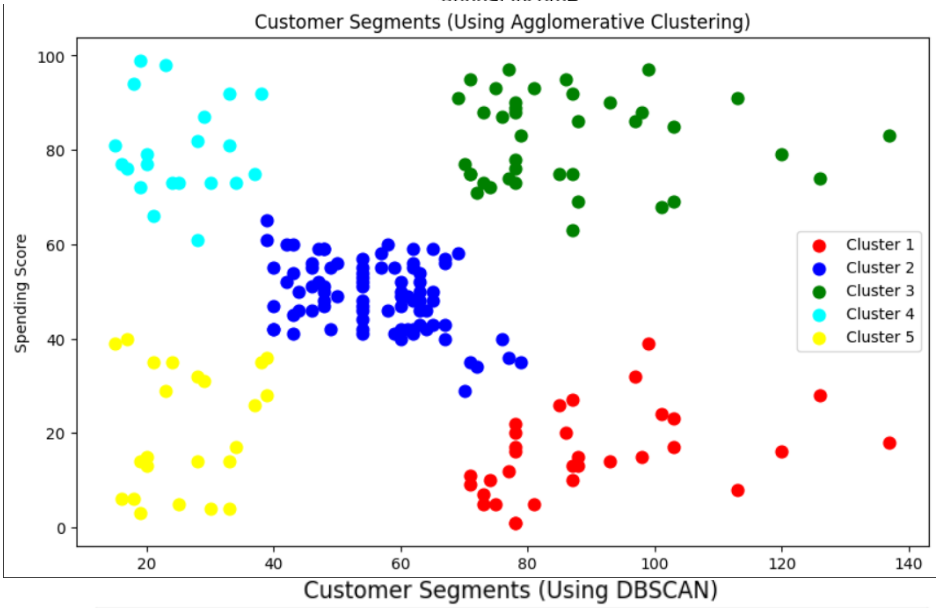
The Optimal Number of clusters identified for K-Means and Hierarchical Clustering is 5.
For K-Means, the WCSS doesn't reduce much after the elbow point, which is identified as 5 here.
For Hierarchical Clustering, we cut the tree where the vertical distance between the merges is largest, which indicates the number of natural clusters.

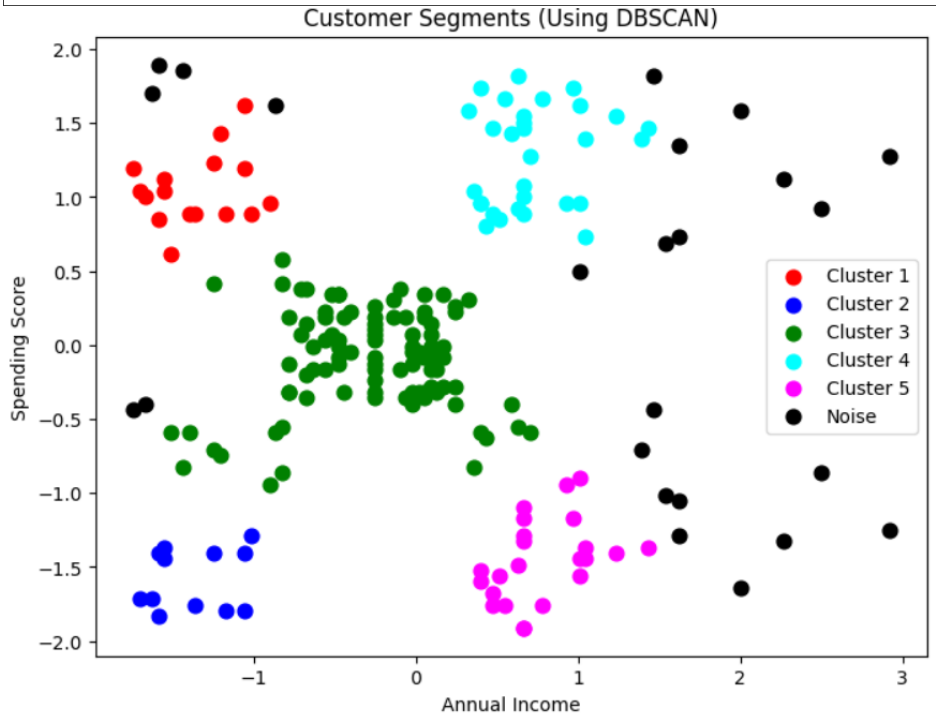
The optimal number of clusters is **5**. This selection is made by setting a **lower distance threshold** (e.g., around an Euclidean distance of 100 on the dendrogram) to cut the hierarchical tree. Choosing a threshold at this height results in **five distinct vertical branches** being intersected, which corresponds to five different customer segments.

2. Cluster Comparison

Based on the visual results, the three algorithms produced different clusterings of the data. **K-Means** produced five distinct, circular clusters of roughly equal size, which is characteristic of its centroid-based approach that aims to partition data into a pre-defined number of groups. **Agglomerative Clustering** also partitioned the data into five clusters, but its hierarchical, bottom-up approach resulted in clusters with more irregular and elongated shapes, reflecting how it merges the closest data points without assuming a specific cluster form. In contrast, **DBSCAN** took a density-based approach, identifying a different number of clusters and also classifying some data points as **"noise"** (the black dots). This is a key difference, as K-Means and Agglomerative Clustering force every point into a cluster, whereas DBSCAN is designed to identify and exclude outliers that do not belong to a dense region. Thus, while K-Means and Agglomerative Clustering shared a similar goal of partitioning the entire dataset, DBSCAN provided a unique, more nuanced result by focusing on data density and noise detection.


Customer Segments (Using KMeans)
Centroids
Spending Score
Annual Income


Customer Segments (Using Agglomerative Clustering)
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Spending Score


Customer Segments (Using DBSCAN)
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Noise
Spending Score
Annual Income

3. DBSCAN Performance

DBSCAN performed effectively on the dataset by identifying **five major clusters** and also classifying a number of data points as **noise**—a capability the other two methods lack. Unlike K-Means and Agglomerative Clustering, which force every data point into a cluster, DBSCAN's approach is more flexible; it only includes points that are part of a sufficiently dense region. As a result, its clustering is fundamentally different, yielding a more accurate representation of the underlying data structure, particularly for datasets that contain outliers or clusters with irregular shapes

4. Algorithm Suitability

For this dataset, the customer groups are compact and clearly separated in the two-dimensional space of Annual Income versus Spending Score. Consequently, K-Means proved to be the most appropriate algorithm. Hierarchical clustering yielded similar results but is less efficient when handling larger datasets. DBSCAN, on the other hand, is better suited for detecting outliers or identifying clusters with irregular shapes.

5. Real World Application

A real-world business scenario where these customer segments could be used is for a targeted marketing campaign. The mall's marketing team could use the segment of **high-income, low-spending** customers to drive a new loyalty program or offer exclusive, high-value promotions. For instance, they could send personalized invitations to a VIP shopping event featuring luxury brands, or offer early access to new product launches. By understanding that this group has the financial capacity but is not currently spending much at the mall, the team can craft specific strategies to incentivize them, converting their high potential into increased sales and engagement.