Small Project 1 – Clustering Algorithms

Jacob D. Evans

University of West Georgia

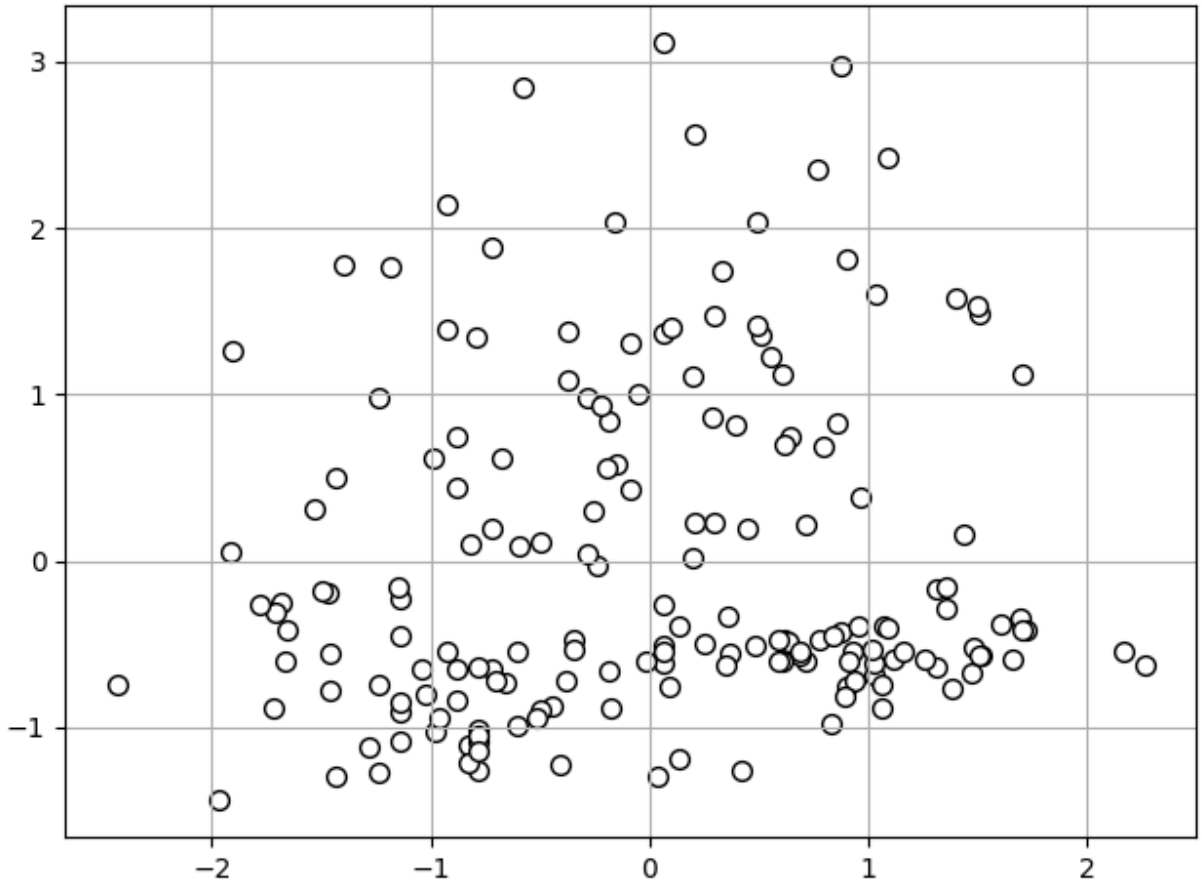
CS 4275 Machine Learning Foundations

Dr. Md Shirajum Munir

October 8, 2024

Small Project 1 – Clustering Algorithms

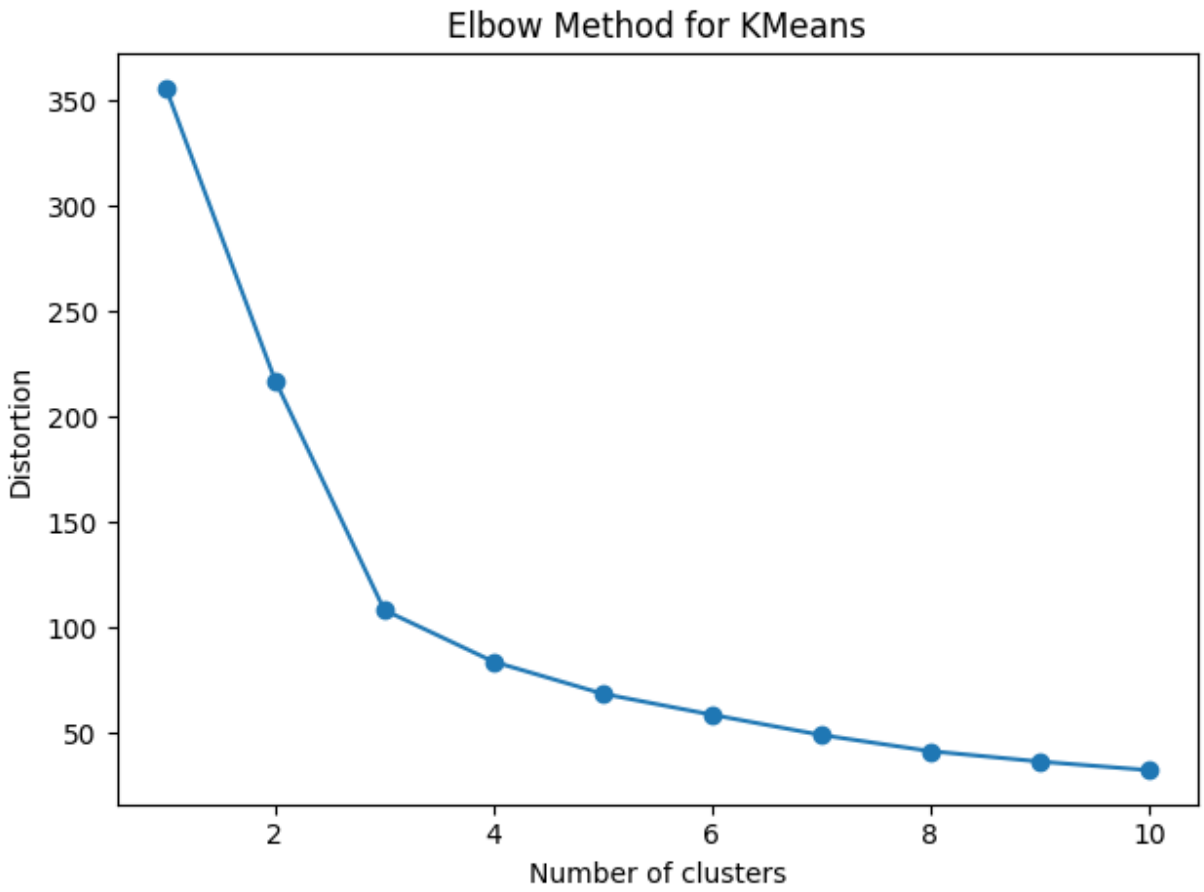
This report compares the performance of three types of clustering algorithms, K-means, DBSCAN and hierarchical clustering. Each algorithm used an unlabeled variation of the wine dataset located locally within the project to generate a silhouette plot for three specified cluster amounts. We chose three, four and five clusters for these amounts. The following figure shows the initial points of data before any structuring by the algorithms. The data remained unchanged throughout the project.



K-means

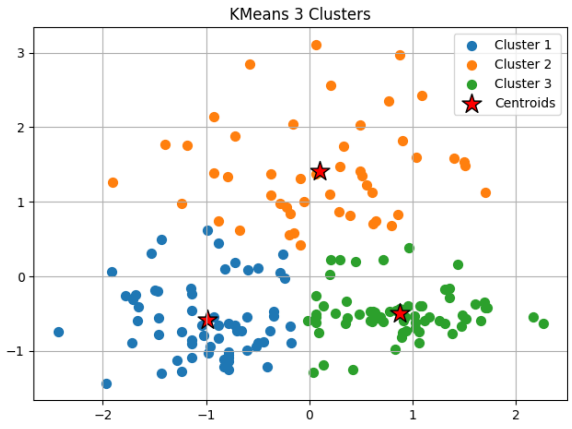
K-means is a clustering algorithm that divides data into a specified number of clusters (K). It starts by randomly selecting K centroids and assigns each data point to the nearest centroid based on distance. After assignment, the centroids are recalculated as the average of the points in each cluster. This process repeats until the centroids stabilize. K-means is simple and efficient but requires specifying the

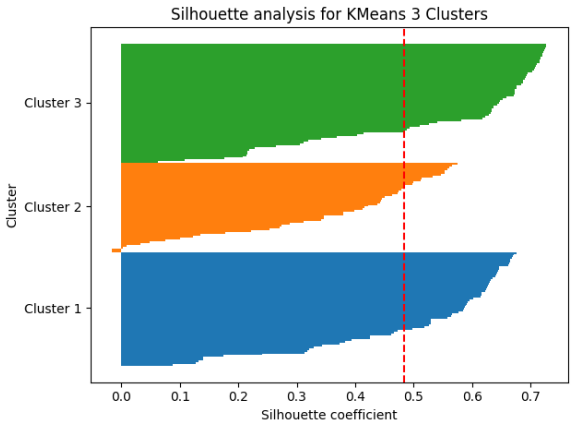
number of clusters in advance. Before applying the K-means algorithm, it's essential to examine the elbow plot to hypothesize where the algorithm might achieve optimal performance. The figure below illustrates the elbow plot for up to ten clusters after running the algorithm.



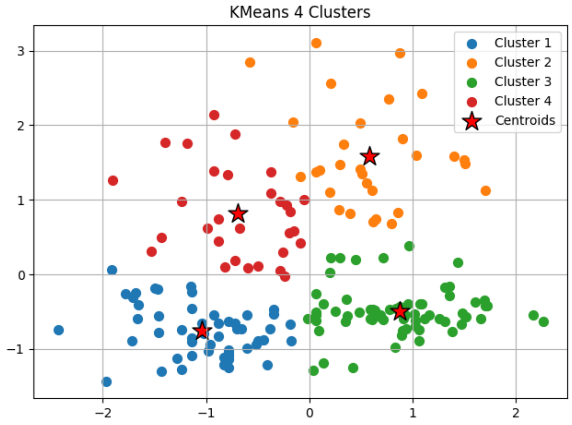
According to the figure, the optimal number of clusters should be around 2-4 clusters before we start to see diminishing returns.

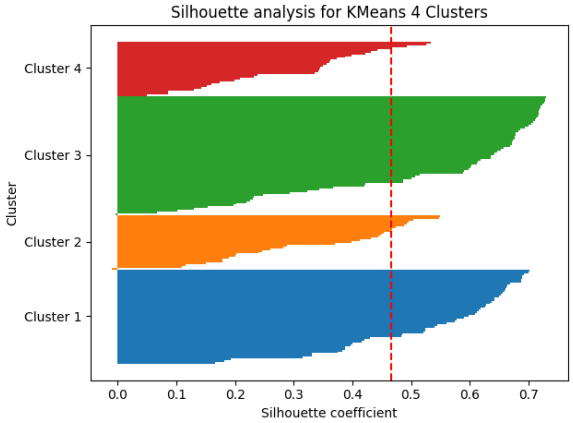
K-means - 3 clusters



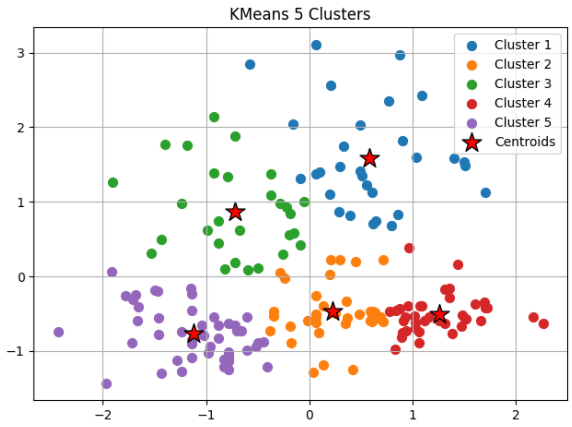


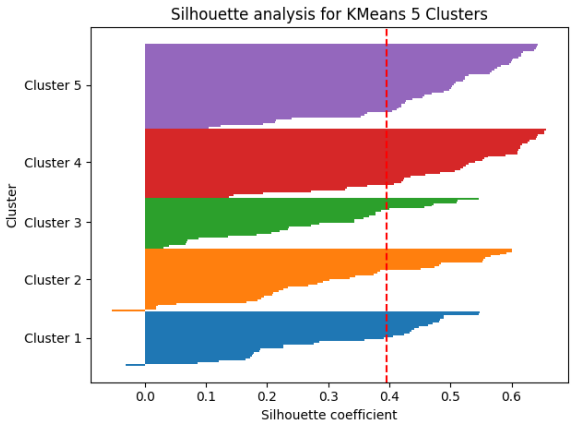
K-means - 4 clusters





K-means - 5 clusters



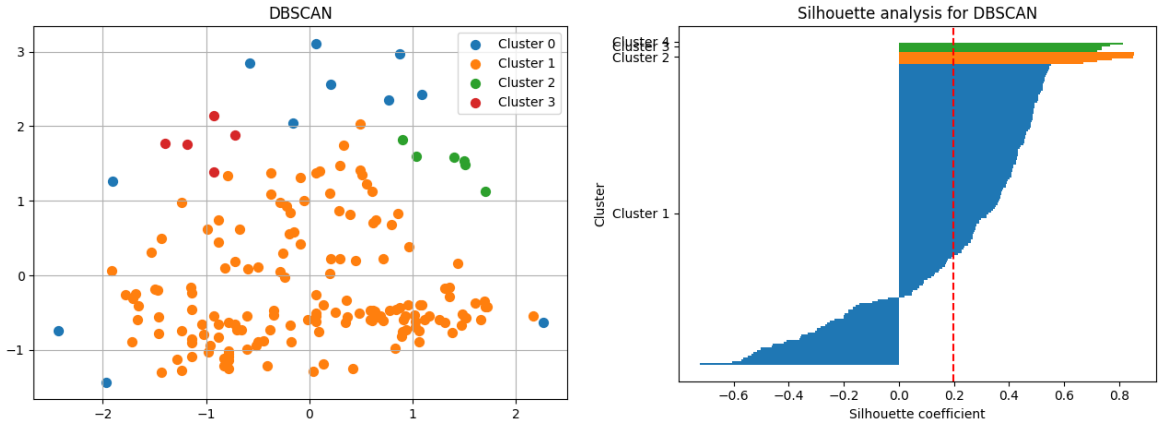


Analyzing the silhouette plots provides insight into the optimal number of clusters for structuring the data. With four clusters, the K-means algorithm shows signs of overfitting, resulting in a noticeable disparity between the blue/green and red/orange clusters. Based on the output, I would recommend using either three or five clusters, as both seem to better represent the data, with three clusters exhibiting less overfitting.

DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm that groups data points based on their density. It identifies clusters as dense regions separated by areas of lower density using two parameters: epsilon (ε), the maximum distance to consider points as neighbors, and minPts, the minimum number of points required to form a dense area. Points are classified as core points (which have enough neighbors), border points (near core points), or noise points (not part of any cluster), making DBSCAN effective for detecting clusters of varying shapes and sizes, even in noisy data.

DBSCAN – clusters



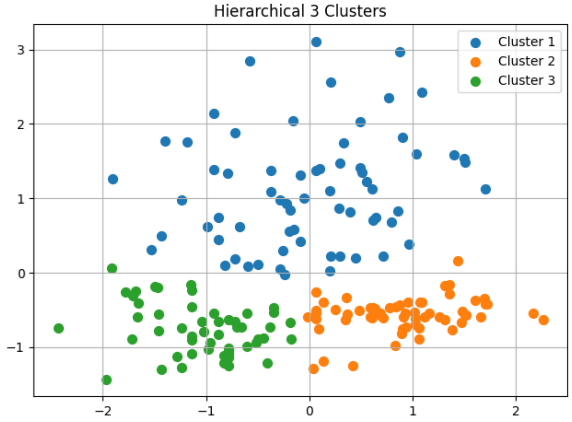
Analyzing the silhouette plots for the DBSCAN shows that the majority of the data points are grouped together, indicating a dense region. However, the presence of multiple small clusters indicates that the algorithm is overfitting to noise or outliers, which distorts the true data structure. These small clusters do not represent meaningful groupings but rather minor variations or isolated points within the data. This implies that while the algorithm captures the main trend of the data, it may be overly sensitive to noise,

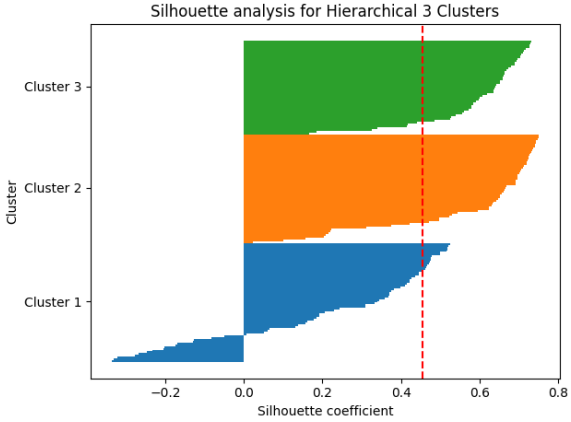
leading to the fragmentation of the dataset into numerous small clusters instead of a more coherent clustering structure. A better outcome would balance the large cluster with fewer, more substantial clusters that reflect the actual relationships within the data.

Hierarchical clustering

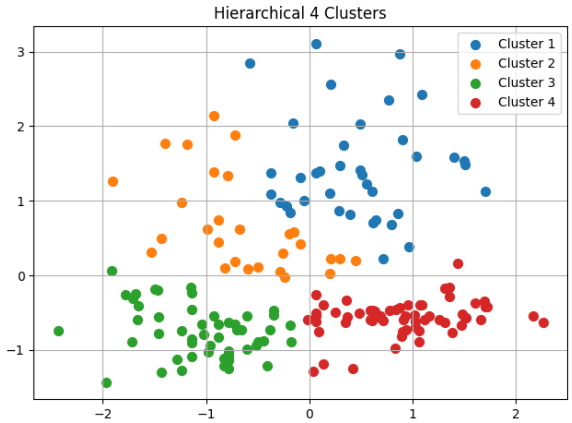
Hierarchical clustering creates a tree-like structure (dendrogram) to group data points based on similarities. It can be agglomerative (starting with individual points and merging clusters) or divisive (starting with one cluster and splitting it). This method does not require specifying the number of clusters beforehand and allows for visualization of clustering at different levels. However, it can be computationally intensive for large datasets.

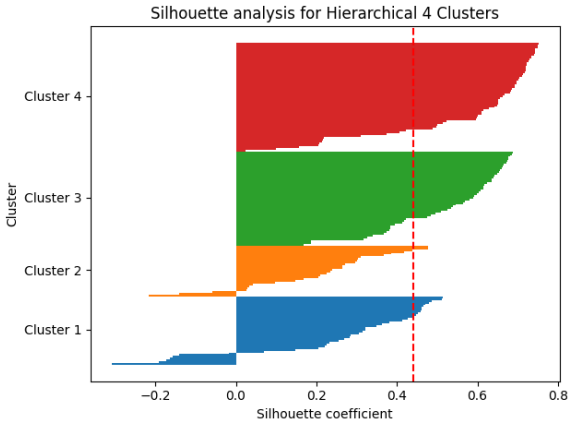
Hierarchical clustering - 3 clusters



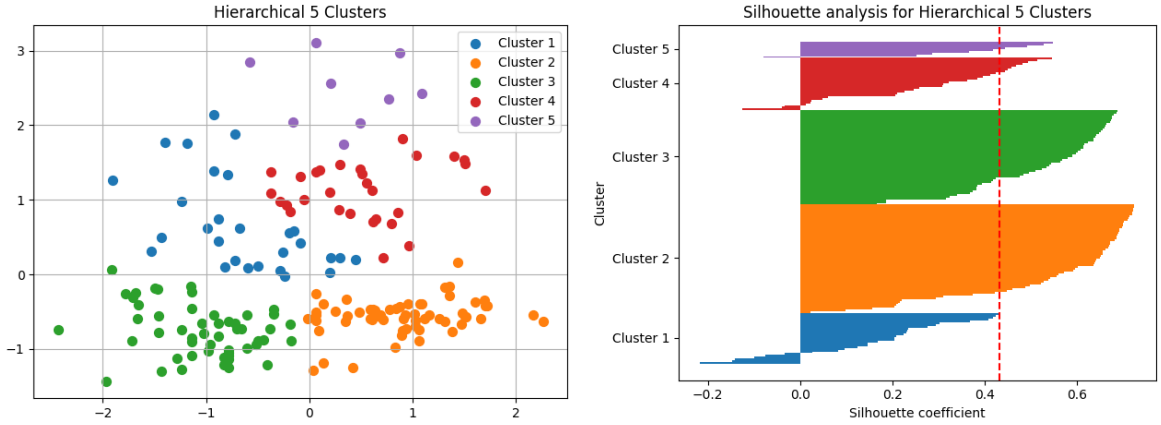


Hierarchical clustering - 4 clusters





Hierarchical clustering - 5 clusters



Sharing similar results with the K-means algorithm, the hierarchical clustering algorithm shows signs of overfitting, especially with four and five clusters. Based on the output, I would recommend using the three-cluster iteration.

Conclusion

In this project, we explored and compared the performance of three clustering algorithms: K-means, DBSCAN, and hierarchical clustering, using an unlabeled variation of the wine dataset. Each algorithm provided unique insights into the structure of the data, but they also exhibited similar challenges, particularly regarding overfitting. K-means showed that while three and five clusters could effectively represent the data, four clusters led to noticeable overfitting. DBSCAN revealed a dominant cluster accompanied by numerous small clusters, indicating sensitivity to noise rather than meaningful groupings. Finally, hierarchical clustering mirrored the results of K-means, with three clusters appearing to provide the most coherent representation. Overall, the analysis suggests that selecting the appropriate number of clusters is crucial for optimizing the clustering process and capturing the underlying data structure, highlighting the importance of evaluating clustering algorithms through multiple perspectives.