Thomas Ehret

Assignment 6

CS519 SP23

Dr. Cao

Clustering Analysis

The IRIS and MNIST datasets were used to test the unsupervised learning abilities of KMeans and Hierarchical models.

Testing Specifications

The performance of each clustering model was measured by timing the 'fit' process and computing the Sum Squared Error (SSE) and silhouette score. The hierarchical models were visualized with dendrograms and computed cluster labels. The hierarchical clustering algorithm has two main parameters: "linkage" and "distance_threshold". The linkage parameter determined the method used to calculate the distance between clusters, while the distance_threshold parameter determined the distance threshold at which clusters would be merged. The Elbow method was implemented to make a comparison on the number of clusters needed for the datasets.

Defaults Parameters:

Iris:

clusters = 3

KMeans(n_clusters=clusters, init='k-means++', n_init='auto', max_iter=iterations, tol=1e-04, random_state=0)

AgglomerativeClustering(n_clusters=None, metric='euclidean', linkage='complete', distance_threshold=0)

linkage(X, method='average', metric='euclidean')

MNIST:

clusters = 10

KMeans(n_clusters=clusters, init='k-means++', n_init='auto', max_iter=iterations, tol=1e-04, random_state=0)

AgglomerativeClustering(n_clusters=None, metric='euclidean', linkage='complete', distance_threshold=0)

linkage(X, method='average', metric='euclidean')

Iris Dataset

The initial performance of the models on the dataset is shown in (table 1). The clustering parameter of 3 was informed by the Elbow method (figure 1). The Silhouette plot and coefficient was

computed on the KMeans algorithm (figure 2) with a coefficient of 0.55, which indicates that the clusters are reasonably well-separated, but there may be some overlap between the clusters. For this small dataset changing the number of clusters to 6 for KMeans didn't produce any significant computational time increase but did decrease the SSE to 39. However, we know that 6 isn't an informative clustering result. Furthermore, modifying the metric, linkage, and distance_threshold parameters had no significant effect on computation time or the SSE.

Default Parameters		
Model	**Time**	**Sum Squared Error**
KMeans	0.02ms	78.856
AggClustering	0.10ms	89.5
SciPy Linkage	0.08ms	150

Table 1

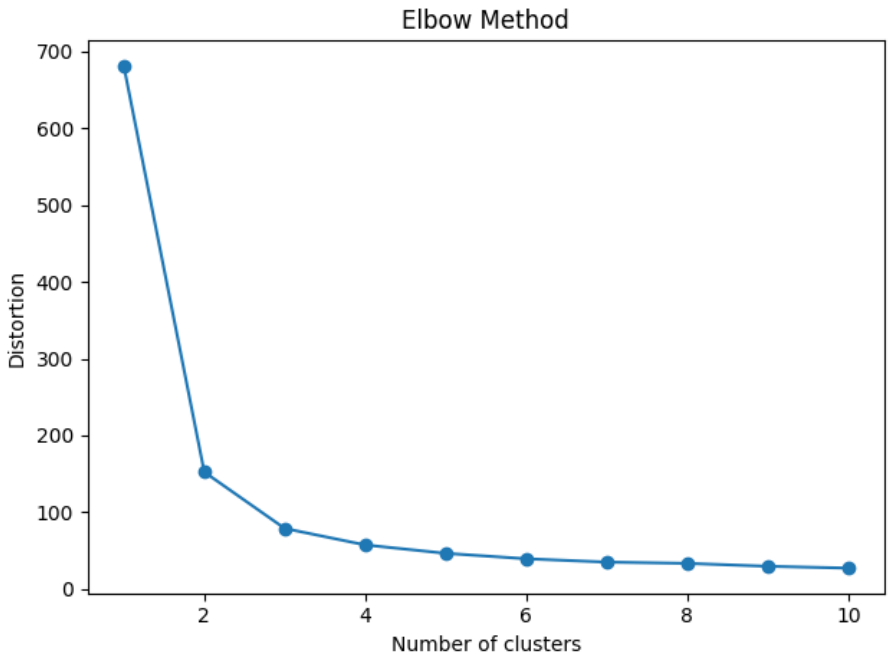


Figure 1

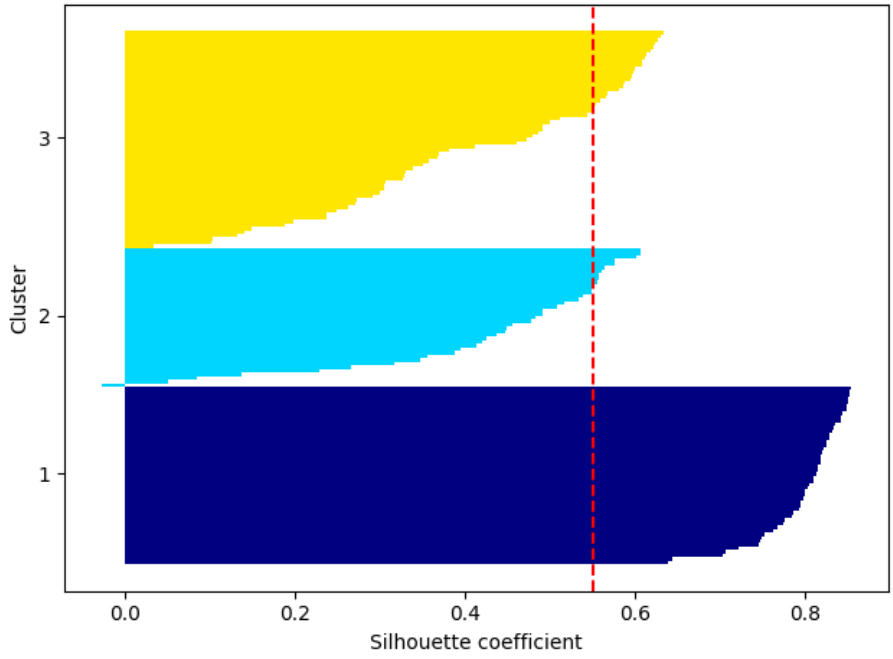


Figure 2

The accompanying dendrograms are shown in figures 3 and 4. A truncation level was set at 5 to make the diagrams easier to visualize

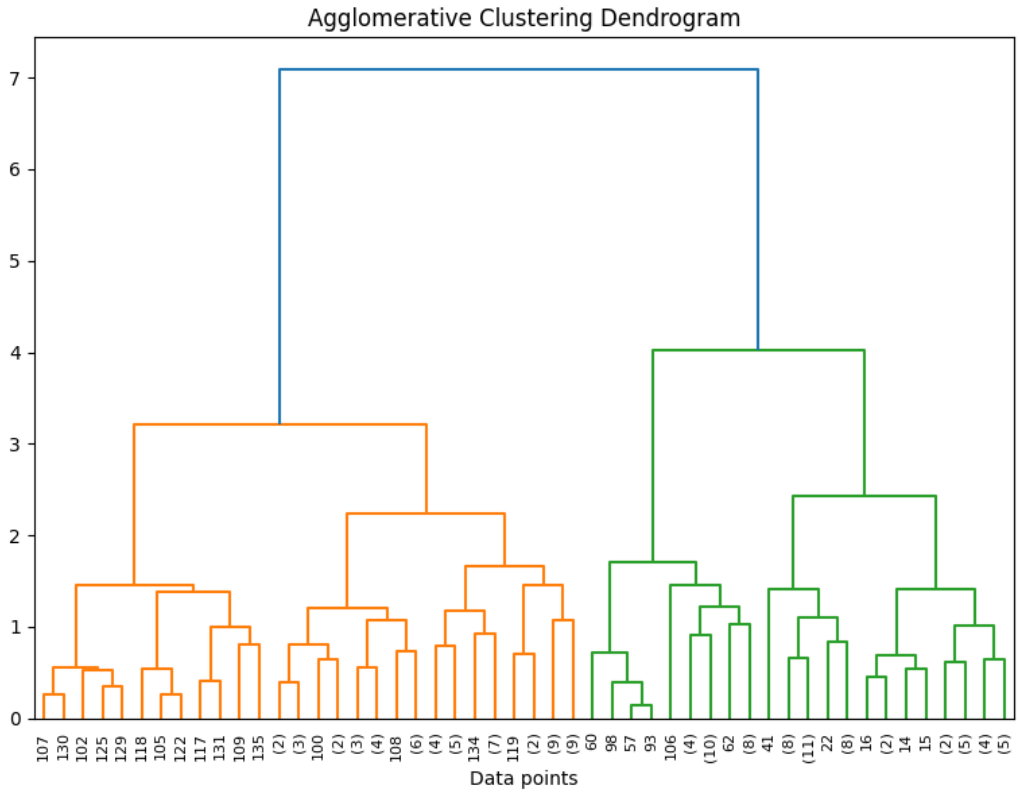


Figure 3

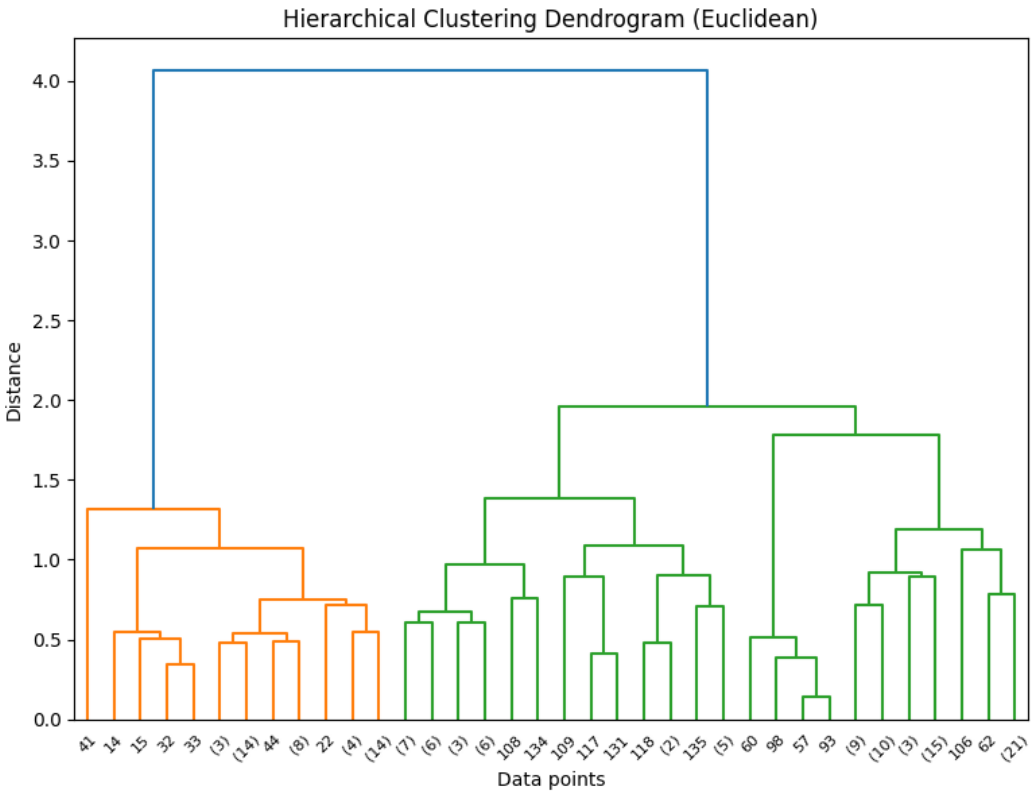


Figure 4

MNIST Dataset

Default Parameters		
Model	**Time**	**Squared Error**
KMeans	0.008ms	1202391
AggClustering	0.009ms	89.5
SciPy Linkage	0.003ms	150

Table 2

The clustering parameter of 10 was informed by the Elbow method (figure 4), however the shape of the curve with the MINIST dataset is less informative than for the Iris dataset. Based on the results of the initial testing (table 2), we can see that the algorithm was run with two different linkage methods: agglomerative and agglomerative_2. Both methods achieved similar performance metrics, with agglomerative_2 having a slightly worse training time. Therefore, we can assume that the choice of linkage method may not have a significant impact on the performance of the algorithm. However, this assumption may not hold true for all datasets, as different linkage methods may be more suitable for different types of data. We can assume that increasing the distance threshold may lead to the merging of more clusters, while decreasing the distance threshold may lead to the creation of more clusters.

The average silhouette_score is: 0.189. Given the size of the MNIST dataset this score indicates that the clustering algorithm has room for improvement, but it is still producing reasonably well-separated clusters (figure 6). Dendrograms are shown in figures 7 through 9.

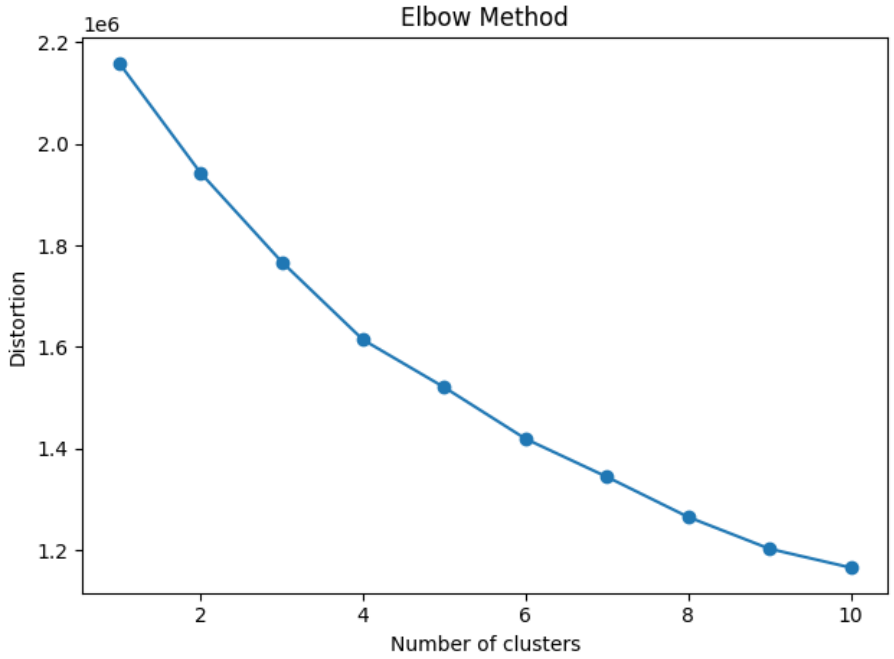


Figure 5

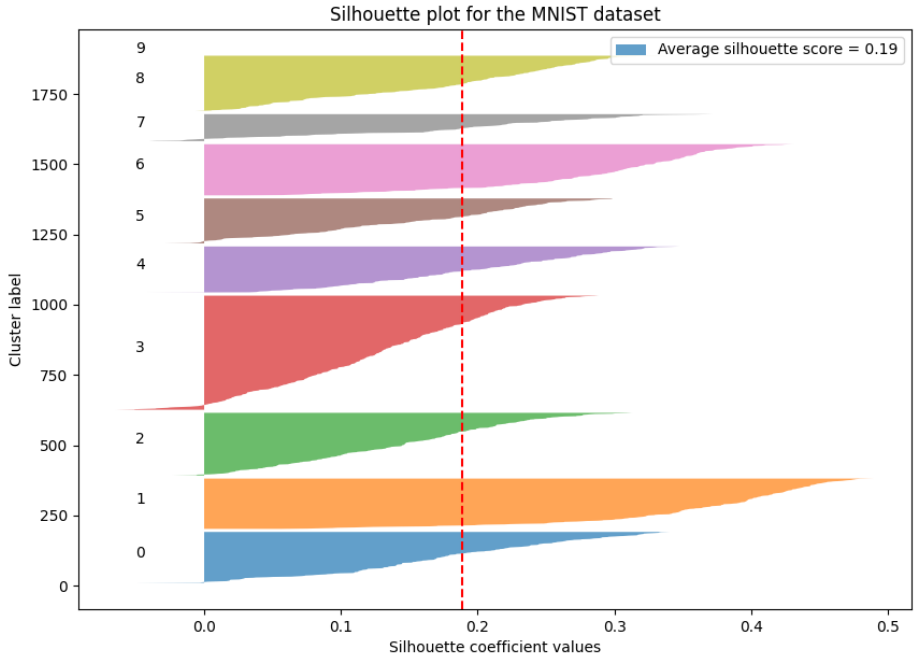


Figure 6

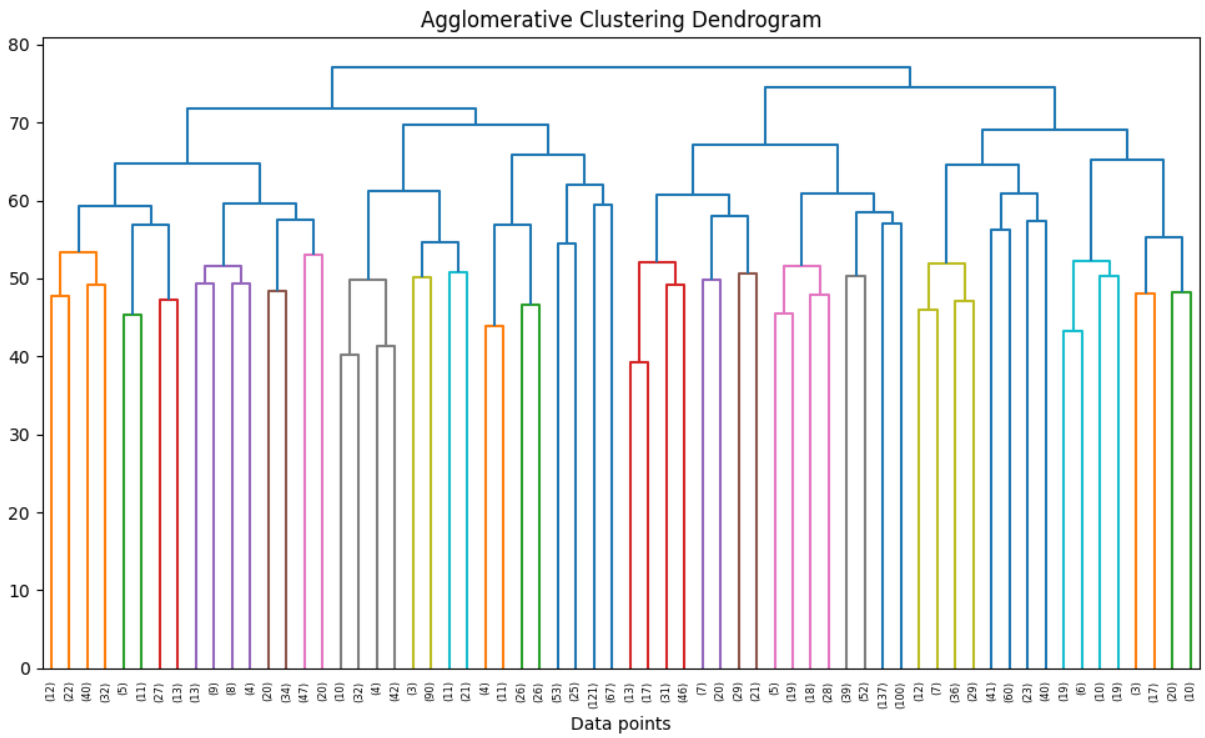


Figure 7

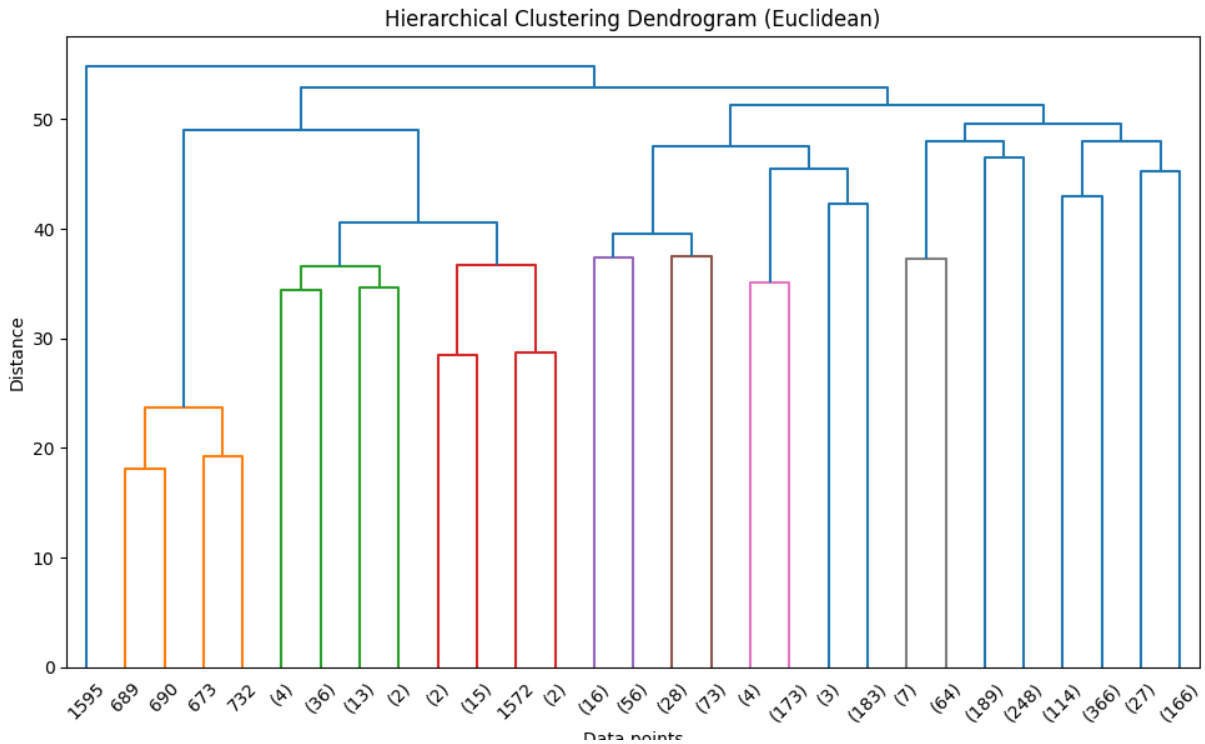
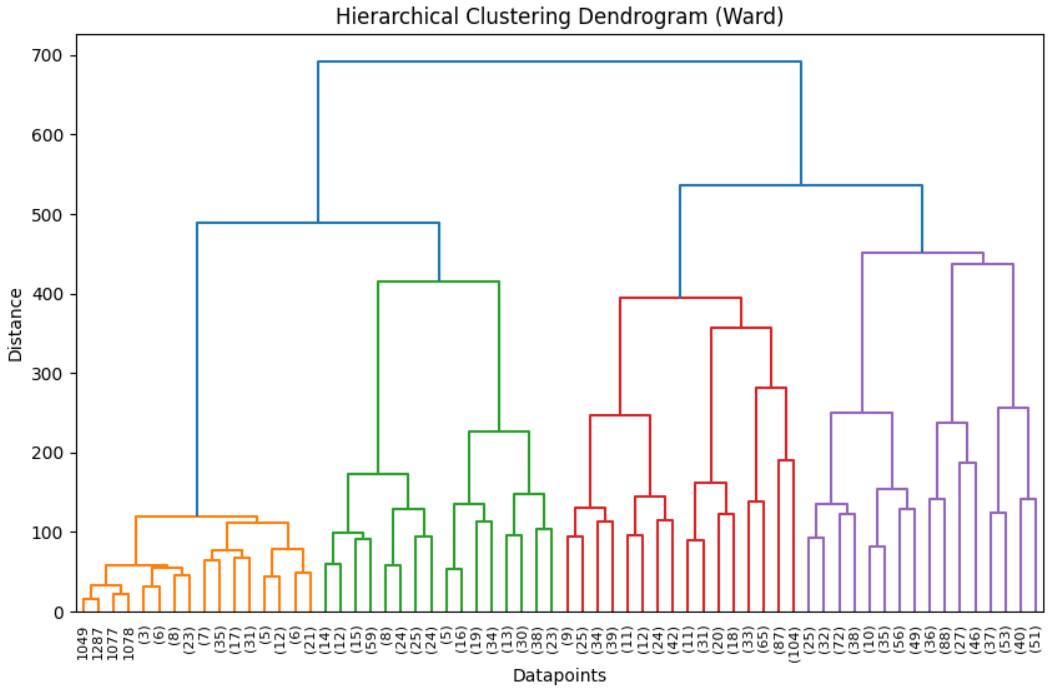


Figure 8



Overall this first set of metrics is a bit underwhelming. While KMeans was able to cluster the data points quickly, it was not able to group them tightly around their respective cluster centers.

For the next round of testing the parameters were changed to parameters column in table 3:

KMeans Testing		
Parameters	**Time**	**Squared Error**
max_iter=500	0.02ms	1165290
max_iter = 1000, init = random	0.14ms	1165142
max_iter = 1000, init = K-means++	0.11ms	1165177
max_iter = 1000, init = K-means++, algorithm=elkan	0.05ms	1165177

Table 3

Once again, the results are very similar regardless of the iterations and algorithm. The only significant change is computation time decreases when using the 'Elkan' algorithm. Several rounds of testing followed with changes to :Kmeans (n_*clusters = " 5 -> 10", n_init = "10 -> 500");, skleanrn hierarchical (metric = linkage (ward, Euclidean), manhatten, cosine;. Scipy hierarchical(single,complete,centroid)*

While these metrics provide some insights into the performance of the clustering algorithm, it appears that changing the parameters had a limited impact on their performance. While there were differences in training time and silhouette scores between the KMeans and Hierarchical clustering algorithms, these differences were relatively small. It is possible that the dataset itself is a more significant factor in determining the performance of the clustering algorithms than the specific parameters used. As such, it may be necessary to explore alternative datasets or preprocessing techniques in order to improve the performance of these algorithms.