SENG 474 Assignment 3

Analysis of Clustering Methods

Ryan Russell

July 29, 2020

1. K-Means Algorithm Analysis

The K-means algorithm, also known as Lloyd's Algorithm, was the first clustering method used on the two datasets. This clustering method divides a cluster of data points into K different groups by analyzing the data to find points with similar characteristics [2]. The data points in each cluster group will be closer to the centroid for their group than the centroids for other groups. The Euclidean distance was used for this implementation along with two forms of initialization: uniform random initialization and k-means++ initialization. For each experiment in this section, the optimal K value will be determined using the elbow method and a graph of the cost vs. number of clusters K. Furthermore, the model using this optimal K value will be compared with other experiments using a metric known as the completeness score.

1.1 Two-Dimensional Data

The first set of tests were run on the two-dimensional data available in dataset1.csv, which were generated by a Gaussian mixture model. The first initialization method that was tested was uniform random initialization. The K value (number of clusters) was varied from 1 to 20 to produce the cost graph shown in Figure 1. The cost was calculated using the sum of squared errors. Using the elbow method, the optimal number of clusters is K = 4, which produced a 0.898 completeness score.

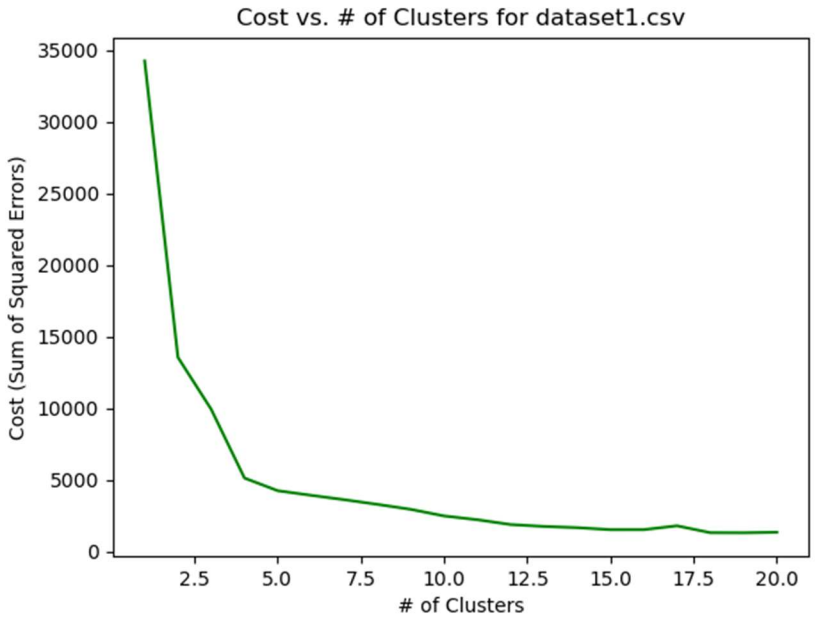

Figure 1: Cost Graph for 2D K-means

After producing the cost graph, The K value was varied from 2 to 20 to produce scatter plots and visualize the 2D data. Figures 2 through 6 show several of these scatter plots. Specifically, the plots for K values of 2, 6, 10, 14, and 18 are displayed.

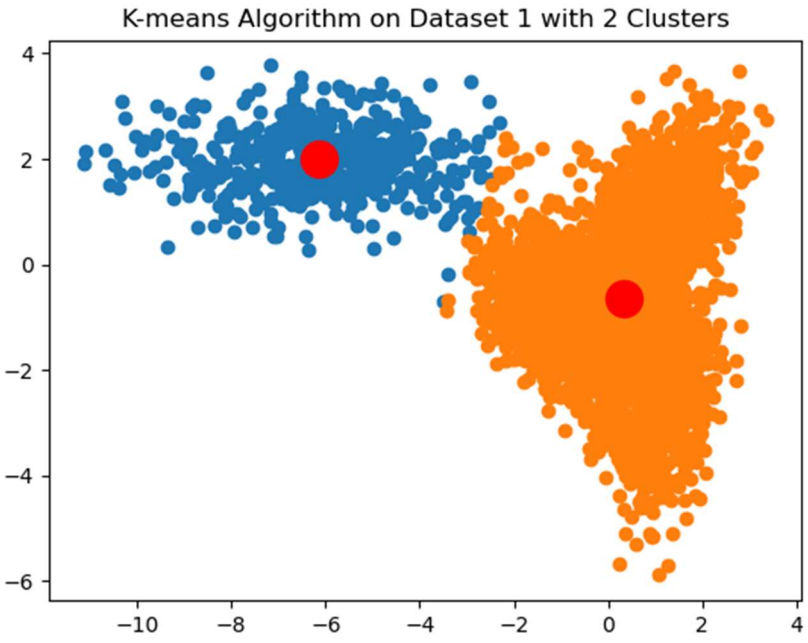

Figure 2: 2D K-means with K = 2

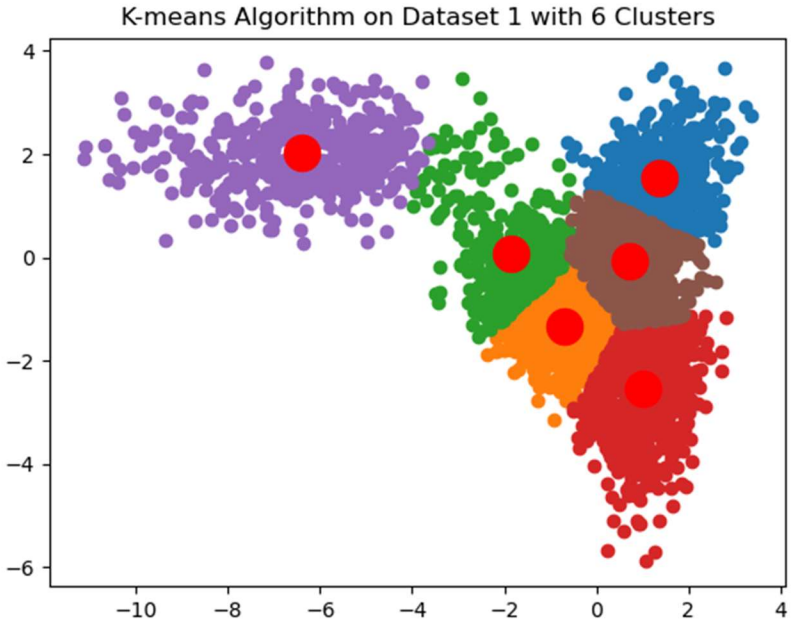

Figure 3: 2D K-means with K = 6

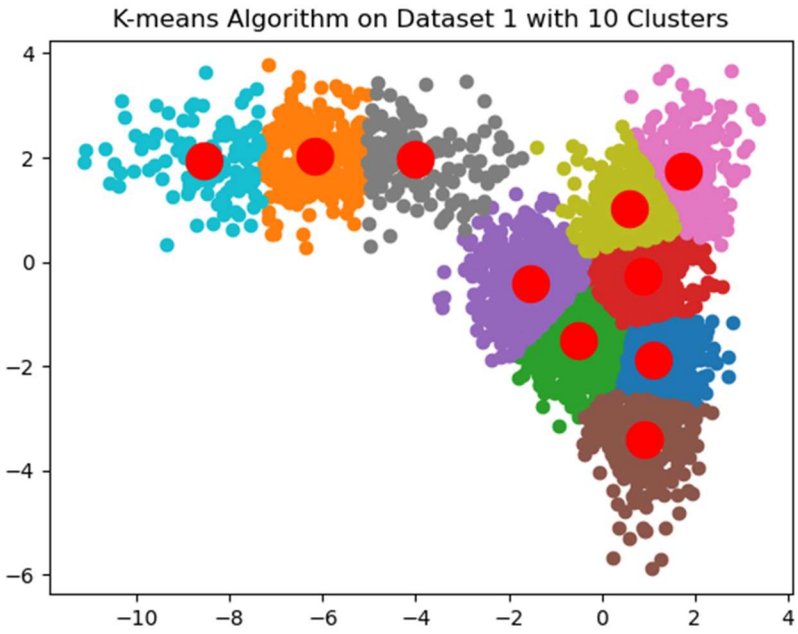

Figure 4: 2D K-means with K = 10

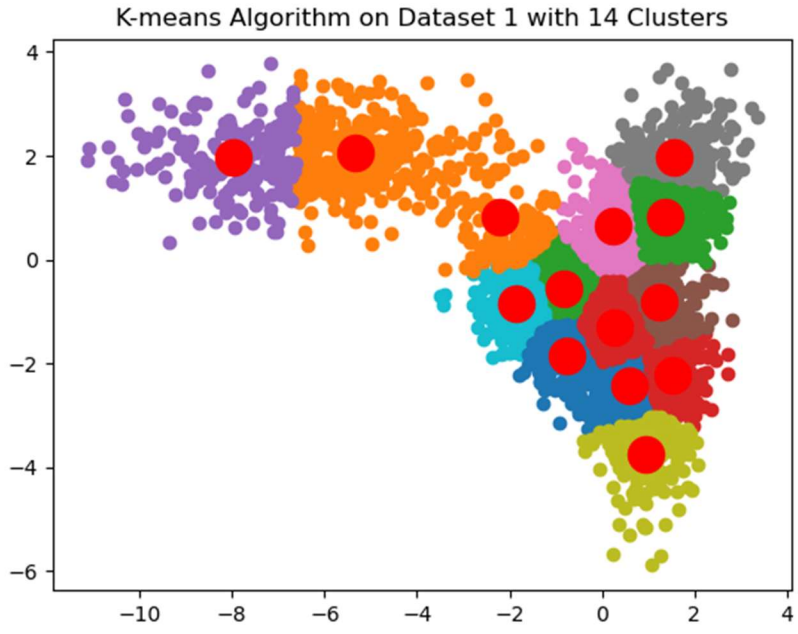

Figure 5: 2D K-means with K = 14

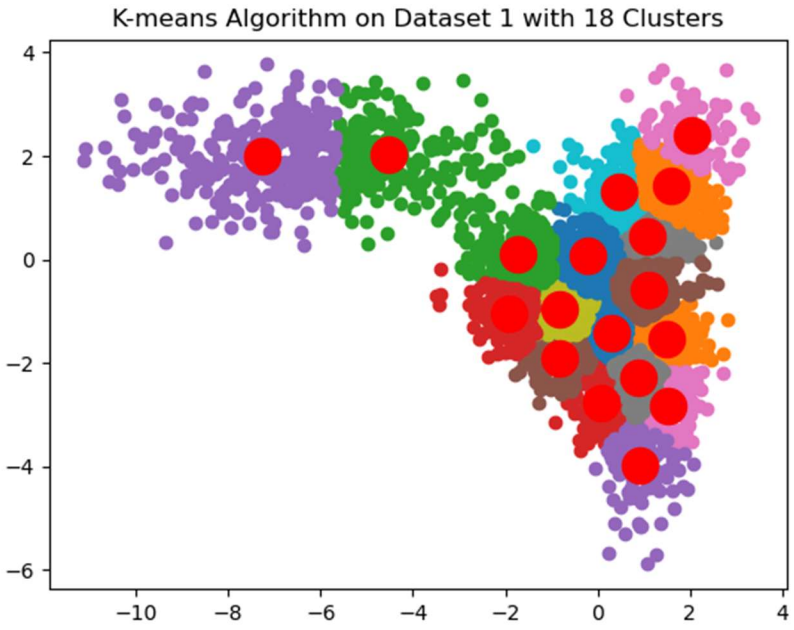

Figure 6: 2D K-means with K = 18

The second initialization method that was tested was k-means++ initialization. The K value (number of clusters) was again varied from 1 to 20 to produce the cost (sum of squared errors) graph shown in Figure 7. In the same way as before, the optimal number of clusters was discovered to be K = 4, which produced a 0.943 completeness score.

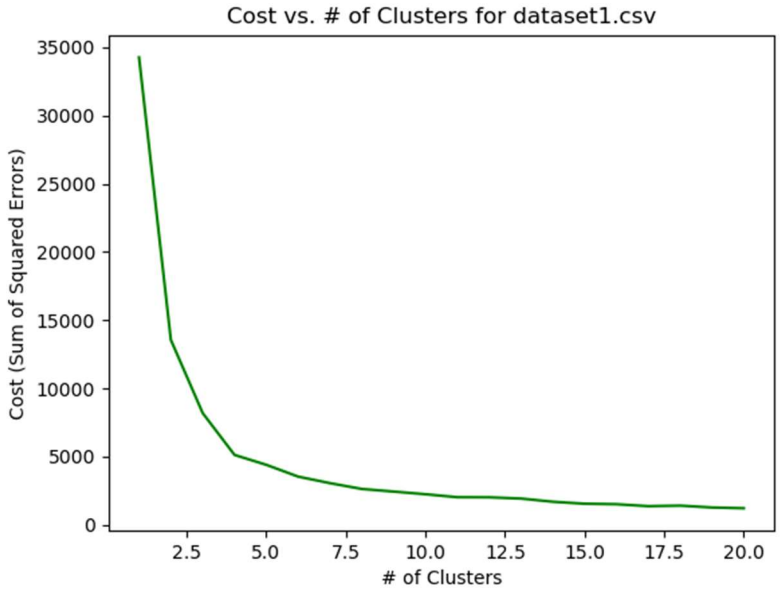

Figure 7: Cost Graph for 2D K-means++

After producing the cost graph, The K value was varied from 2 to 20 to produce scatter plots and visualize the 2D data. Figures 8 through 12 show several of these scatter plots. As with uniform random initialization, the plots for K values of 2, 6, 10, 14, and 18 are displayed.

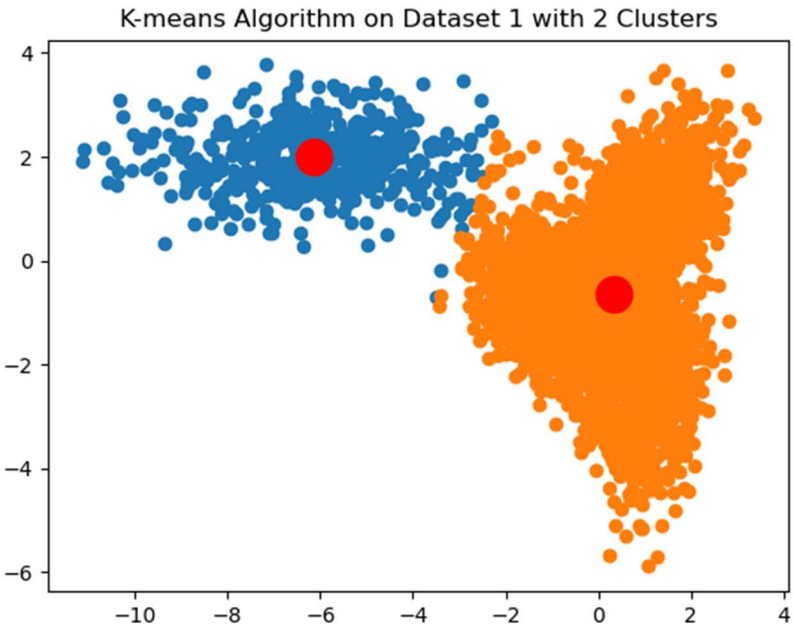

Figure 8: 2D K-means++ with K = 2

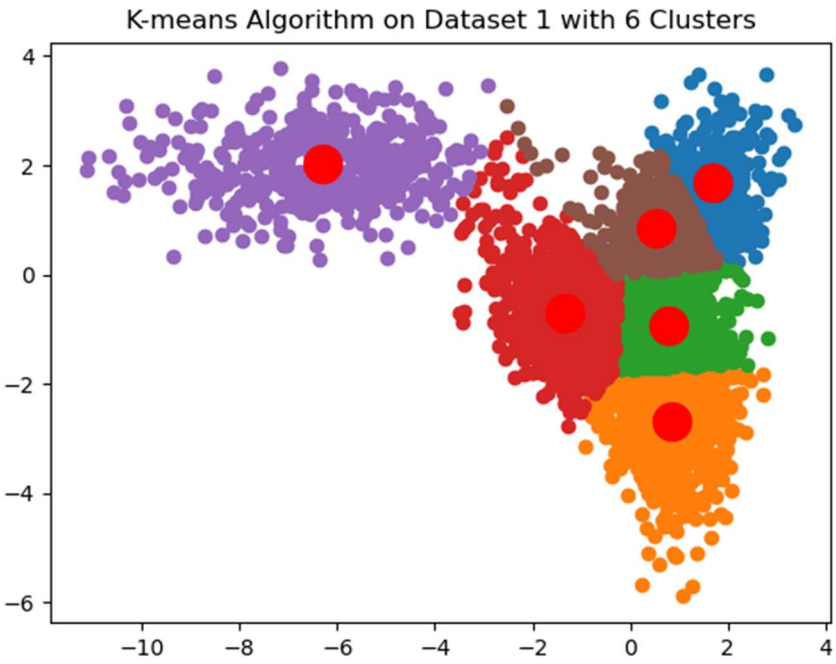

Figure 9: 2D K-means++ with K = 6

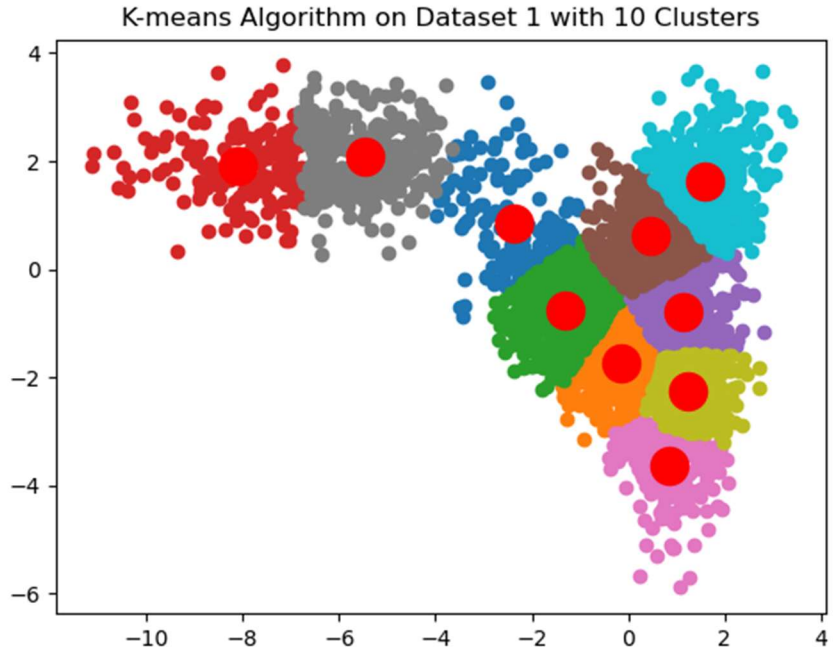

Figure 10: 2D K-means++ with K = 10

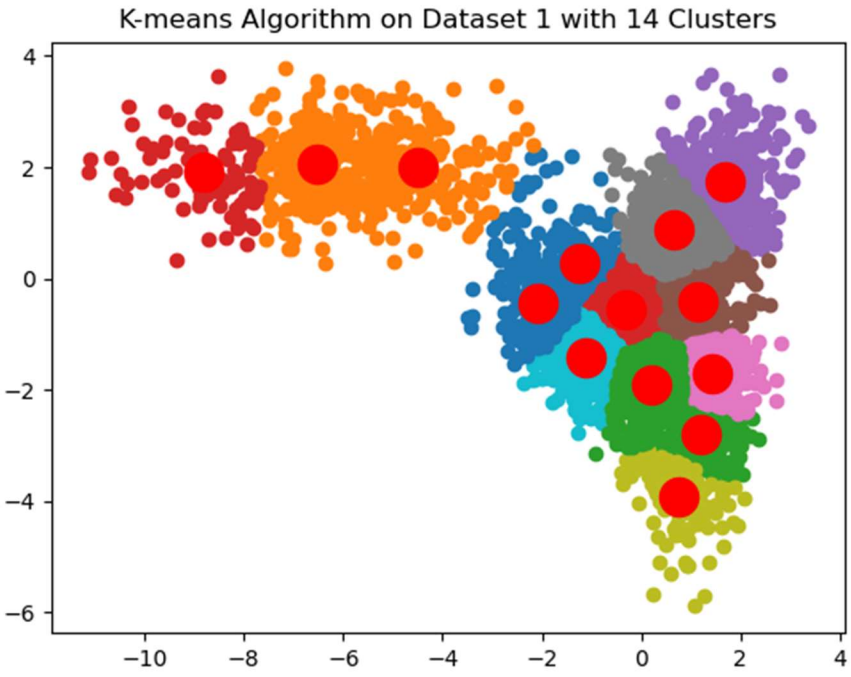

Figure 11: 2D K-means++ with K = 14

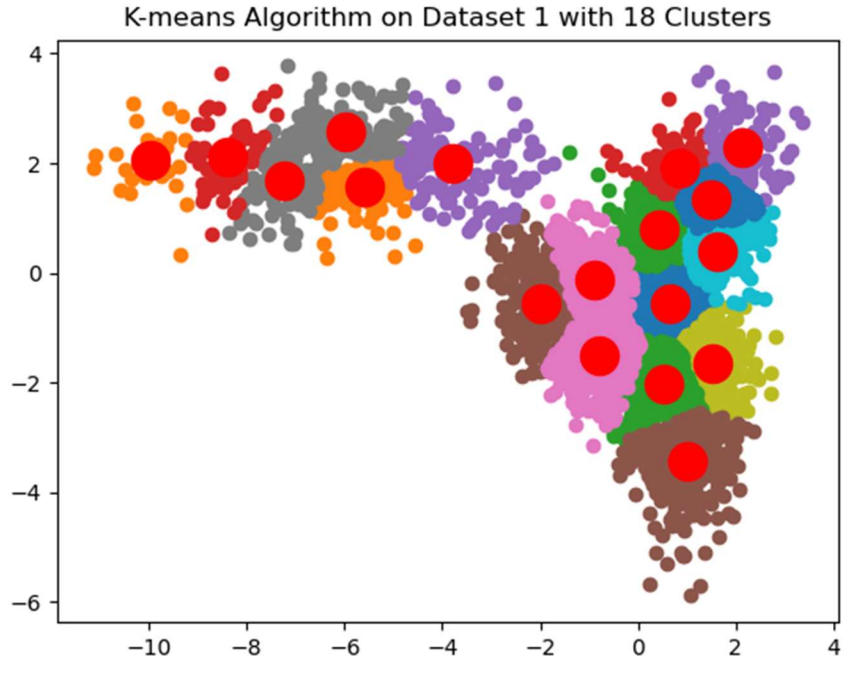

Figure 12: 2D K-means++ with K = 18

1.2 Three-Dimensional Data

The second set of tests were run on the three-dimensional data available in dataset2.csv. The first initialization method that was tested was uniform random initialization. The K value (number of clusters) was varied from 1 to 20 to produce the cost graph shown in Figure 13. The cost was calculated using the sum of squared errors. Using the elbow method, the optimal number of clusters is K = 8, which produced a 0.936 completeness score.

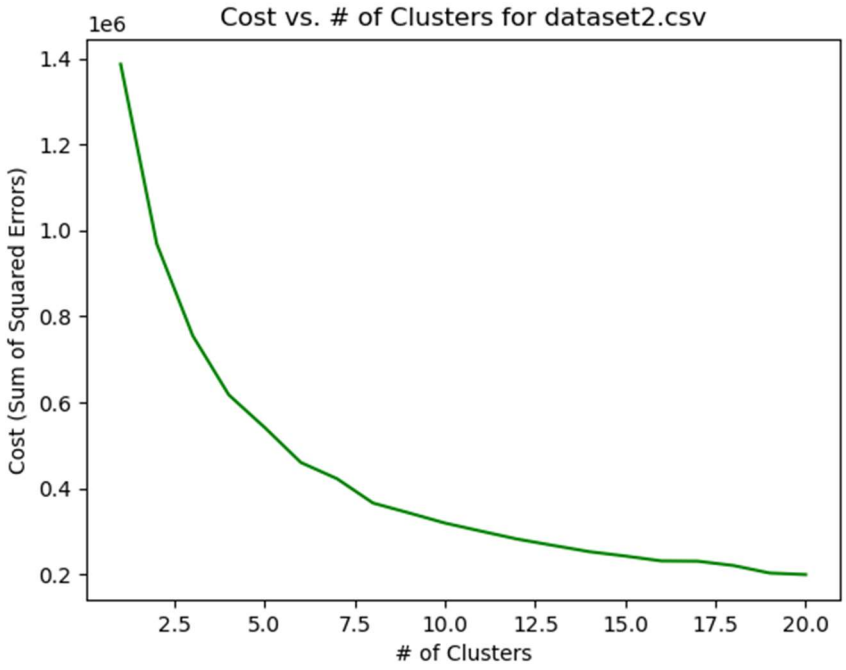

Figure 13: Cost Graph for 3D K-means

After producing the cost graph, The K value was varied from 2 to 20 to produce 3D scatter plots for data visualization purposes. Figures 14 through 18 show several of these scatter plots. Specifically, the plots for K values of 2, 6, 10, 14, and 18 are displayed.

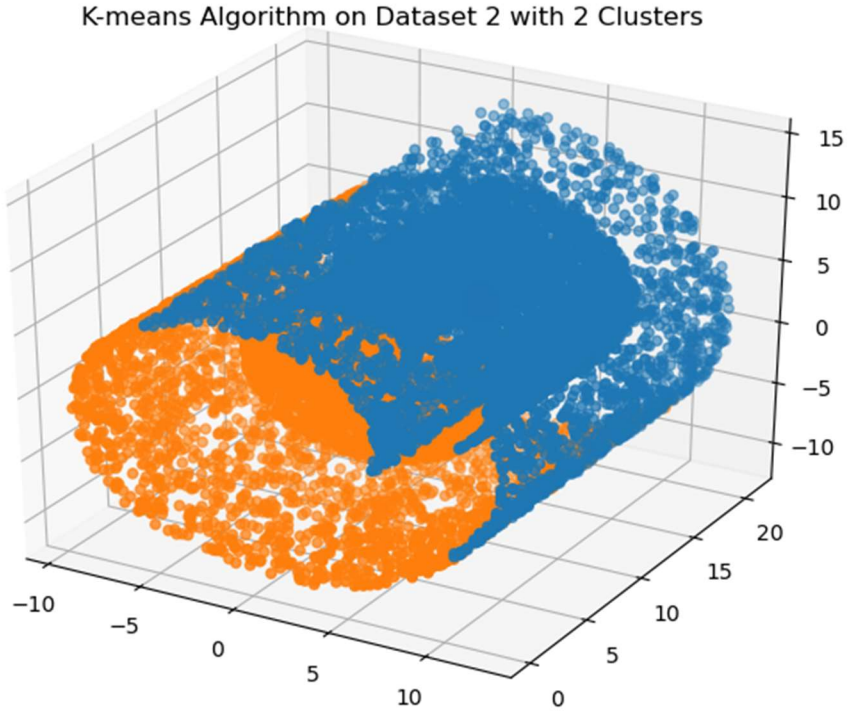

Figure 14: 3D K-means with K = 2

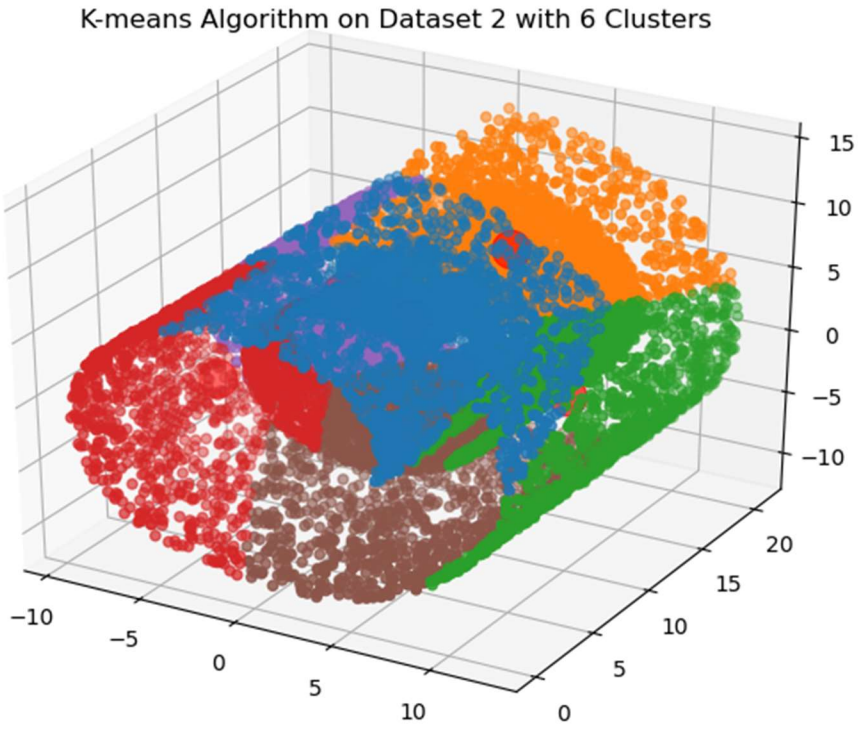

Figure 15: 3D K-means with K = 6

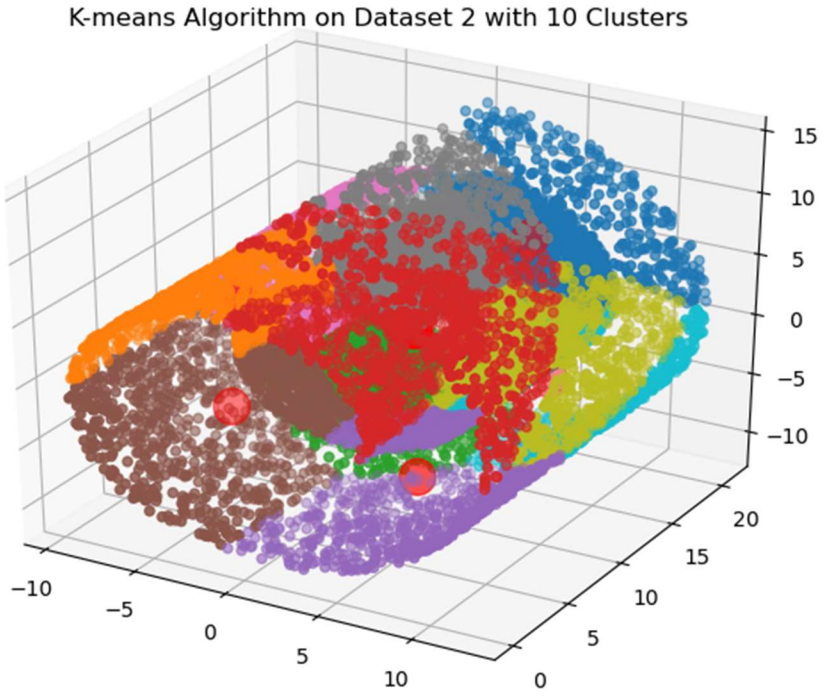

Figure 16: 3D K-means with K = 10

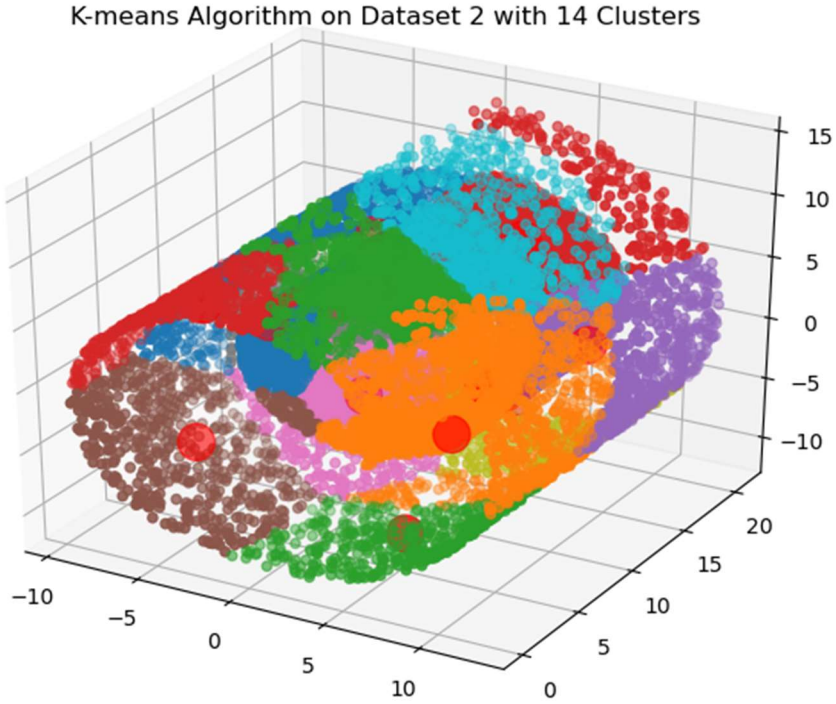

Figure 17: 3D K-means with K = 14

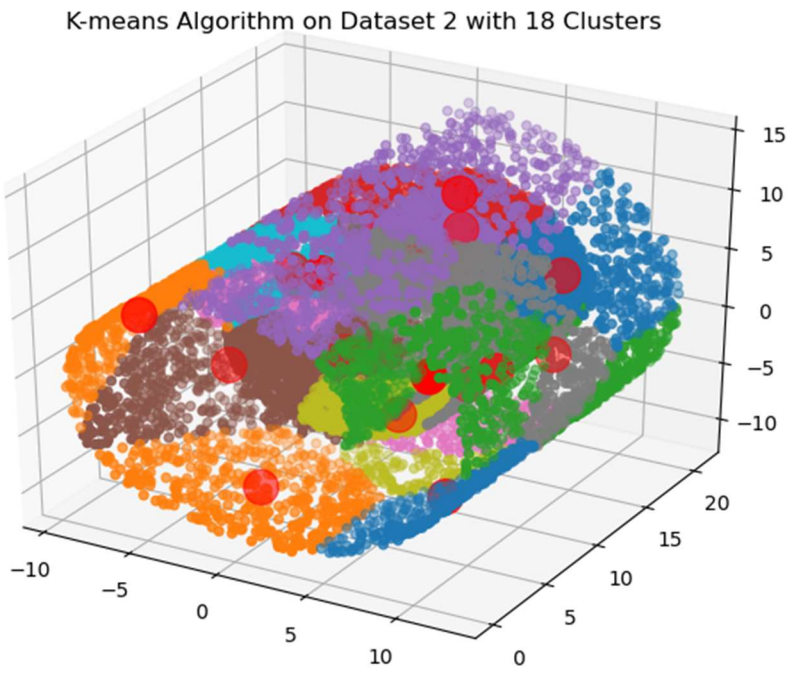

Figure 18: 3D K-means with K = 18

The second experimental initialization method for the three-dimensional data was k-means++ initialization. The K value (number of clusters) was again varied from 1 to 20 to produce the cost (sum of squared errors) graph shown in Figure 19. The optimal number of clusters for this experiment was discovered to be K = 10, which contrasts the previous results and produces a 0.914 completeness score.

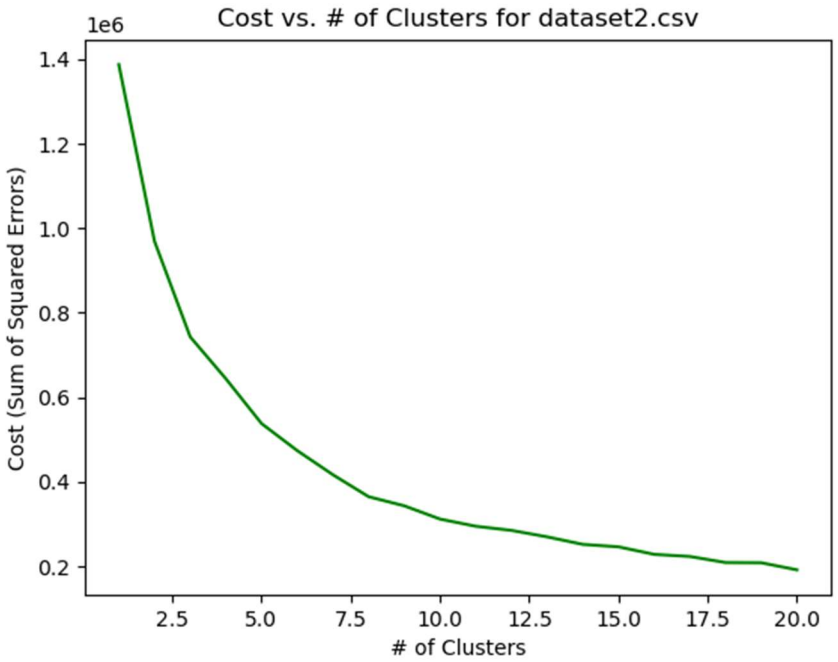

Figure 19: Cost Graph for 3D K-means++

Finally, The K value was again varied from 2 to 20 to produce 3D scatter plots for data visualization purposes. Figures 20 through 24 show the plots for K values of 2, 6, 10, 14, and 18.

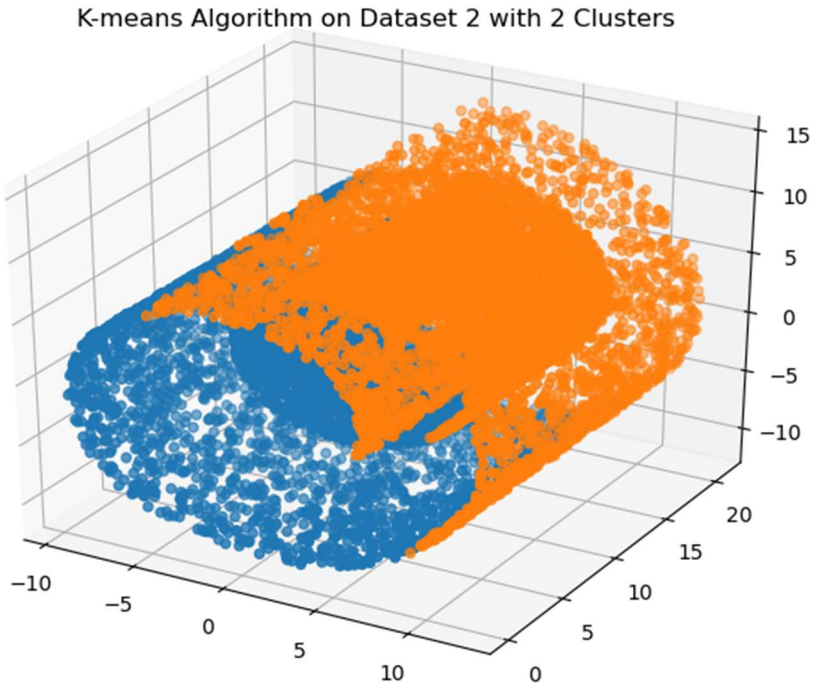

Figure 20: 3D K-means++ with K = 2

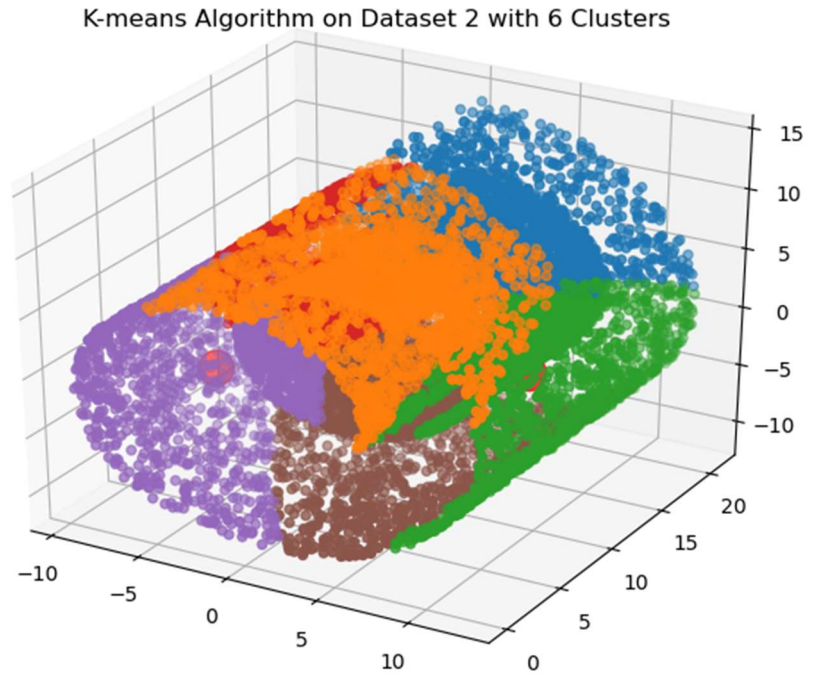

Figure 21: 3D K-means++ with K = 6

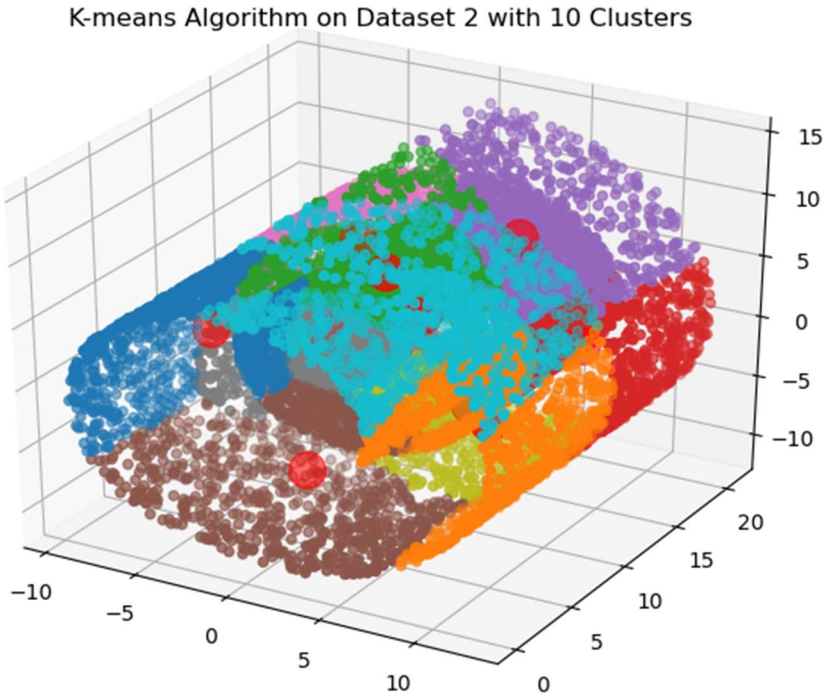

Figure 22: 3D K-means++ with K = 10

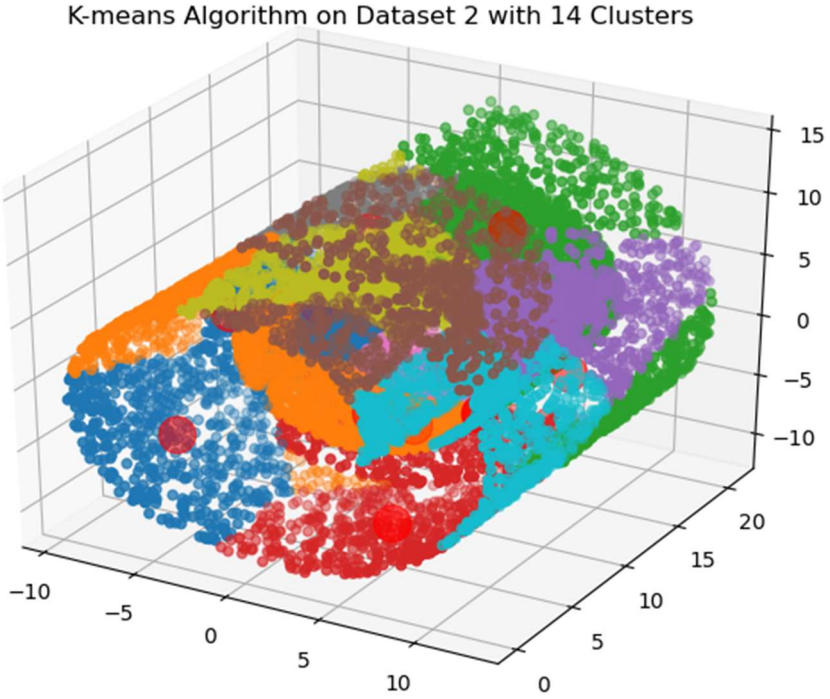

Figure 23: 3D K-means++ with K = 14

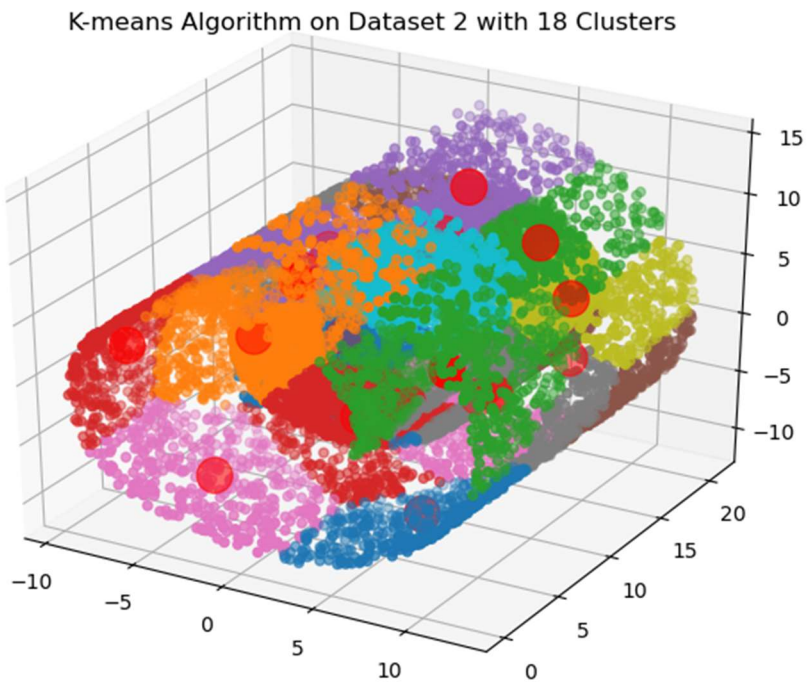

Figure 24: 3D K-means++ with K = 18

Following these experiments using the k-means algorithm, we can compare the completeness scores obtained using the optimal K values for each of the initialization method and n-dimensional data combinations. Uniform Random Initialization achieved scores of 0.898 and 0.936 on 2D and 3D data, respectively. On the other hand, K-means++ Initialization produced scores of 0.943 and 0.914 on 2D and 3D data, respectively. From this, it appears that the optimal combination is k-means++ initialization with two-dimensional data. Furthermore, we obtain the hypothesis that k-means++ is better suited for 2D data but can obtain high accuracy regardless of the data's dimension. Follow-up experiments would provide additional insight and potentially confirm or falsify this hypothesis. Additional tests for a future experiment could also involve higher dimensional data and a third initialization method.

2. Hierarchical Agglomerative Clustering Analysis

Hierarchical Agglomerative Clustering (HAC) was the second clustering method used on the two datasets. This method initially treats each data point as a cluster, successively merging clusters and making them larger (a "bottom-up" methodology), until each data point has been collected [7]. The resulting representation of the data points is a dendrogram. The implementation of HAC that was used was Scikit Learn's Agglomerative Clustering model with two linkage dissimilarity measures: single linkage and average linkage [8]. Additionally, Scipy's dendrogram implementation was used to visualize the data [9]. For each experiment, a cut will be taken from the dendrogram to determine the optimal number of clusters. Due to the lack of a commonly accepted method for cutting the dendrogram, we will cut along the longest visibly untouched area of the dendrogram. These cuts will be described using the closest y-axis values. Like the k-means experiments, the completeness score will be used for evaluation.

2.1 Two-Dimensional Data

The first set of HAC tests were run on the two-dimensional data available in dataset1.csv, which were generated by a Gaussian mixture model. Figure 25 shows the dendrogram for HAC using single linkage while Figure 26 shows the dendrogram for HAC using average linkage.

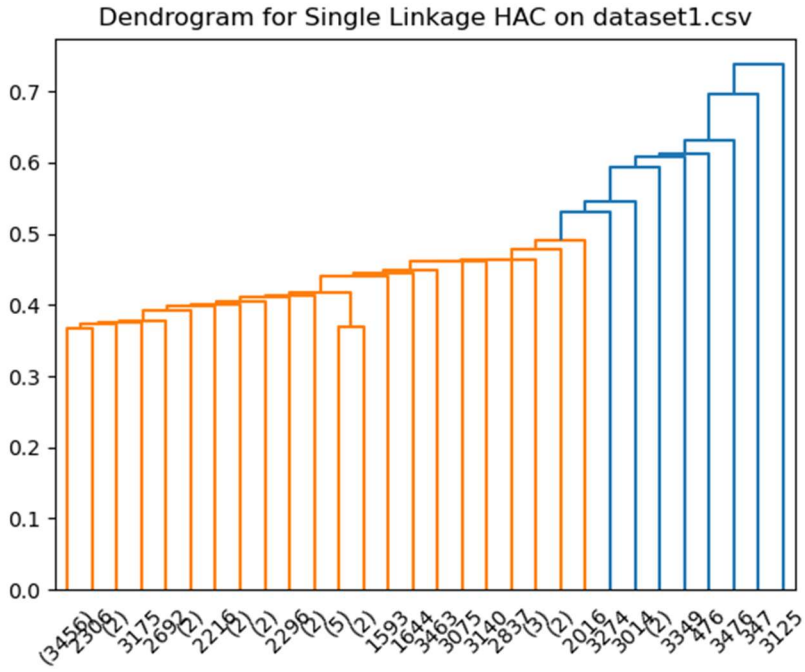

Figure 25: 2D Single Linkage HAC Dendrogram

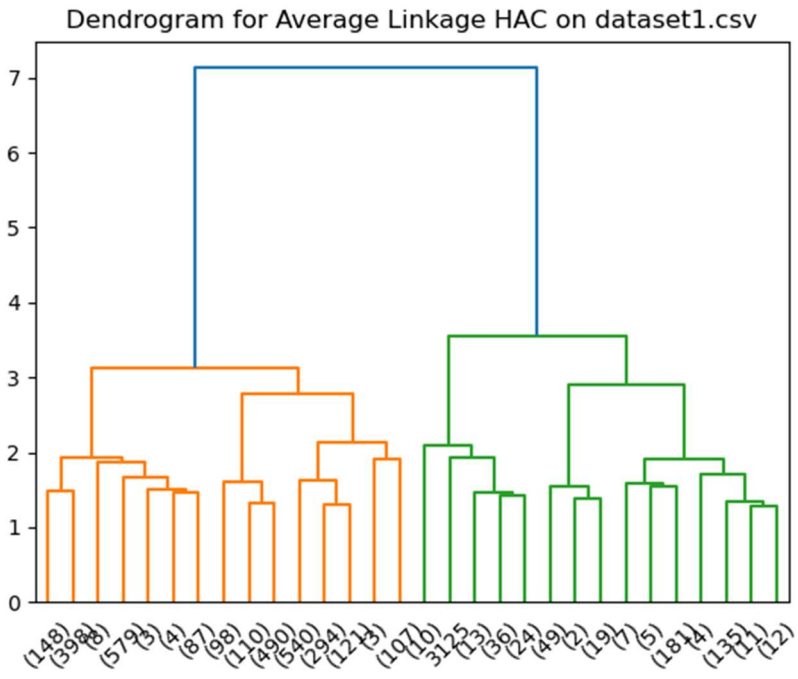

Figure 26: 2D Average Linkage HAC Dendrogram

We can cut each of these dendrograms to obtain K values. The single linkage dendrogram was cut at roughly 0.36 to obtain a K value of 28 while the average linkage dendrogram was cut at roughly 3.8 to obtain a K value of 3. The scatter plot for single linkage HAC with K = 28, shown in Figure 27, achieved a score of 0.966. The scatter plot for average linkage HAC with K = 3, displayed in Figure 28, achieved a score of 0.927.

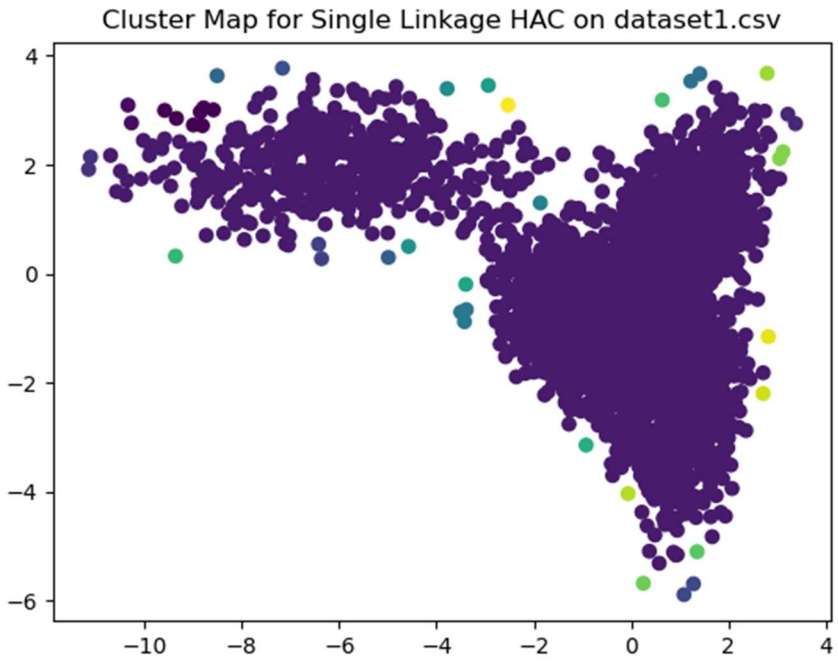

Figure 27: 2D Single Linkage HAC Scatter Plot

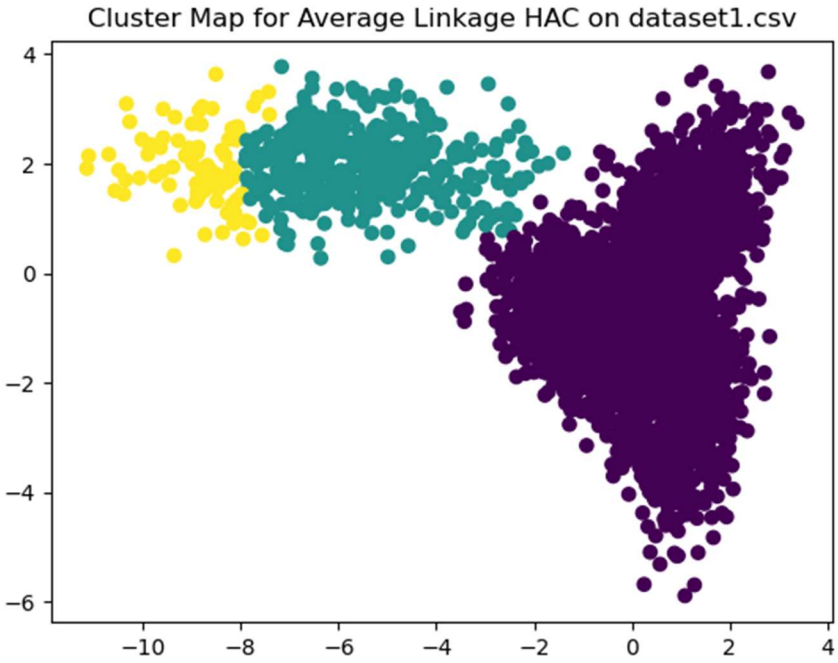

Figure 28: 2D Average Linkage HAC Scatter Plot

2.2 Three-Dimensional Data

The second set of HAC tests were run on the three-dimensional data available in dataset1.csv. Figure 29 shows the dendrogram for HAC using single linkage while Figure 30 shows the dendrogram for HAC using average linkage.

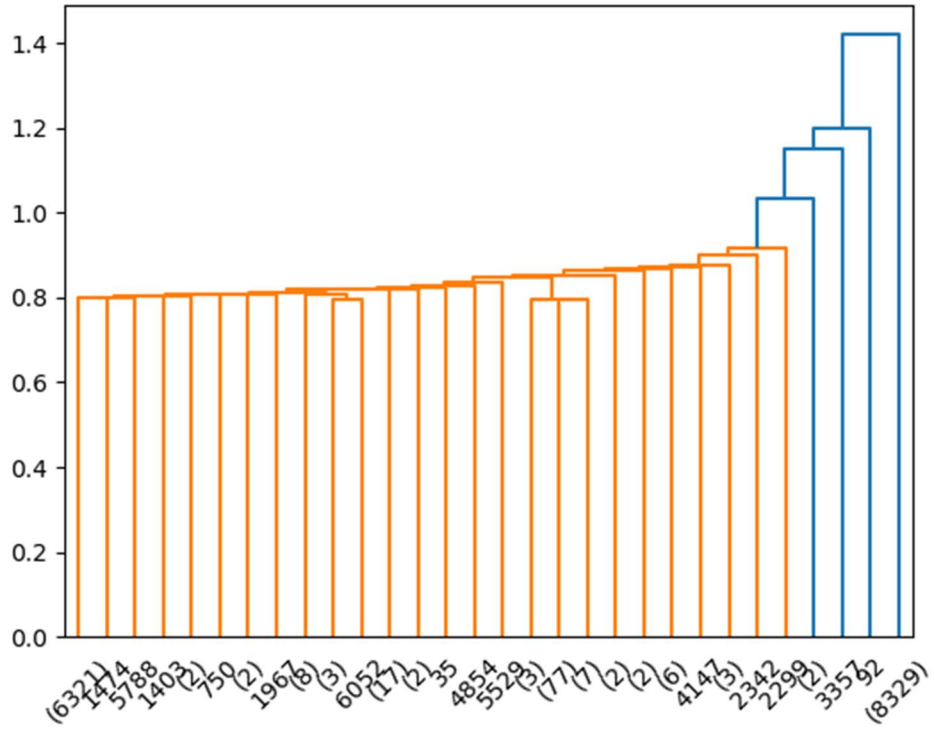

Figure 29: 3D Single Linkage HAC Dendrogram

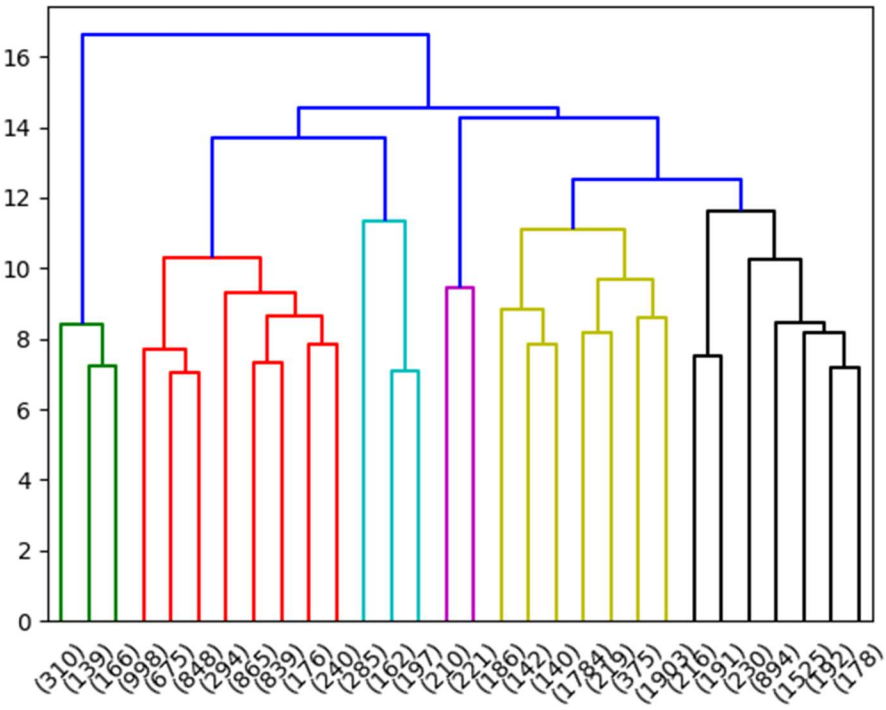

Figure 30: 3D Average Linkage HAC Dendrogram

We can cut each of these dendrograms to obtain K values. The single linkage dendrogram was cut at roughly 0.75 to obtain a K value of 28 while the average linkage dendrogram was cut at roughly 3.8 to obtain a K value of 26. The scatter plot for single linkage HAC with K = 28, shown in Figure 31, achieved a score of 0.959. The scatter plot for average linkage HAC with K = 26, displayed in Figure 32, achieved a score of 0.981.

Following these experiments using the HAC algorithm, we can compare the completeness scores obtained using the optimal K values for each of the linkage dissimilarity measures and n-dimensional data combinations. Single linkage HAC produced scores of 0.966 and 0.959 on the 2D and 3D data, respectively. Average linkage HAC resulted in scores of 0.927 and 0.981 on the 2D and 3D data, respectively. Evidently, average linkage on the 3D dataset produced the highest score among all experiments. Follow-up experiments could involve research into the ideal criteria for cutting a dendrogram, as a change in this experimental variable could greatly impact the test results.

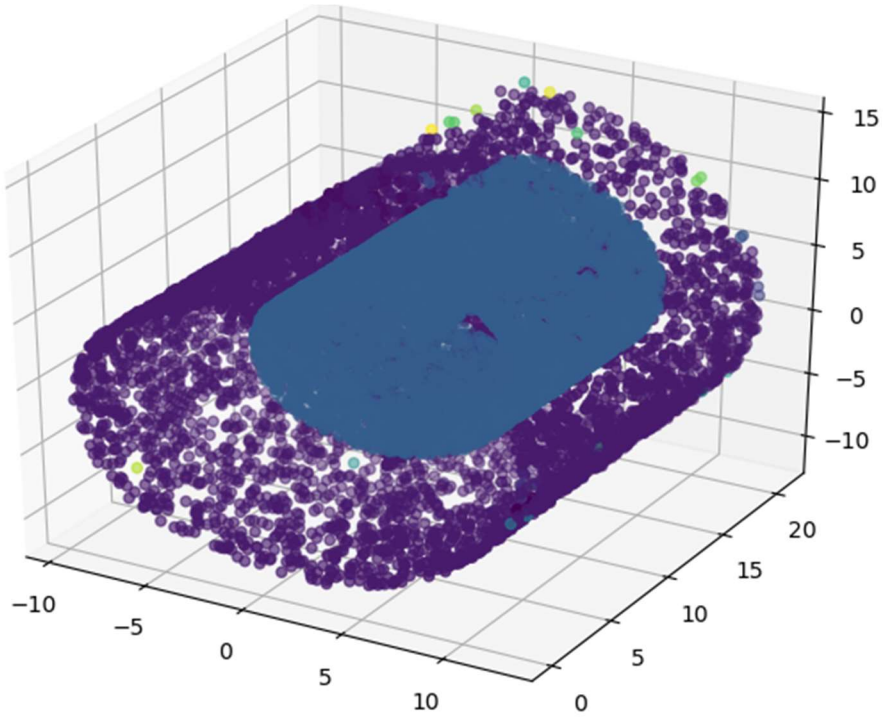

Figure 31: 3D Single Linkage HAC Scatter Plot

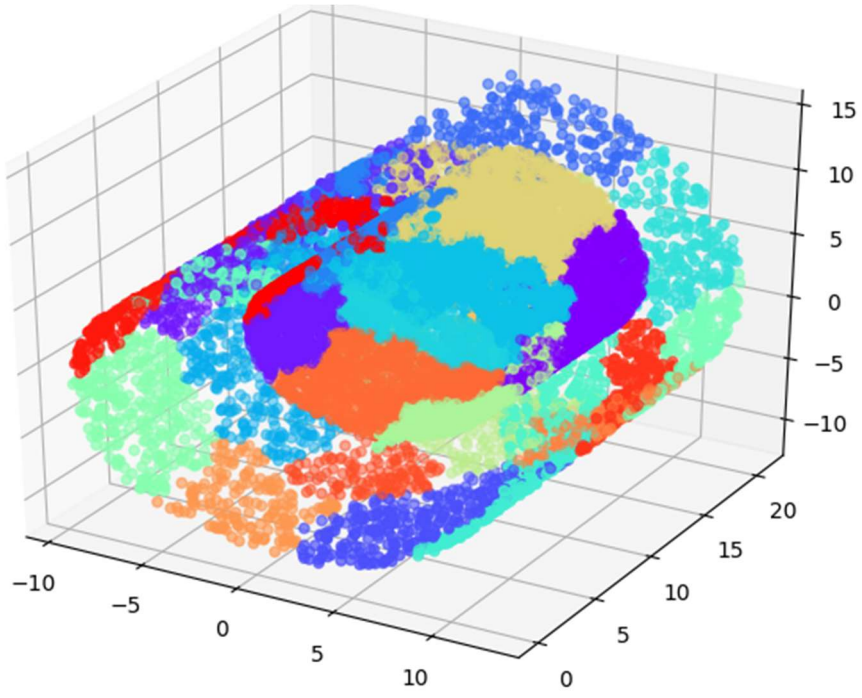

Figure 32: 3D Average Linkage HAC Scatter Plot