Joel Kerfoot
V00855134

Assignment 3 – SENG 474

Lloyd's Algorithm

The first part of the assignment was implementing Lloyd's algorithm (k-means) with uniform random initialization and k-means++ initialization.

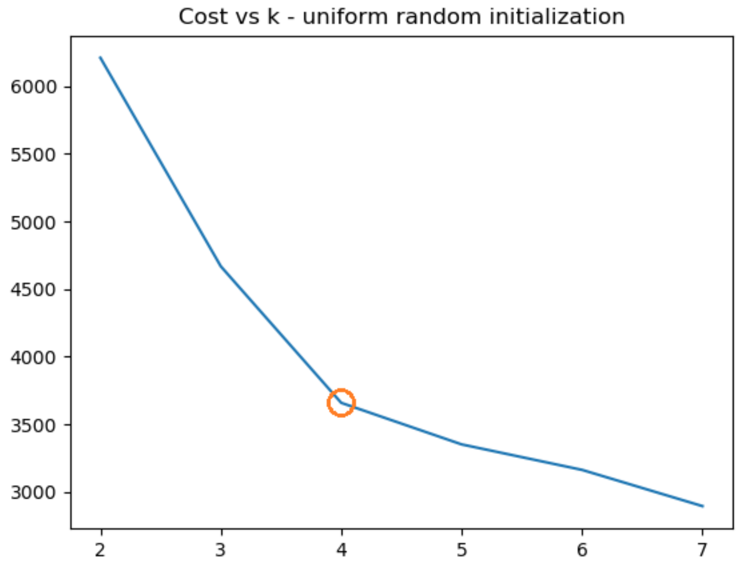
2D Data Set

Using the 2-dimensional data from dataset1.csv values of k ranging from 2 to 7 inclusively were used with both uniform random initialization and k-means++ initialization. The results of the experiments are shown below.

Uniform Random Initialization

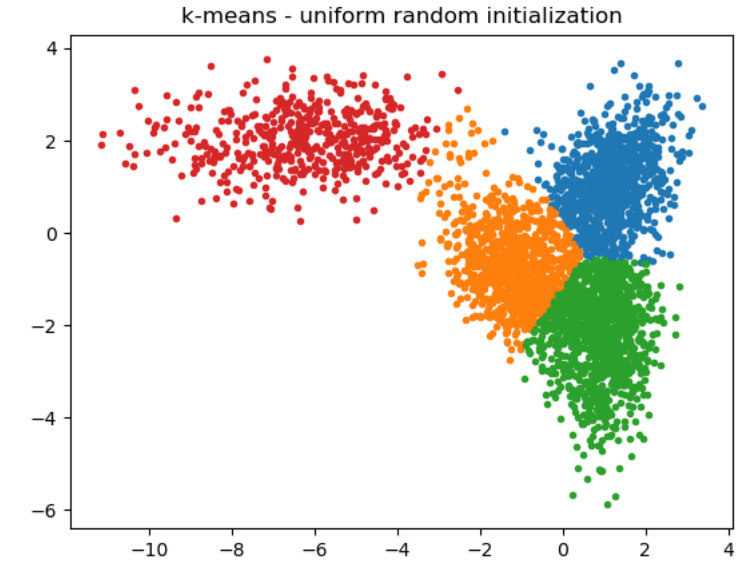
The cost of running Lloyd's algorithm with uniform random initialization is shown in figure 1 below. The cost for each k is calculated by taking the sum of squared errors from each point to its cluster center.

Figure 1 – Lloyds Algorithm Cost vs Number of Clusters



When k = 4 the cost stops decreasing as rapidly and is a good choice for the number of clusters. The scatter plot for k = 4 is shown below in figure 2.

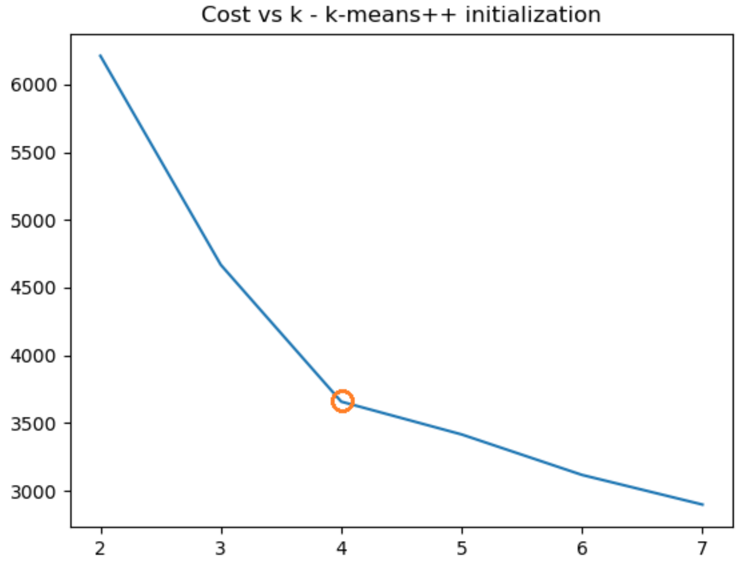
Figure 2 – Uniform Random Initialization Clustering k=4



k-means++ Initialization

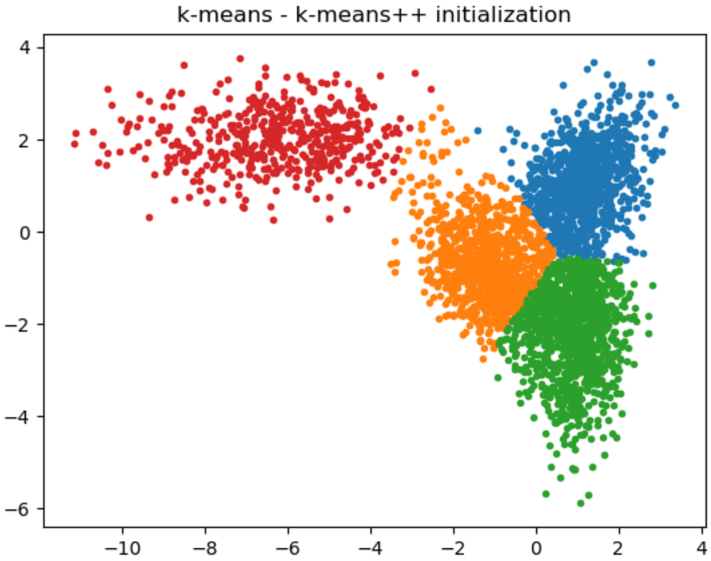
The cost of running Lloyd's algorithm with k-means++ initialization is shown in figure 3 below. Again, the cost for each k was calculated by taking the sum of squared errors from each point to its cluster center.

Figure 3 – Lloyds Algorithm Cost vs Number of Clusters



Using k-means++ initialization produces a similar cost curve to random initialization. K = 4 is a good choice because the cost stops decreasing rapidly. The scatter plot corresponding to k = 4 is shown below in figure 4.

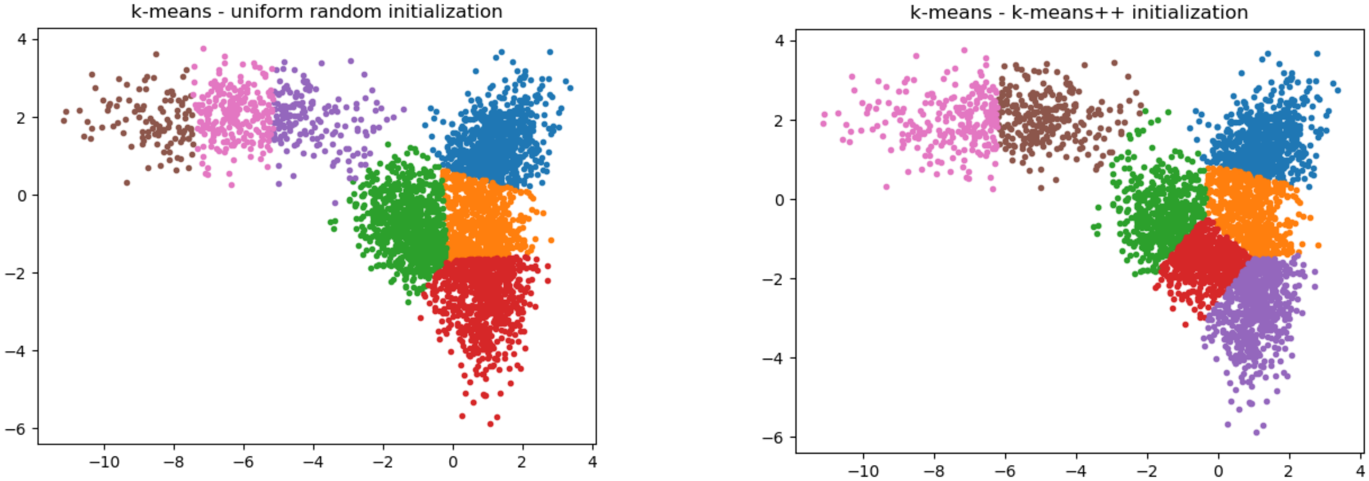
Figure 4 –k-means++ Initialization Clustering k=4



Observations

For values of k ranging from 2 to 6 uniform random initialization and k-means++ produced the same or very similar clusters. K = 7 was the first time there was a difference in the clustering. The different clusters are shown if figure 5.

Figure 5 – Uniform Random vs k-means++ Initialization Clustering k=7



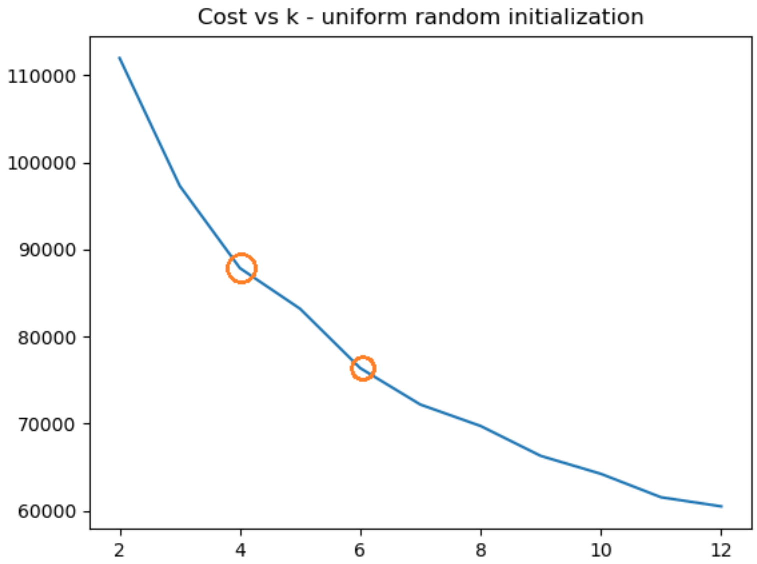
3D Data Set

Using the 3-dimensional data from dataset2.csv values of k ranging from 2 to 12 inclusively were used with both uniform random initialization and k-means++ initialization. The results of the experiments are shown below.

Uniform Random Initialization

The cost of running Lloyd's algorithm with uniform random initialization is shown in figure 6. The cost is calculated the exact same way as the 2-dimensional dataset by taking the sum of squared distances from each point to its cluster center.

Figure 6 – Lloyds Algorithm Cost vs Number of Clusters



When k = 4 or 6 the cost stops decreasing as rapidly. Either of these could be a good choice for the number of clusters. The scatter plot for k = 4 is shown below in figure 7 and k = 6 in figure 8.

Figure 7 – Uniform Random Initialization Clustering k=4

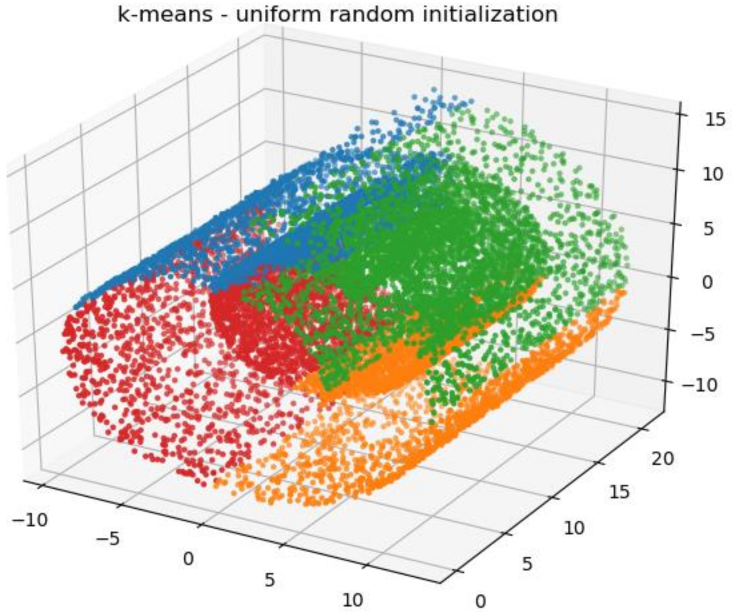
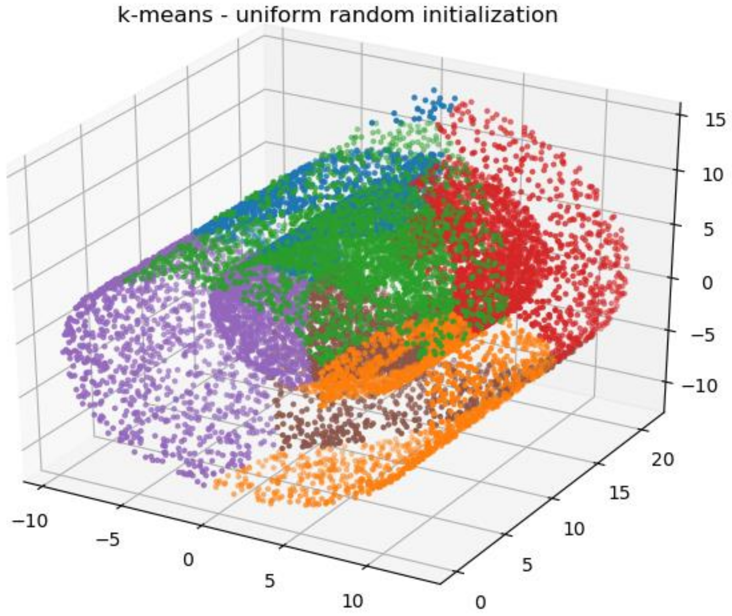


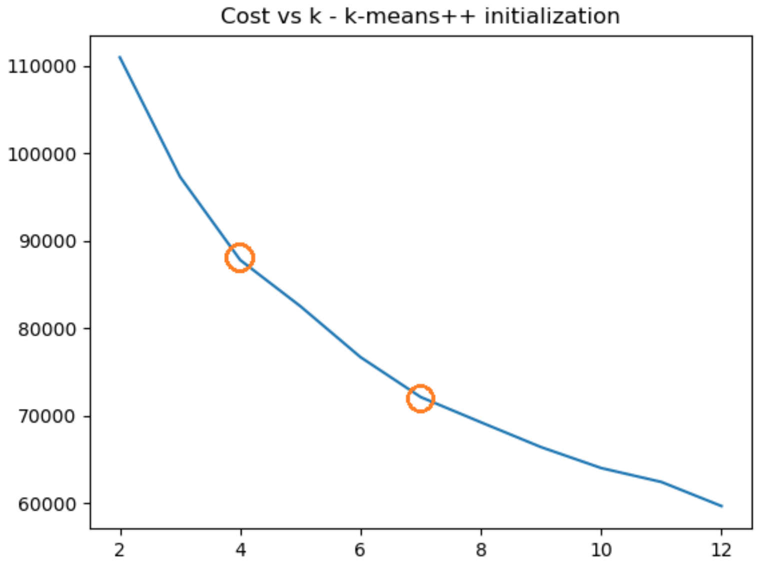
Figure 8 – Uniform Random Initialization Clustering k=6



k-means++ Initialization

The cost of running Lloyd’s algorithm with k-means++ initialization is shown in figure 9 below. Again, the cost for each k was calculated by taking the sum of squared errors from each point to its cluster center.

Figure 9 – Lloyds Algorithm Cost vs Number of Clusters



When using k-means++ initialization there isn’t as quite as noticeable decrease in the cost curve. Some values that look promising are k = 4 and k = 7 which are shown in figures 10 and 11 respectively.

Figure 10 – k-means++ Initialization Clustering k=4

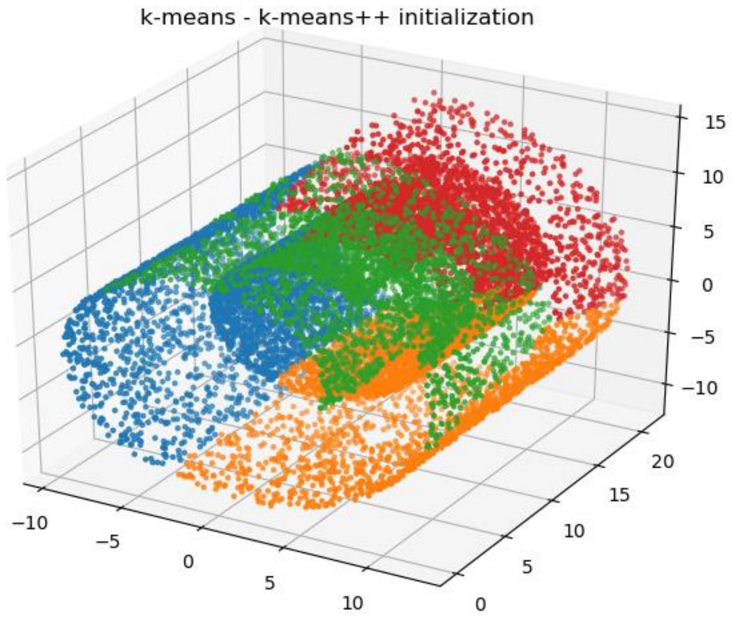
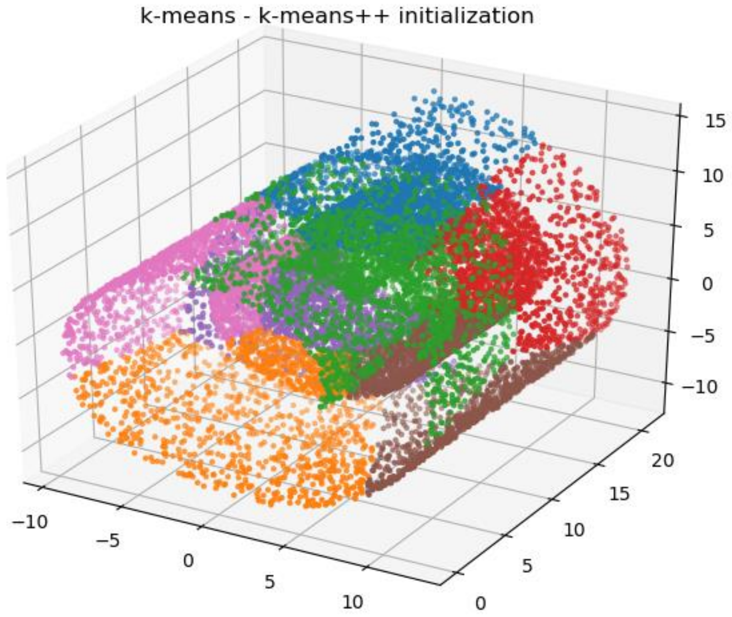


Figure 11 – k-means++ Initialization Clustering k=7



Observations

By observing the scatter plots for both initialization methods, it is clear that the clusters produced don’t necessary match the nature of the data structure. Both methods fail to cluster the outer shell from the inner glob of points.

Hierarchical Agglomerative Clustering

The second portion of the assignment was preforming agglomerative clustering on the same datasets. To accomplish this the dendrogram and AgglomerativeClustering libraries from sklearn were used [1,2]. The agglomerative clustering was done using both single and average links. The results from the experiments are shown below.

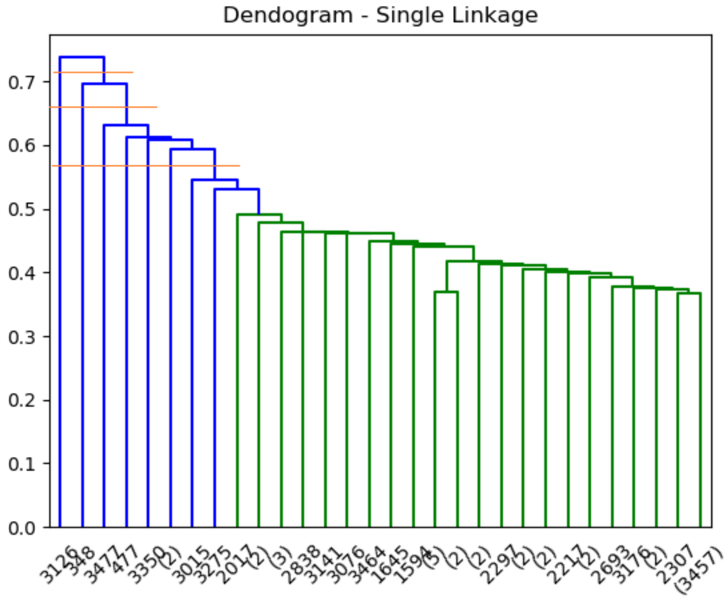
2D Data Set

For both single and average links dendrograms of the data were generated. The dendrogram were used to determine the values of k to use for agglomerative clustering.

Single Linkage

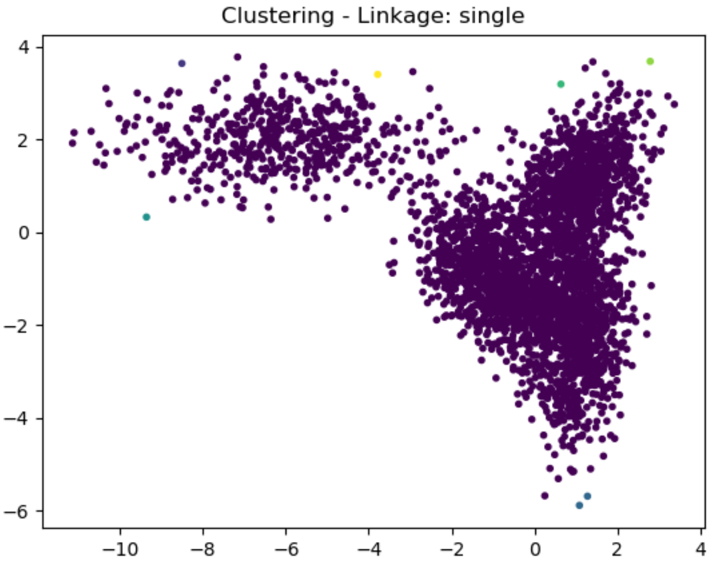
The dendrogram for single linkage is shown in figure 12.

Figure 12 – Dendrogram Single Linkage



Given that the height of a node indicates the dissimilarity between the 2 child nodes some reasonable cuts would be k = 2, 3, and 7. Using single linkage didn't produce sensible clustering so only the results from k = 7 is shown in figure 13.

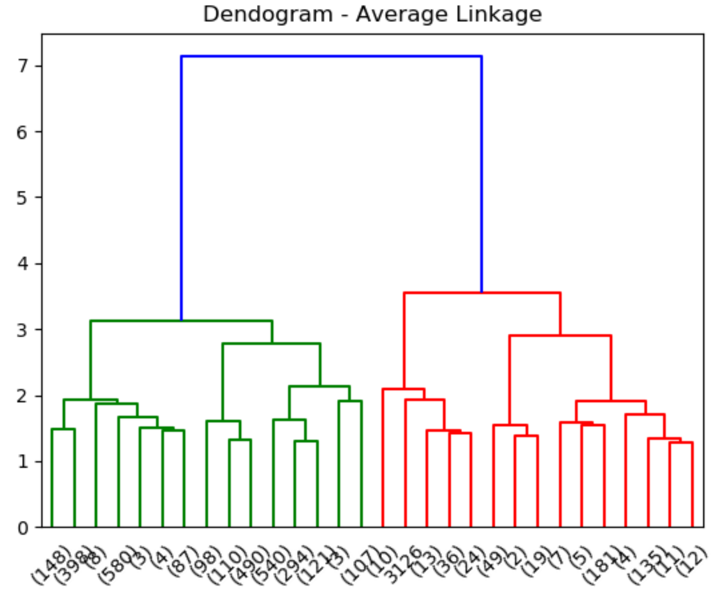
Figure 13 – Clustering Single Linkage k = 7



Average Linkage

The dendrogram for average linkage is shown in figure 14.

Figure 14 – Dendrogram Average Linkage



The largest dissimilarity is k = 2. Some other reasonable values include 4 and 6. The results are show in figures 15, 16, and 17.

Figure 15 – Clustering Average Linkage k = 2

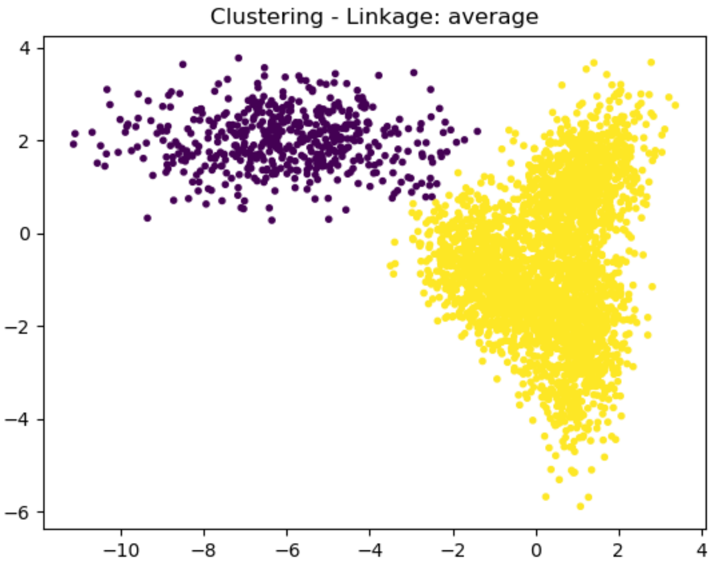


Figure 16 – Clustering Average Linkage k = 4

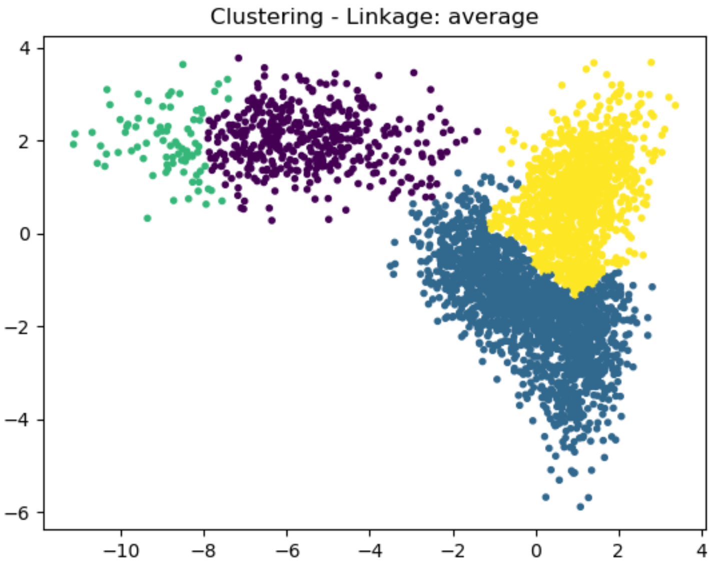
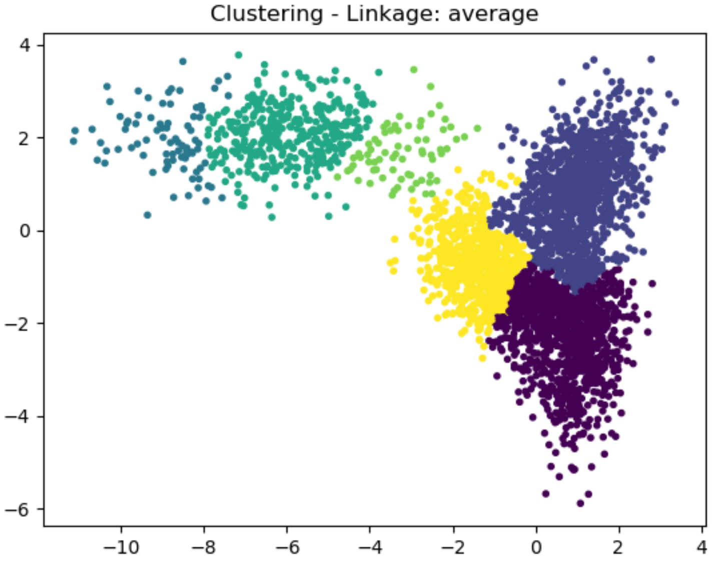


Figure 17 – Clustering Average Linkage k = 6



Observations

Single linkage didn’t work well with this dataset. However average linkage performed much better. Using average linkage produced clusters similar to Lloyd’s algorithm but the density of the clusters doesn’t affect the results as much as it did with k-means. Visually k = 2 produces the clustering that is most natural.

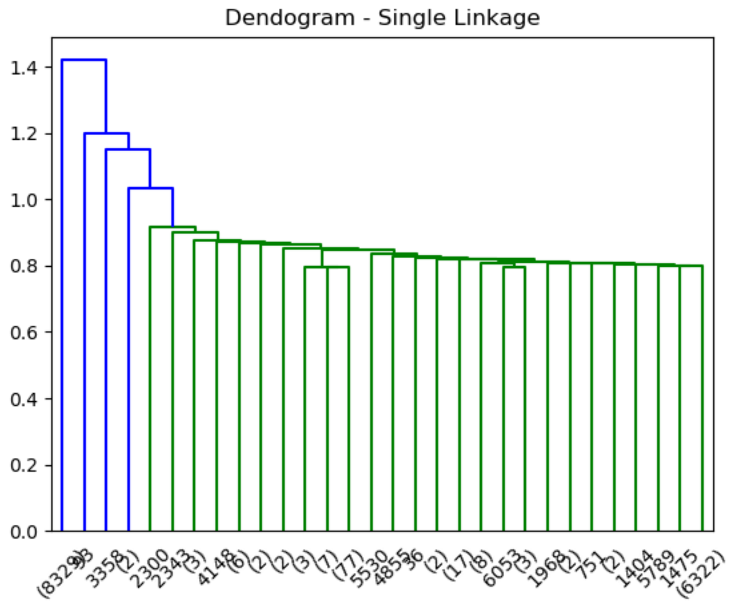
3D Data Set

The same method as the 2-dimensional data set was used for the 3-dimensional one. For both single and average links dendrograms of the data were generated. The dendrograms were used to determine the values of k to use for agglomerative clustering.

Single Linkage

The dendrogram for single linkage is shown in figure 18.

Figure 18 – Dendrogram Single Linkage



From the dendrogram some potential values for k include 2, 4, 5. The results of those clustering's are show in figures 19, 20, and 21.

Figure 19 – Clustering Single Linkage k = 2

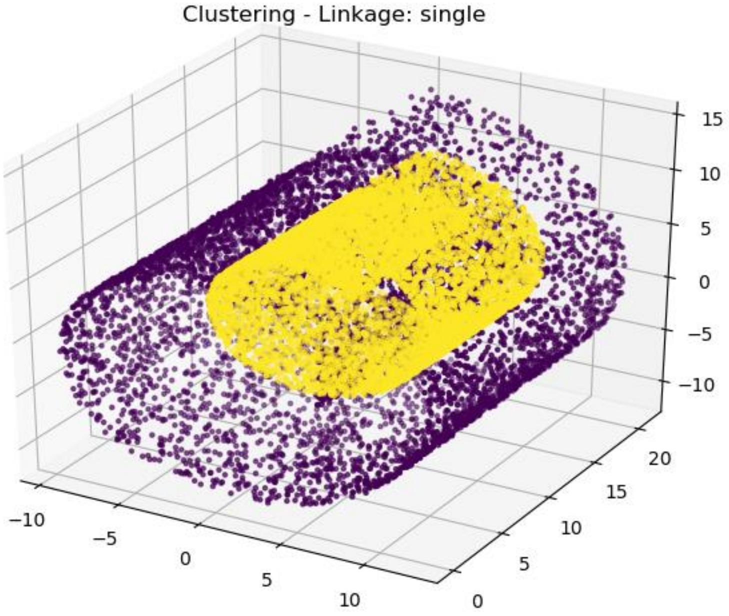


Figure 20 – Clustering Single Linkage k = 4

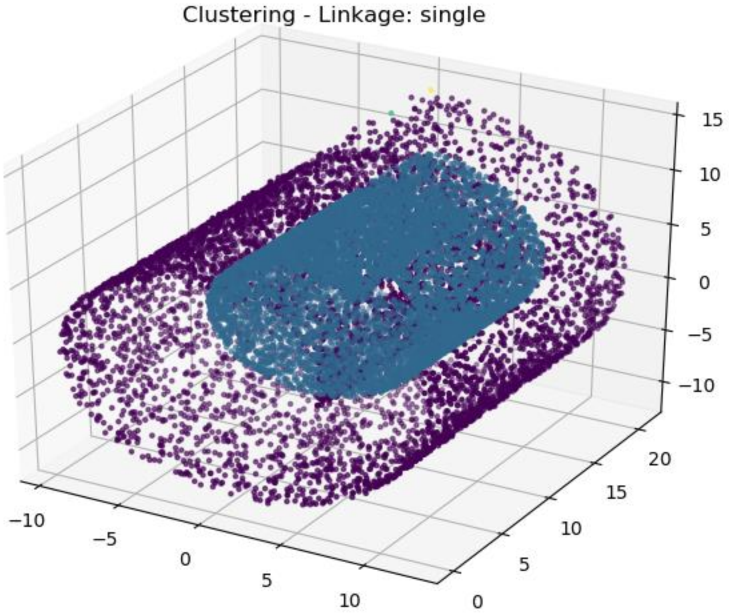
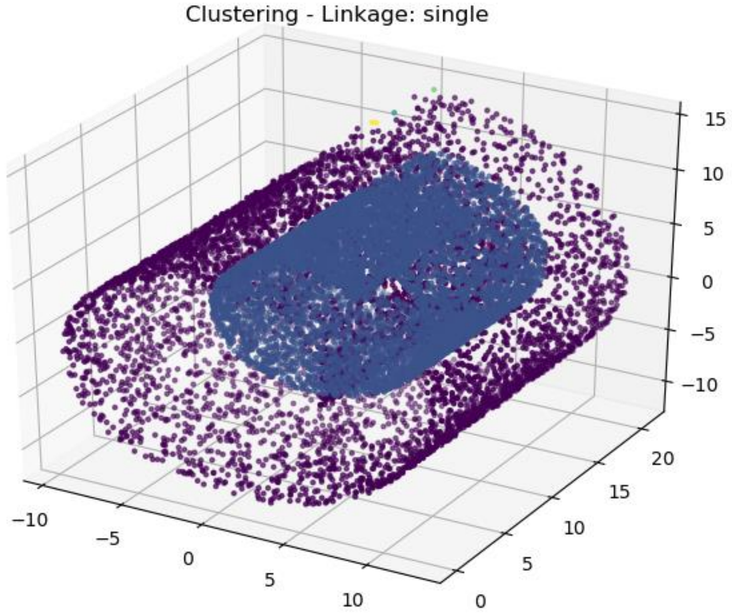


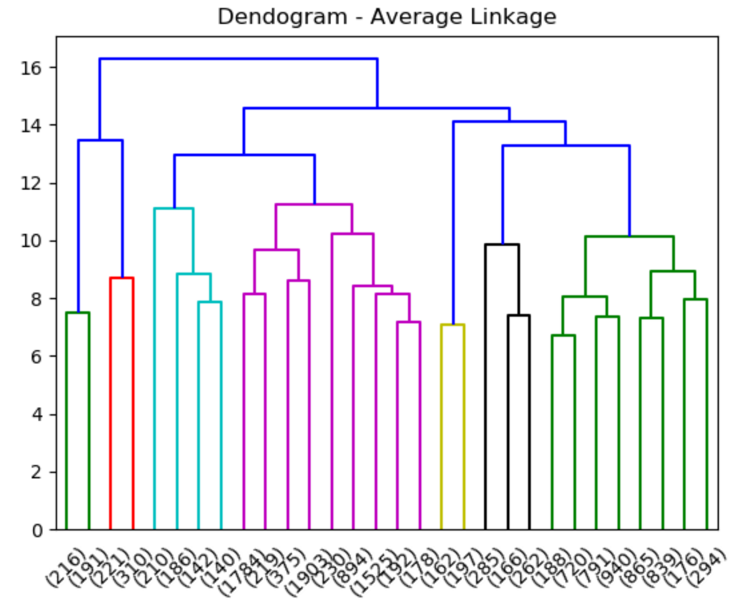
Figure 21 – Clustering Single Linkage k = 5



Average Linkage

The dendrogram for average linkage is shown in figure 22.

Figure 22 – Dendrogram Average Linkage



From the dendrogram some potential values for k include 2 and 7. The results of those clustering’s are show in figures 23, and 24.

Figure 23 – Clustering Average Linkage k = 2

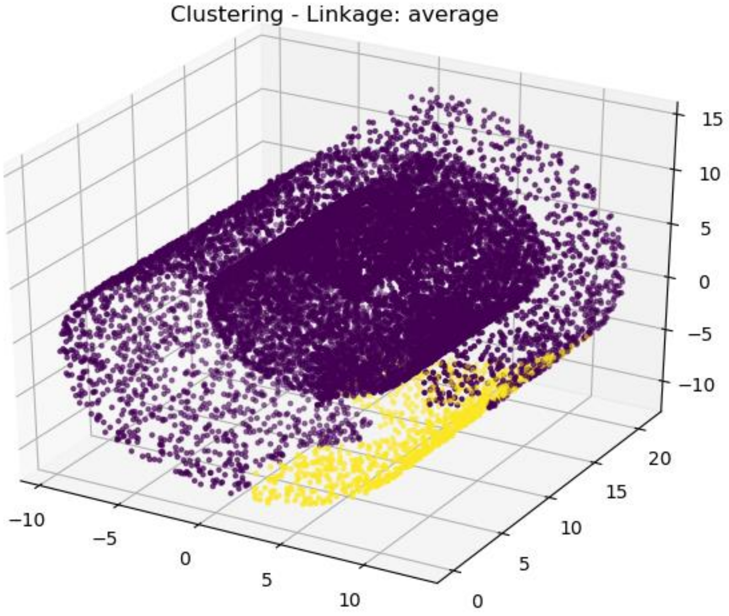
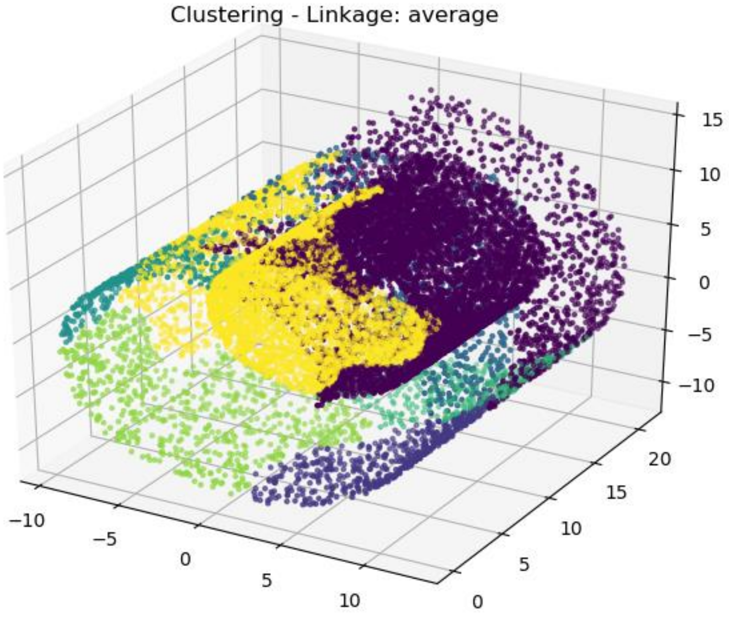


Figure 24 – Clustering Average Linkage k = 7



Observations

Single linkage performed the best on the 3-dimensional data set and was able to cluster the outer shell and inner glob separately. However, values of k higher than 2 didn’t add significant clusters and only classified single points at the top of the graph into separate clusters. Average linkage performed similarly to k-means in that the inner and outer groups were clustered together but the shapes of the clusters were distinctly different.

References

[1] https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html