Cameron Lindsay

V00927778

Undergraduate Student

SENG 474 – Assignment 3

1. Dataset 1

Dataset 1 is a collection of 3500 two-dimensional examples generated by a Gaussian mixture model (From Assignment 3 description). No formal preprocessing or normalization strategy was implemented.

1.1. Lloyd's algorithm (k-means)

Lloyd's algorithm is an algorithm that takes a dataset and separates data points within it into k clusters. The process is generally summarized as the initialization of k initial centroid data points that act as the mean center for new clusters, the assignment of the remaining data values to a cluster, and the updating of the new cluster center according to the mean value for the cluster data points. The assignment and updating stages are repeated until there are no changes in the centroids used [1]. Euclidean distance was used to calculate the distance between centroids and data points. Elbow plots to determine appropriate k values compared cost in the form of sum of squared error (SSE) formulated from squared Euclidean distance of data values from their associated centroid.

Two different initialization strategies were utilized in this analysis: Uniform Random Initialization (k-means) and k-means++ initialization(k-means++). Uniform Random Initialization randomly chooses 3 data points from the dataset as the initial centroids. K-means++ only has the first centroid randomly assigned, the remaining centroids are chosen according to the probability proportional to the squared distance of that example to its closest existing center [1]. While k-means++ has a higher initialization cost of O(log k) relative to the random counterpart of O(1), the initial centroids are more likely to be in optimal positions and thereby allowing the algorithm to converge faster and reduce overall runtime [1].

1.1.1. Uniform Random Initialization

To obtain an optimal k value, elbow plots observing cost (described above) to value k was performed using the k-means algorithm with random initialization. Values of k used ranged from 1 – 10 and an optimal k was determined based on a point where exponential decrease in cost (SSE) was no longer occurring. Graphically the k-value was chosen just before the curve converges toward 0.

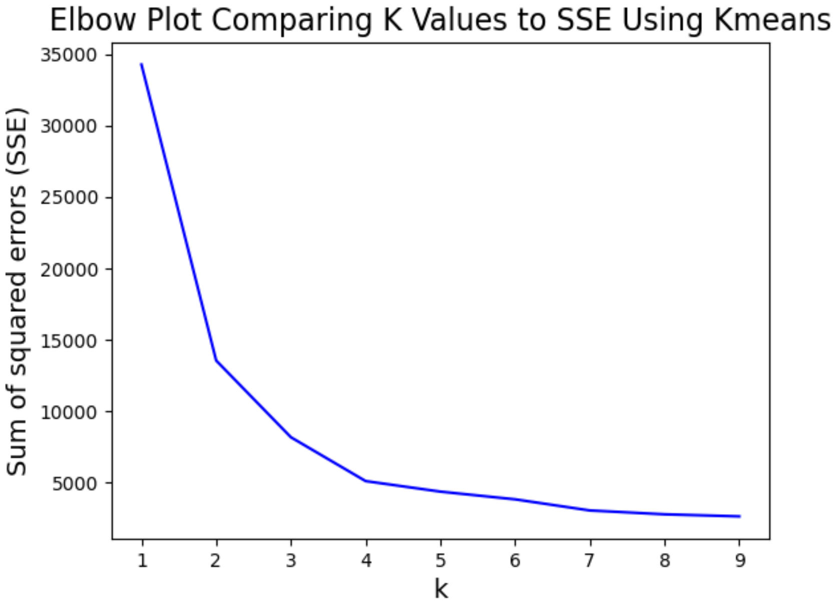

Figure 1.1.1.1. Elbow Plot using K-means with Random Initialization

From the elbow plot (Figure 1.1.1.1), a k-value of 4 was chosen and the associated model was subjected to a scatterplot (Figure 1.1.1.2).

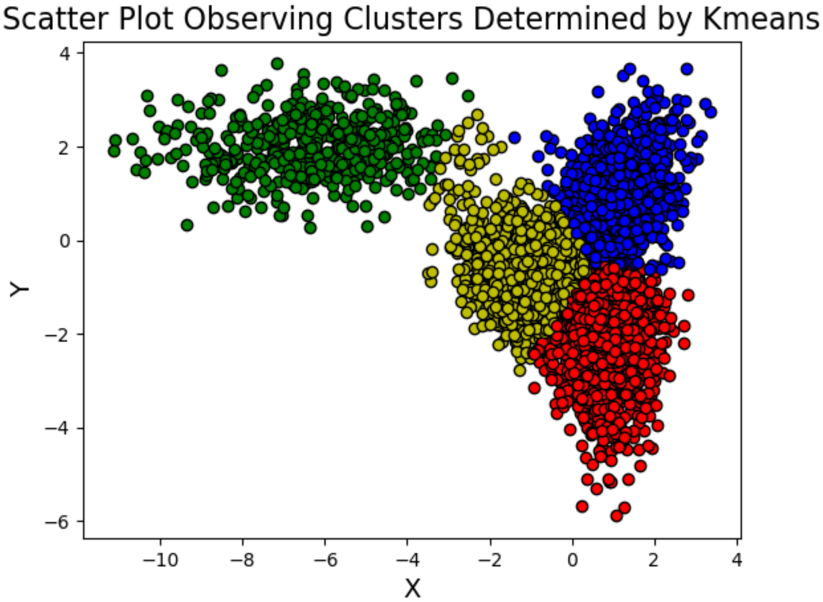

Figure 1.1.1.2. Scatterplot of Dataset 1 with Clustering via K-means with Random Initialization

Upon initial inspection of the data, 2 distinct clusters are apparent. The first is from X:(-10, -1) and Y:(0, 4), and the second from X:(-4, 4) and Y:(-6, 4). Interestingly, the 4 clusters suggested by the algorithm divides this second cluster into 3, and arguable intrude into the first cluster.

1.1.2. k-means++ Initialization

Like previous, an optimal k value was obtained using elbow plots observing SSE to value k was performed using the k-means++ algorithm. Values of k used ranged from 1 – 10 and an optimal k was determined based on a point where exponential decrease in SSE was no longer occurring. Graphically the k-value was chosen just before the curve converges toward 0.

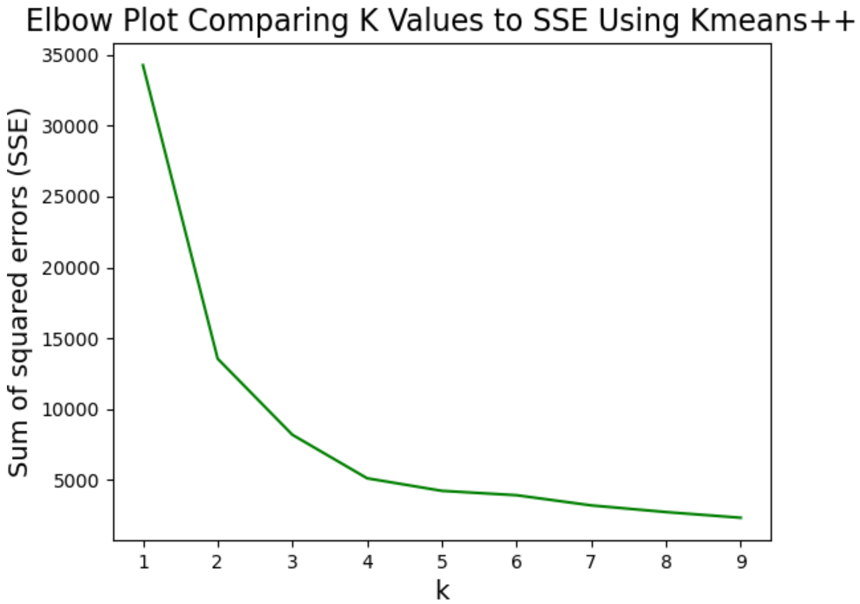

Figure 1.1.2.1. Elbow Plot using K-means with k-means++ Initialization

From the elbow plot (Figure 1.1.2.1), a k-value of 4 was chosen and the associated model was subjected to a scatterplot (Figure 1.1.2.2).

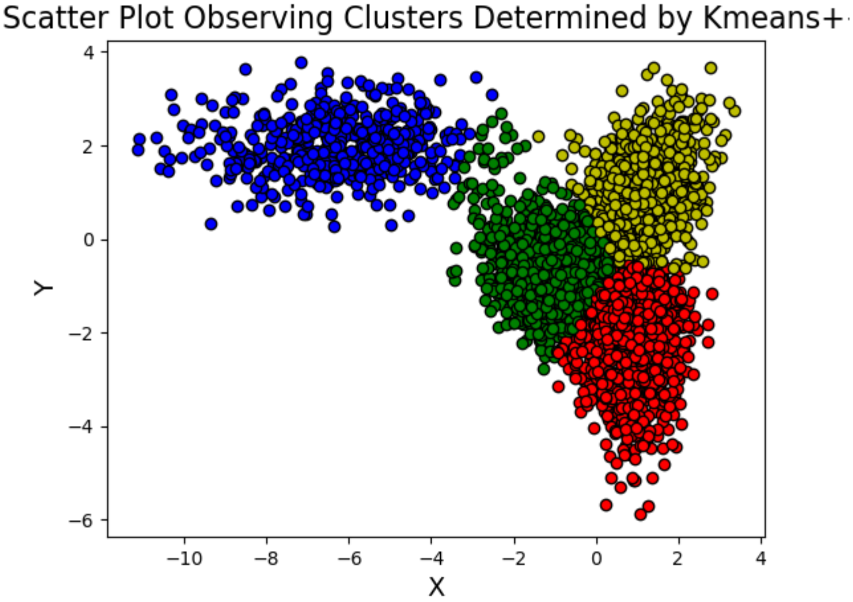

Figure 1.1.2.2. Scatterplot of Dataset 1 with Clustering via K-means with k-means++ Initialization

This graph is functionally equivalent to Lloyd's using random initialization. The difference in cluster colors suggests the centroids were initialized in difference orders from those in Figure 1.1.1.2. This is in line with the different strategies utilized. Initial inspection of the data, 2 distinct clusters are apparent with the first from X:(-10, -1) and Y:(0, 4), and the second from X:(-4, 4) and Y:(-6, 4). The 4 clusters suggested by the algorithm divides this second cluster into 3, with arguable intrusion into the first cluster.

1.2. Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering (HAC) models were created using sklearn's 'AgglomerativeClustering' function [2]. All HAC models used a distance threshold of 0 (distance_threshold = 0), no clusters to find (n_clusters = None), and Euclidean distance as a measure between points (affinity = 'euclidean'). Remaining parameters are set to their default values. Dendrograms utilized a level truncation strategy and a maximum depth of 5 (p=5). HAC involves the continual merging of comparable data points (clusters) until only a single cluster remains [1]. Optimal cluster values are chosen based on clustering iterations with the longest branch length. Two linkage strategies are compared for efficacy of clustering: Single linkage, where two clusters are merged according to whose two closest members have the smallest distance (smallest minimum pairwise distance), and Average linkage, where two clusters are merged according to the average distance between data points in the first and second cluster [1].

1.2.1. Single Linkage

To obtain an optimal number of clusters, a dendrogram was performed and cluster numbers were determined according to branch length. Graphically, clusters are mapped to different colors, with the cut-off cluster labelled blue.

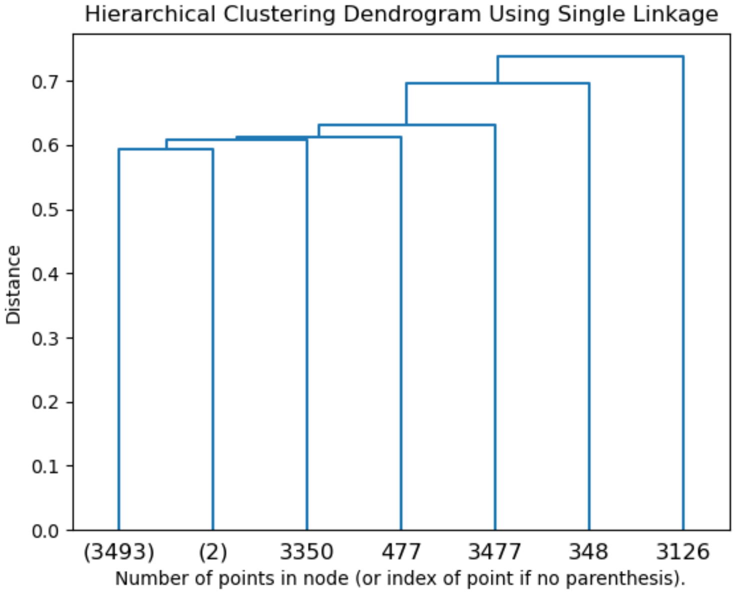

Figure 1.2.1.1. Dendrogram Using Single Linkage

From the dendrogram (Figure 1.2.1.1), no formal cut-off is apparent and a cluster number of 1 was chosen for further investigation via scatterplot (Figure 1.2.1.2).

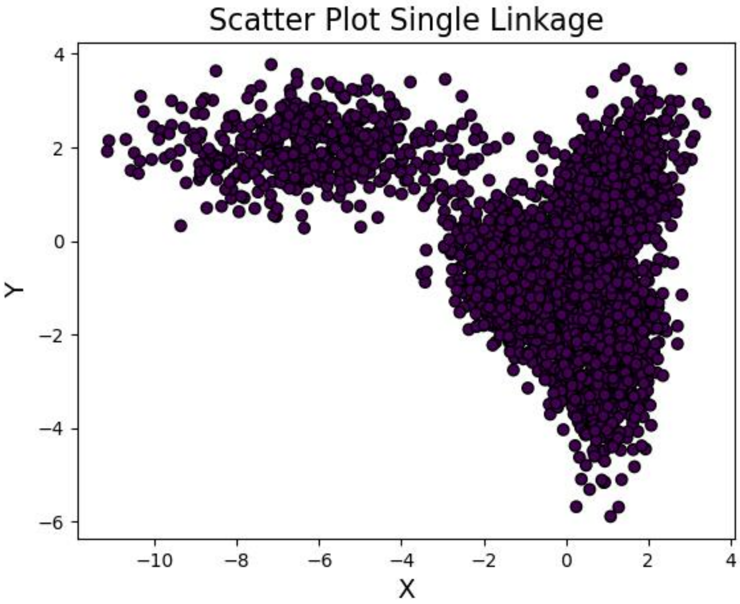

Figure 1.2.1.2. Scatterplot of Dataset 1 with Clustering via HAC using Single Linkage

Inspection of the data suggests 2 distinct clusters (described in detail in Section 1.1). As only one cluster was recommended from the model, this would suggest there is no separable differences in the data.

1.2.2. Average Linkage

To obtain an optimal number of clusters, a dendrogram was performed and cluster numbers were determined according to branch length. Graphically, clusters are mapped to different colors, with the cut-off cluster labelled blue.

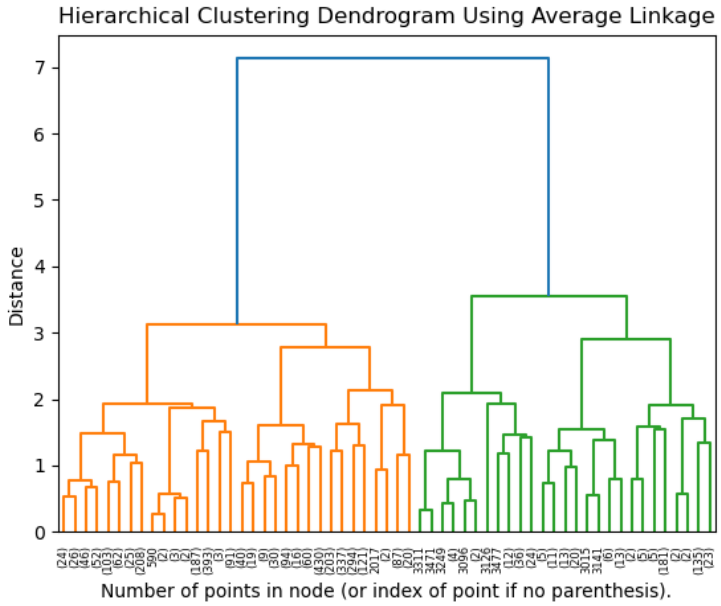

Figure 1.2.2.1. Dendrogram Using Average Linkage

From the dendrogram (Figure 1.2.2.1), a cut-off value above a distance of 4 is apparent and a cluster number of 2 was chosen for further investigation via scatterplot (Figure 1.2.2.2).

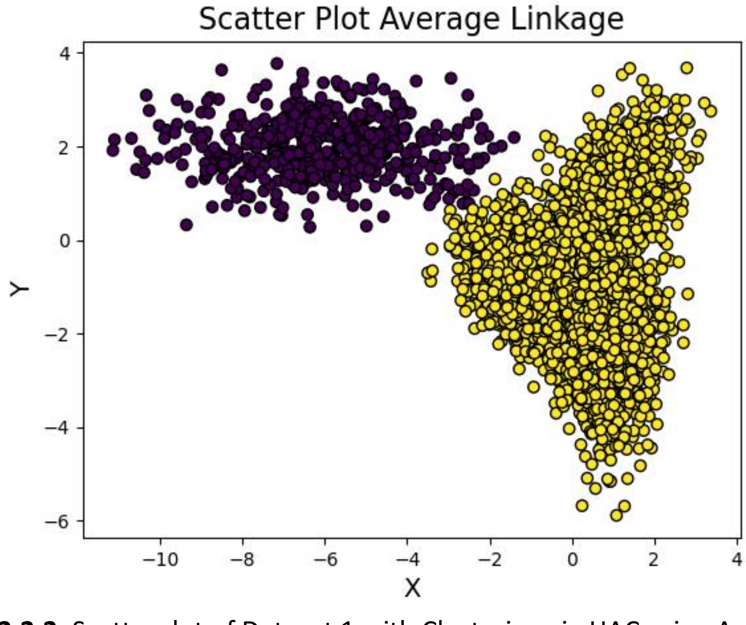

Figure 1.2.2.2. Scatterplot of Dataset 1 with Clustering via HAC using Average Linkage

Inspection of the data suggests 2 distinct clusters (described in detail in Section 1.1). The model displays these proposed clusters near ideally, with some potential overlap around X = -3 and Y = 1. These data points are contentious to begin with and would require further examination if used in a real-world context.

2. Dataset 2

Dataset 2 is a collection of 14,801 three-dimensional examples. No formal preprocessing or normalization strategy was implemented.

2.1. Lloyd's algorithm (k-means)

Like Dataset 1, Euclidean distance was used to calculate the distance between centroids and data points. Elbow plots to determine appropriate k values compared cost in the form of sum of squared error (SSE) formulated from squared Euclidean distance of data values from their associated centroid. The same initialization strategies were utilized in this analysis as Dataset 1: Uniform Random Initialization (k-means) and k-means++ initialization(k-means++).

2.1.1. Uniform Random Initialization

To obtain an optimal k value, elbow plots observing SSE to value k was performed using the k-means algorithm with random initialization. Values of k used ranged from 1 – 20 and an optimal k was determined based on a point where exponential decrease in SSE was no longer occurring. Graphically the k-value was chosen just before the curve converges toward 0.

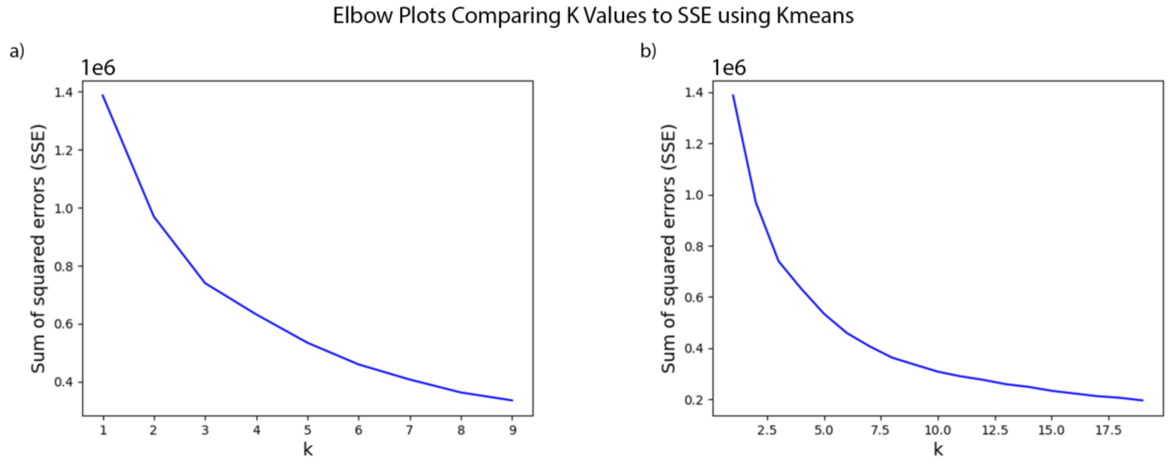

Figure 2.1.1.1. Elbow Plot using K-means with Random Initialization

Initially, elbow plots using k-values from 1 to 10 (Figure 2.1.1.1.a) were observed. Unfortunately, no convergence was readily apparent, and the range was increased to 20 (Figure 2.1.1.1.b). As convergence was apparent from the elbow plot between 8 and 10. Erring on the side of caution, a k-value of 8 was chosen and the associated model was subjected to a 3D scatterplot (Figure 2.1.1.2).

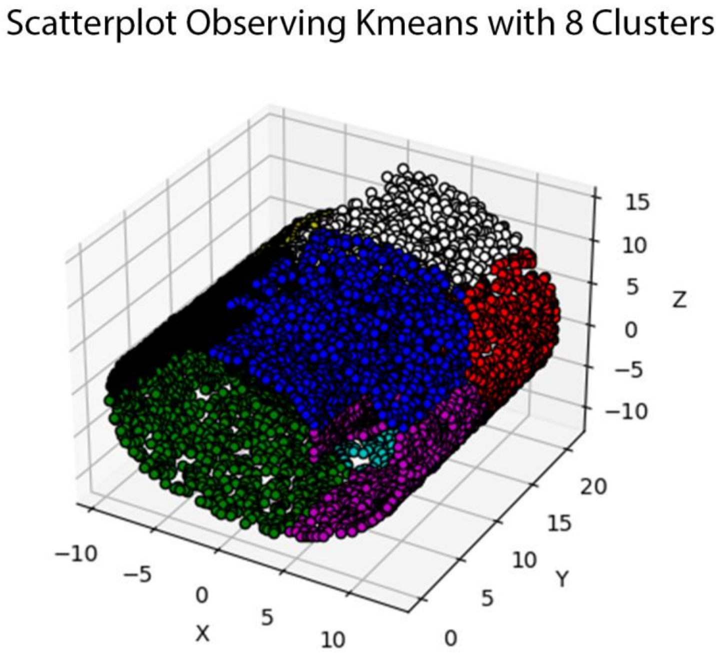

Figure 2.1.1.2. Scatterplot of Dataset 2 with Clustering via K-means with Random Initialization
Please note: Additional angles of the data were not able to be obtained due to hardware issues.

Upon initial inspection of the data, 2 distinct clusters are apparent. The first is from X:(-10, 10), Y:(0, 25), and Z:(-10, 15) and the second from X:(-7, 7), Y:(12, 25) , and Z:(-10, 10). From k-means, clusters used separate the data inappropriately, as the outer larger cluster and smaller inner cluster share several data points within clusters.

2.1.2. k-means++ Initialization

To obtain an optimal k value, elbow plots observing SSE to value k was performed using the k-means algorithm with random initialization. Values of k used ranged from 1 – 20 and an optimal k was determined based on a point where exponential decrease in SSE was no longer occurring. Graphically the k-value was chosen just before the curve converges toward 0.

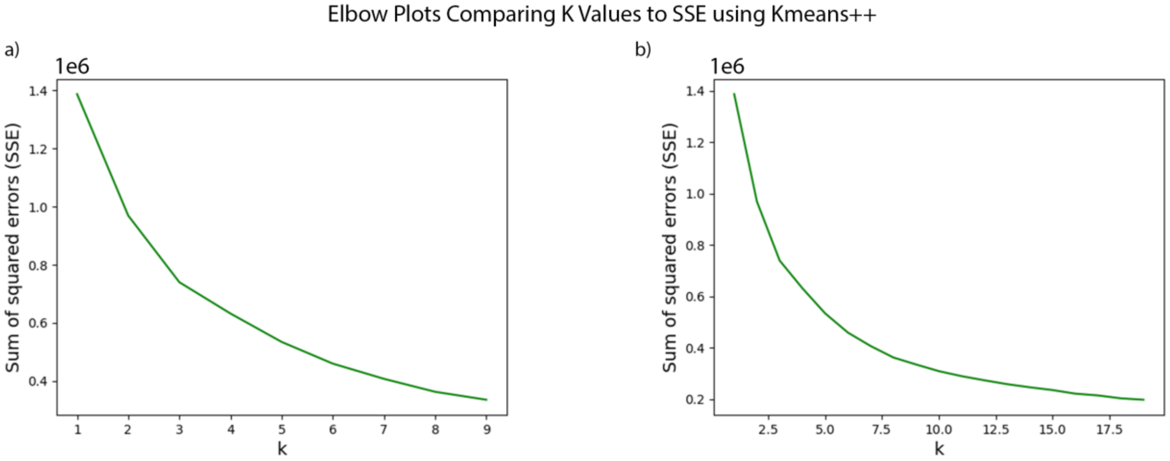

Figure 2.1.2.1. Elbow Plot using K-means++

Initially, elbow plots using k-values from 1 to 10 (Figure 2.1.1.1.a) were observed. Unfortunately, no convergence was readily apparent, and the range was increased to 20 (Figure 2.1.1.1.b). As convergence was apparent from the elbow plot between 8 and 10. Erring on the side of caution, a k-value of 8 was chosen and the associated model was subjected to a 3D scatterplot (Figure 2.1.1.2).

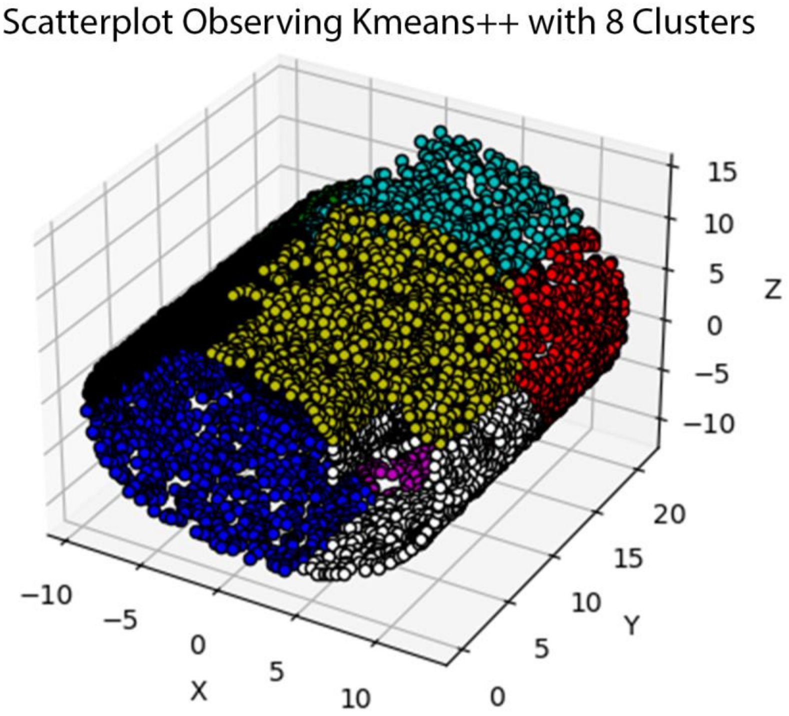

Figure 2.1.2.2. Scatterplot of Dataset 2 with Clustering via K-means++
Please note: Additional angles of the data were not able to be obtained due to hardware issues.

This graph is functionally equivalent to Lloyd's using random initialization. The difference in cluster colors suggests the centroids were initialized in difference orders from those in Figure 2.1.1.2. This is in line with the different strategies utilized. Upon initial inspection of the data, 2 distinct clusters are apparent. The first is from X:(-10, 10), Y:(0, 25), and Z:(-10, 15) and the second from X:(-7, 7), Y:(12, 25) , and Z:(-10, 10). From k-means, clusters used separate the data inappropriately, as the outer larger cluster and smaller inner cluster share several data points within clusters.

2.2. Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering (HAC) models were created using sklearn's 'AgglomerativeClustering' function [2]. All HAC models used a distance threshold of 0 (distance_threshold = 0), no clusters to find (n_clusters = None), and Euclidean distance as a measure between points (affinity = 'euclidean'). Remaining parameters are set to their default values. Dendrograms utilized a level truncation strategy and a maximum depth of 5 (p=5). Optimal cluster values are chosen based on clustering iterations with the longest branch length. Two linkage strategies are compared for efficacy of clustering: Single linkage and Average linkage.

2.2.1. Single Linkage

To obtain an optimal number of clusters, a dendrogram was performed and cluster numbers were determined according to branch length. Graphically, clusters are mapped to different colors, with the cut-off cluster labelled blue.

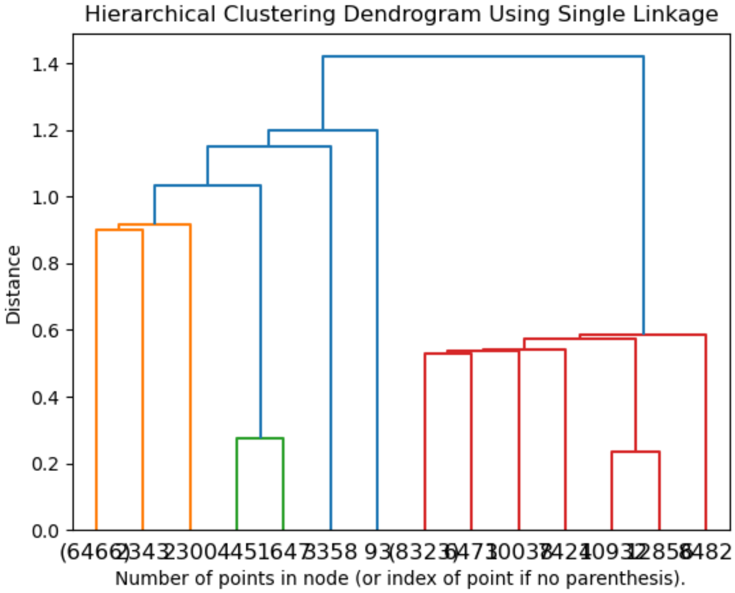

Figure 2.2.1.1. Dendrogram Using Single Linkage

From the dendrogram (Figure 2.2.1.1), a cut-off value above 1 is apparent and a cluster number of 3 was chosen for further investigation via scatterplot (Figure 2.2.1.2).

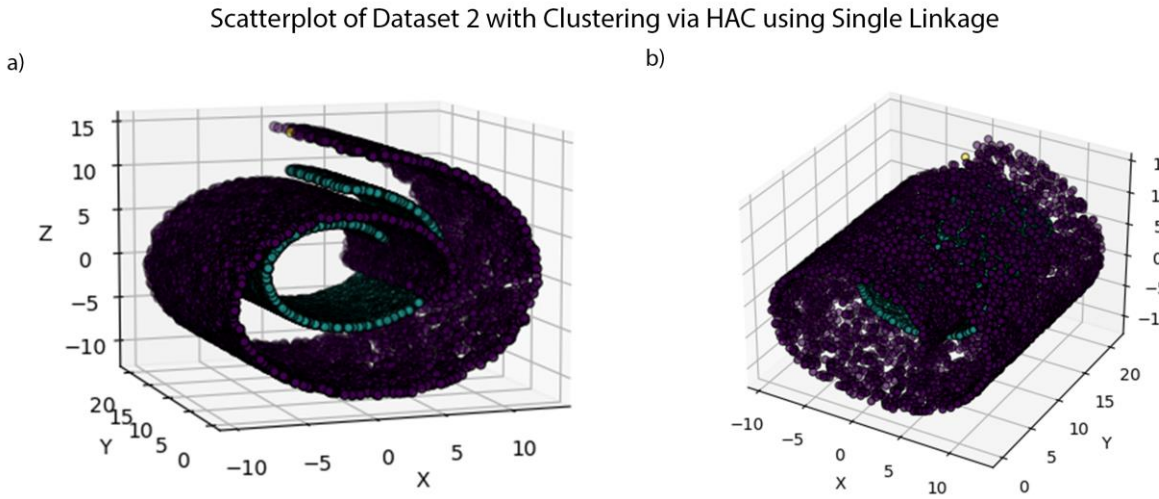

Figure 2.2.1.2. Scatterplot of Dataset 2 with Clustering via HAC using Single Linkage

Inspection of the data suggests 2 distinct clusters (described in detail in Section 2.1). The model displays these proposed clusters near ideally; however, there is a small cluster (yellow in color) around X = 0, Y = 20, and Z = 13 that arguably should not exist. These data points are contentious to begin with and would require further examination if used in a real-world context.

2.2.2. Average Linkage

To obtain an optimal number of clusters, a dendrogram was performed and cluster numbers were determined according to branch length. Graphically, clusters are mapped to different colors, with the cut-off cluster labelled blue.

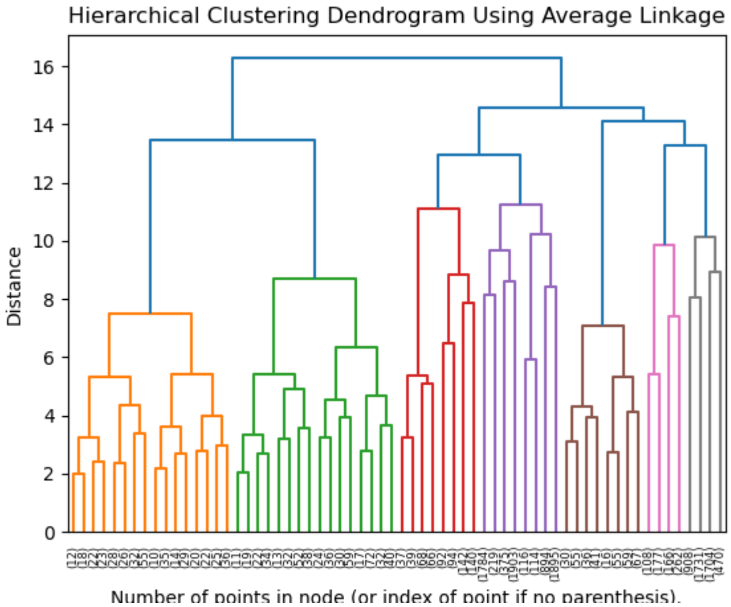

Figure 2.2.2.1. Dendrogram Using Average Linkage

From the dendrogram (Figure 2.2.2.1), a cut-off value above 1 is apparent and a cluster number of 7 was chosen for further investigation via scatterplot (Figure 2.2.2.2).

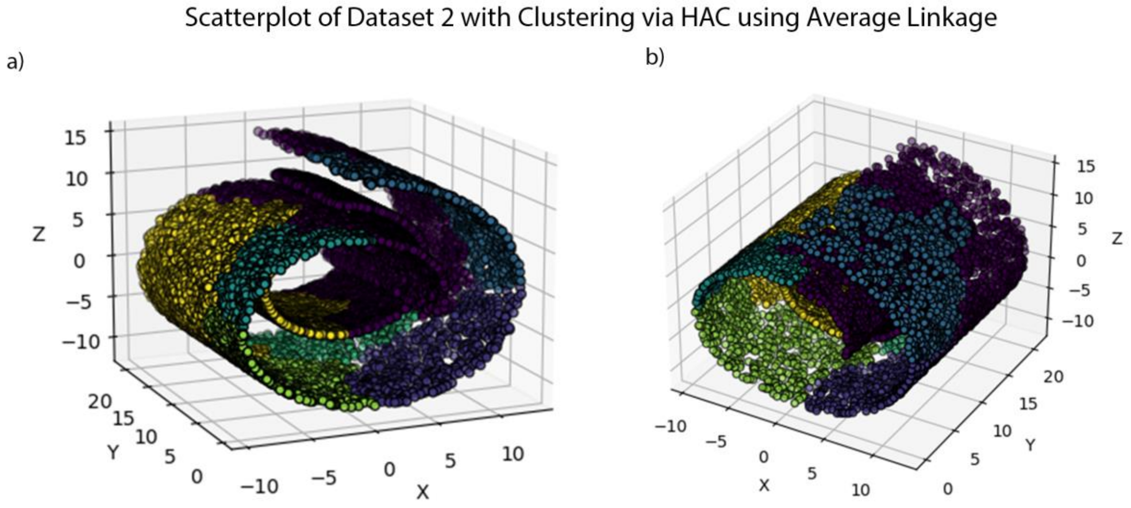

Figure 2.2.2.2. Scatterplot of Dataset 2 with Clustering via HAC using Average Linkage

Once again, inspection of the data suggests 2 distinct clusters (described in detail in Section 2.1). However, the average linkage sensitivity displays abnormal clusters comparable to the ones observed in the k means implementation. The increased distance allowable between clusters results in overlap between the distinct clusters (Section 2.1).

3. Model Comparison and Concluding Remarks

3.1 Comparing Random Initialization to k-means++

To compare the two k-means initialization strategies, a 95% confidence interval comparing squared error (SE) was performed and plotted via boxplot representation (Figure 3.1.1).

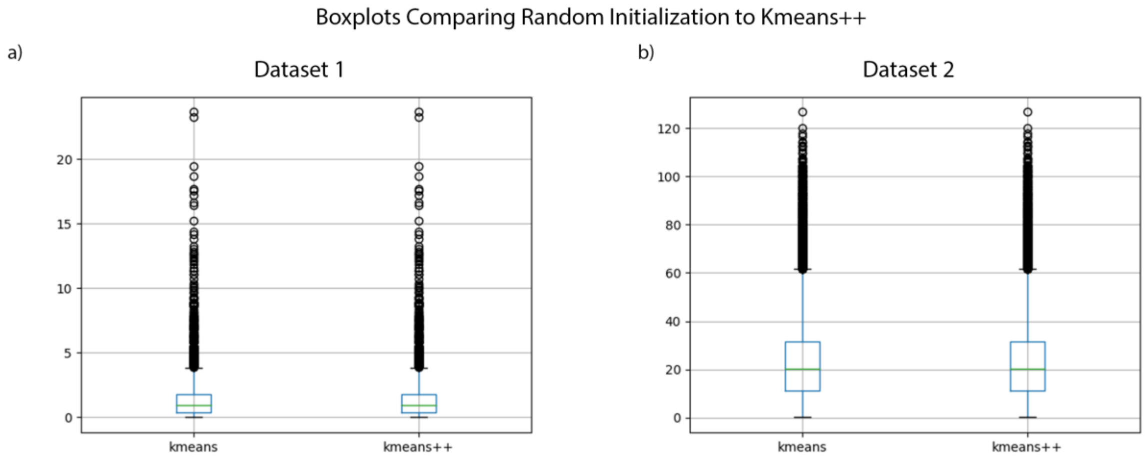

Figure 3.1.1. Boxplots Comparing K-means to K-means++

As both models are nearly identical, hypothesis testing (H_0: SE k-means = SE k-means++, alpha = 0.05) expectedly failed for both datasets (Dataset 1, p = 0.4999; Dataset 2, p = 0.500) and differences in SE were deemed not statistically significant. SSE values also reflect this lack of difference (Table 3.1.1).

Table 3.1.1. Error Comparisons

	SSE	*SSE*
Random Initialization	5117.468273281031	362713.44532064494
k-means++ Initialization	5117.468273281031	362713.44532064494

As runtime is the major defining feature between k-means and k-means++, runtime of each dataset using their optimum k values (4 and 8, respectively) was performed (Table 3.1.2). Surprisingly, the runtime was higher using k-means++ for both datasets. While this would be expectedly higher for smaller datasets given the higher initial runtime costs, we would expect it to decrease with increased dataset size due to decreased convergence time. When observing the percent differences, this trend does manifest as there is a dramatic decrease in percent difference. One could speculate k-means++ to be faster with an even larger database.

Table 3.1.2. Runtime Comparisons

	Runtime Dataset 1 (sec)	*Difference (sec, %)*	*Runtime Dataset 2 (sec)*	*Difference (sec, %)*
Random Initialization	4.004295	(9.942395, 71.3%)	288.352337	(28.516776, 9.0%)
k-means++ Initialization	13.946690		316.869113	

Note: Percent calculated based off k-means++ value (difference/k-means++*100%)

3.2 Comparing Average Linkage to Single Linkage Hierarchical Agglomerative Clustering

The usage of different linkage strategies likely requires a nuanced approach to their utilization. Inspection of the datasets utilized suggest average linkage to be optimal for the smaller dataset (Dataset 1) as the clusters best represent the plotted data. The same is true of single linkage strategy for the larger dataset (Dataset 2). Average strategies would likely be more optimal for data with well defined clusters spread across a further distance. This is due to the larger distance between datapoints used in the formulation of their clusters. When clusters are close to one another like Dataset 2, single linkage is optimal as it is less sensitive to changes in cluster proximity.

3.3. Concluding Remarks

Of the unsupervised models observed, Hierarchical Agglomerative Clustering appears to be the superior clustering model based on the experimentation performed in this study. HAC using average linkage was optimal for the smaller dataset (Dataset 1) and HAC using a single linkage strategy was optimal for the larger dataset (Dataset 2). Both matched the plotted clusters to near perfection. HAC models in general appear to be an optimal unsupervised strategy due to several factors. In addition to the lack of need to observe the cost of multiple k-values to select the optimal value, it also has the benefit of clustering based on variable cluster sizes. As Lloyd's/k-means++ algorithm looks to maximize the median value of a cluster, hierarchical clustering instead clusters solely on proximity and similarity of features. As a result, k-means aims to obtain clusters that are approximately equal in shape versus HAC obtains clusters based on proximity.

References

[1] Mehta, N. (2020). Clustering: Lectures 15 and 16 [PDF slides]. Retrieved from https://web.uvic.ca/~nmehta/data_mining_summer2020/lecture15-16_clustering.pdf.

[2] “Sklearn.cluster.AgglomerativeClustering.” Scikit, scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html. Accessed July 26, 2020.