Seng474 Assignment 3

Ali Gaineshev
V00979349

November 18, 2023

Introduction

In this report, we applied Lloyd's algorithm (k-means) and Hierarchical Agglomerative clustering to analyze two datasets. The main objective is to find the optimal number of k (clusters) in order to divide the dataset into separate sets in a way that objects in it are similar. The only hyperparameter that needs to be adjusted is the number of clusters itself, since we can divide a set with *n* elements into *n* clusters, but it would obviously not be optimal. To find an optimal number of k, we assigned a cost for each group of clusters (in k-means) with the main objective to minimize distance between points in a cluster.

Data

We have 2 datasets with distinct elements in each of them. Dataset 1 consists of 3500 elements, and each element is represented as a two-dimensional point.

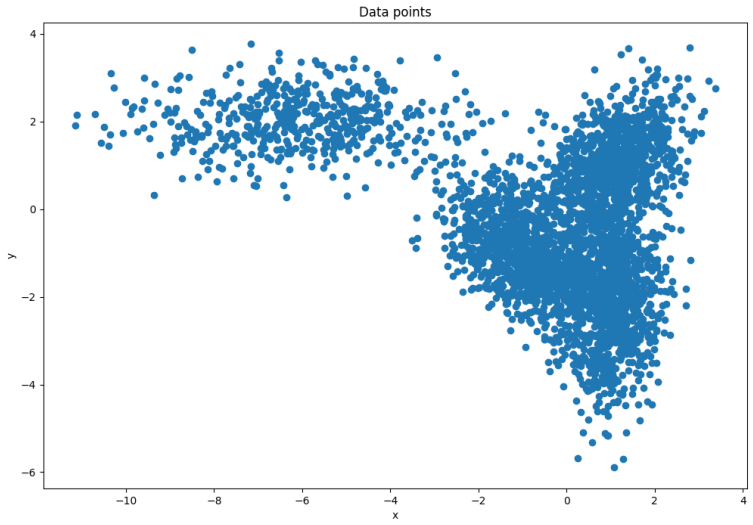

Figure 1. Dataset 1 visualization

We can see points can be separated somewhat into 2 groups on figure 1, however there are some outliers on the plot and it's not clear if that is the right choice. On the other hand, dataset 2 has three-dimensional points with 14801 elements.

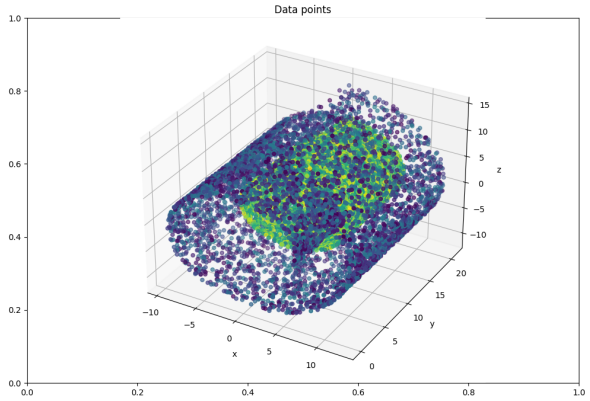

Figure 2: Dataset 2 visualization using matplotlib

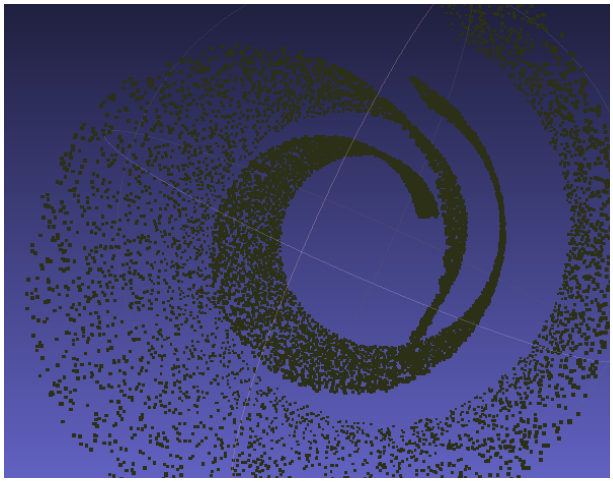

Figure 3. Dataset 2 visualization using meshlab

On figure 3, we can see dataset 2 having a distinct shape, so we expect acquiring optimal clusters to be harder.

Lloyd's algorithm

For implementing k-means clustering, random and k-means++ initializations were used. With random initialization, the initial centroids are assigned uniformly at random. However, lots of problems can arise with this [1], [2]. Another alternative is k-means++ where we assign one centroid randomly, but then place centroids across the dataset. In our runs of both algorithms, we haven't observed major problems with the random initialization, and the difference between within-cluster sum of squares (wcss) was not major. Dataset 1 was run 5 times for each k and the lowest cost was picked to be represented on the plot

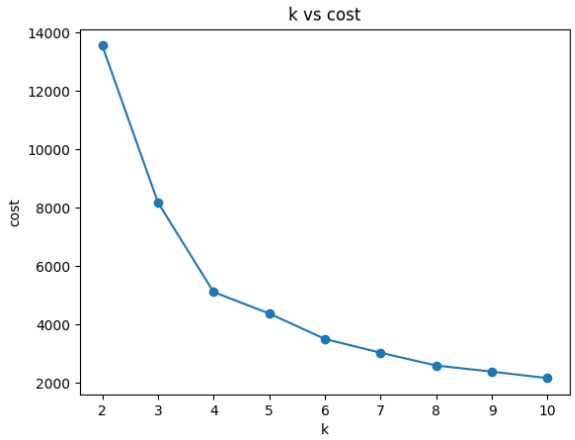

Figure 4: Dataset 1 with random initialization

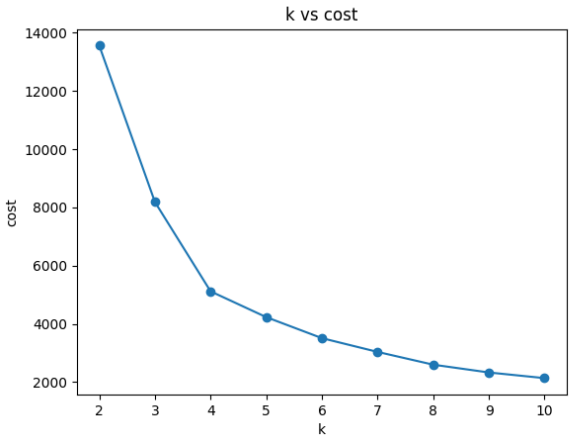

Figure 5. Dataset 1 with k-means ++

We can see that both graphs are identical for dataset 1 (see figures 4, 5), except for some anomaly with random initialization at k = 5. Using the elbow method, we can try to guess what the optimal number of clusters should be. If we look at the both plots, at k = 4 the drop after is not as fast as before and the graph doesn't decrease as rapidly anymore. Therefore, I claim that the optimal number of clusters is 4 for dataset 1.

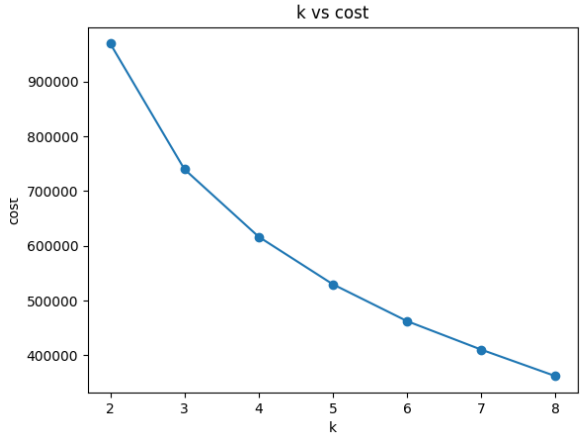

Figure 6. Dataset 2 with random initialization

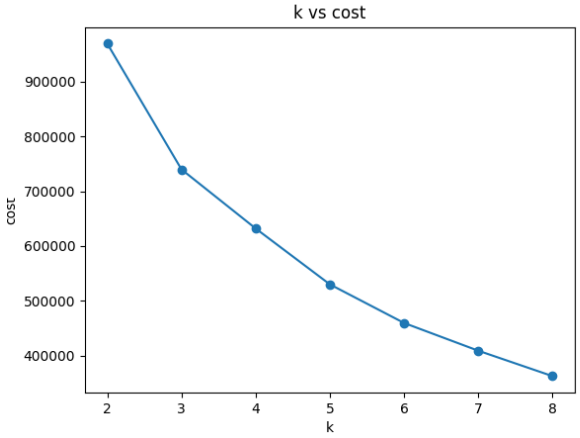

Figure 7. Dataset 2 with k-means ++

This time using the elbow method for finding the optimal number of clusters is harder, the drop is rapid at most times except after k = 4 when it gets a little slower. If we look at figure 3, we can see inner and outer rings, our human intuition tells us the number of clusters should be 2. The drop is not as rapid with k equals to 3 and 4, but it's hard to tell what the actual number of clusters should be, based on the visualization of the data. In figures 6 and 7, we can see our optimal clusters with k equals 4; however there is no difference between the initialization methods (see cost on the title).

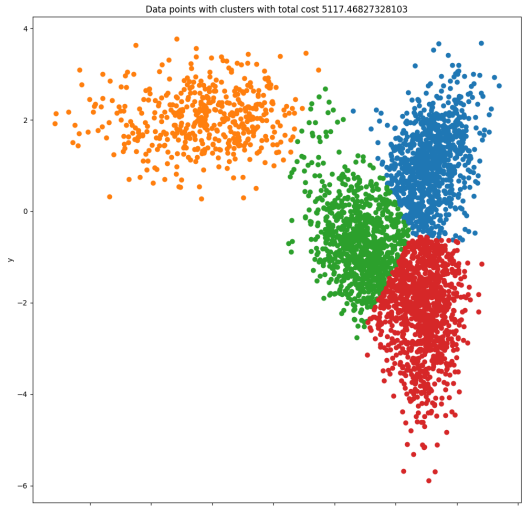

Figure 8: Dataset 1 with random initialization

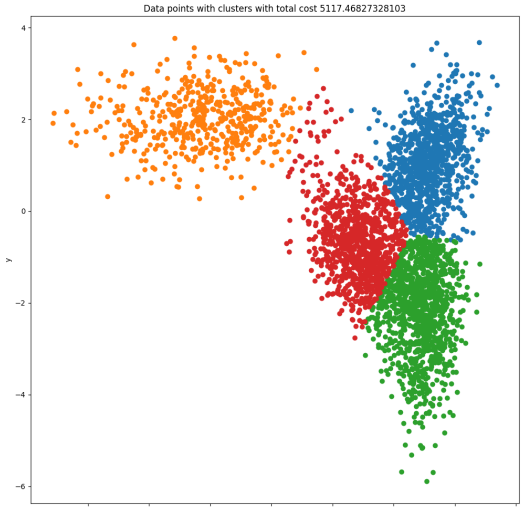

Figure 9. Dataset 1 with k-means ++

Hierarchical agglomerative clustering

For agglomerative clustering we used “single” and “average” linkages, and euclidean distance as dissimilarity measures. In figures 10, 11, 12 and 13, we can see the dendrograms of both datasets. Dendrograms for datasets with single linkage are hard to interpret and making a cut here is not the best idea. However, average linkage dendrograms are easier, and in figure 11, we can see that similarities between 2 clusters with blue lines is the longest, so the cut should be made there. Average linkage dendrogram for dataset 2 is harder to interpret, making a cut at a blue line (7 clusters) is the best one, since we capture most of similar clusters.

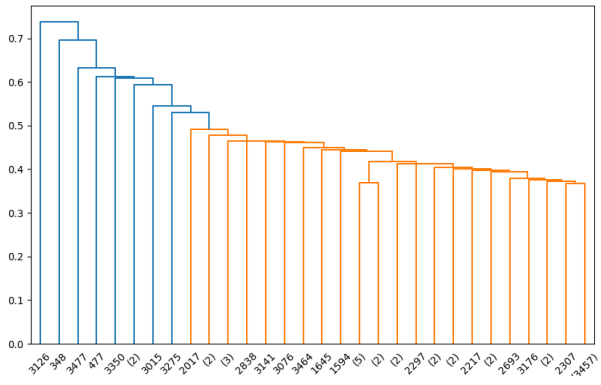

Figure 10. Dataset 1, single linkage dendrogram

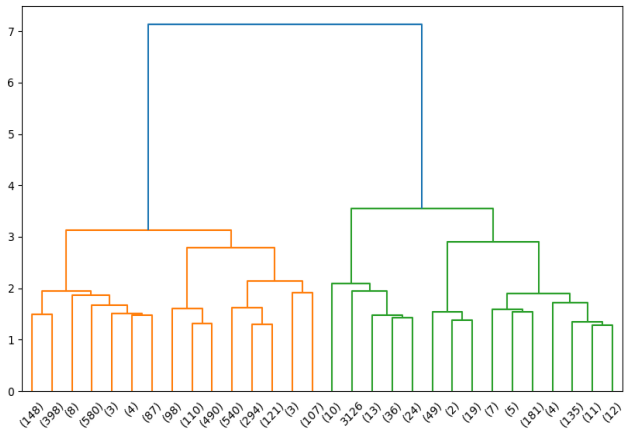

Figure 11. Dataset 1, average linkage dendrogram

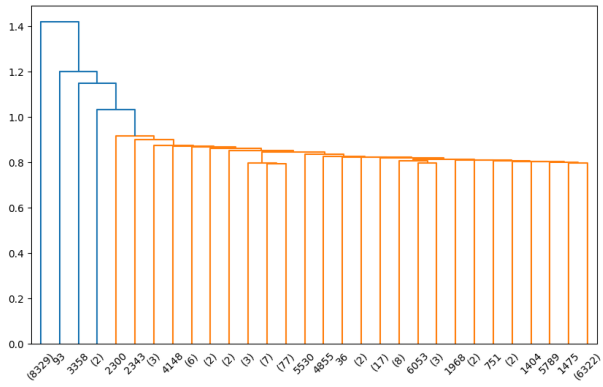

Figure 12. Dataset 2, single linkage dendrogram

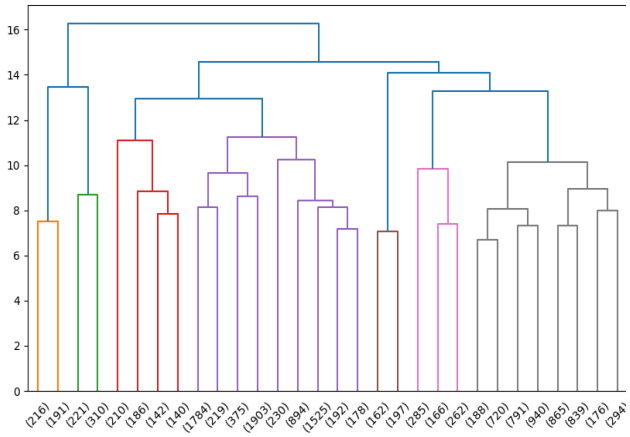

Figure 13. Dataset 2, average linkage dendrogram

Single Linkage

Using single linkage on the first dataset gave us shocking results, until we realized what was going on (see Figures 14 and 15). Single linkage criterion uses minimum distances between all observations [3], so points start to connect to each other, forming the main cluster, while the farthest point makes its own cluster and doesn't connect. It's sensitive to noise, therefore it should not be used on this dataset. However, single linkage works well on globular shapes. It actually identified the two rings in dataset 2, see figures 15, 16 and 17. The dataset 2 is well separated; and since points in the rings are close to each other, they will keep connecting, forming a chain and resulting in these 2 clusters.

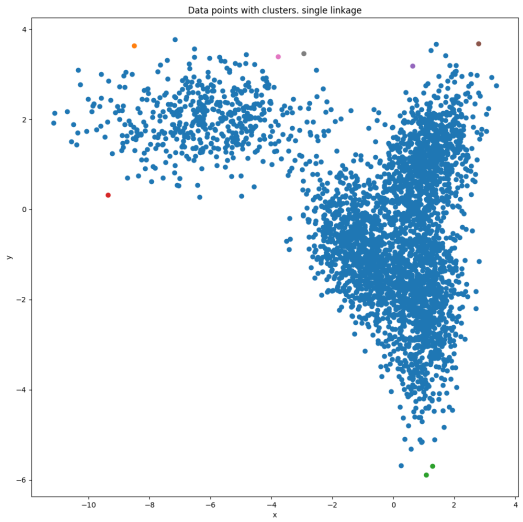

Figure 14: Dataset 1, k = 8

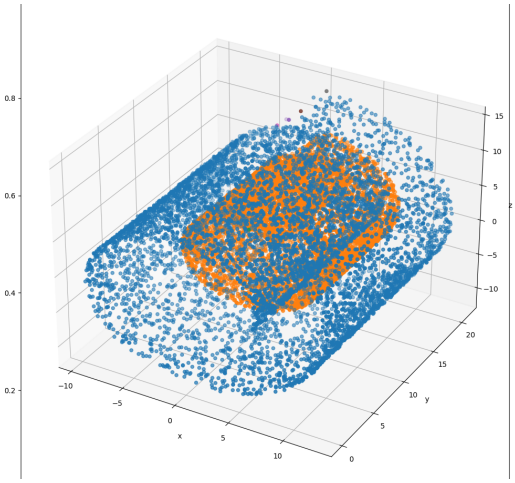

Figure 15. Dataset 2, k =8

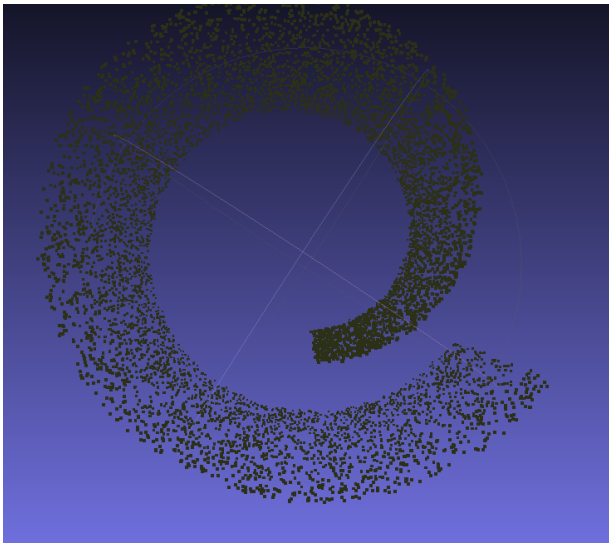

Figure 16. Dataset 2, single linkage cluster 1

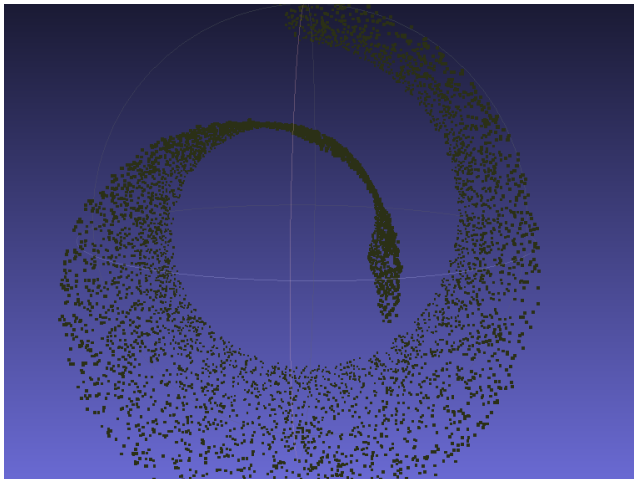

Figure 17. Dataset 2, single linkage cluster 2

Average Linkage

The last criterion is average, where we compute the average distance of all pairs in 2 sets [3]. It solves the problem of outliers and it's a compromise between single and complete linkages. If we follow the claims from the dendrograms part, the optimal clusters are shown in figures 18 and 19. If we look at figure 20, the algorithm preferred making clusters (blue and pink) with combined outer and inner rings, but it also made clusters out of parts of the rings. Figure 21 has dataset 2 with 2 clusters, it hasn't done a very good job and clusters are clearly not the best. It's hard to say if average linkage is the best criterion for dataset 2.

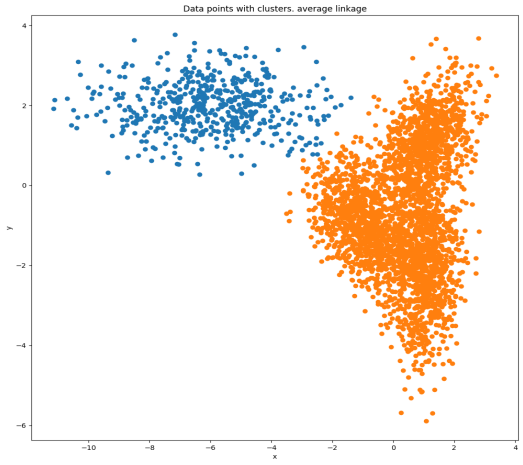

Figure 18: Dataset 1, k = 2

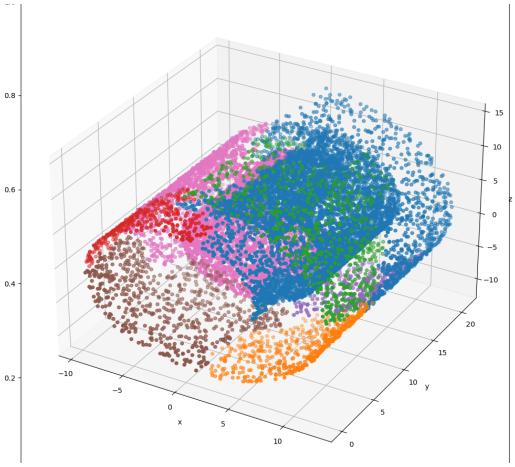

Figure 19. Dataset 2, k = 7

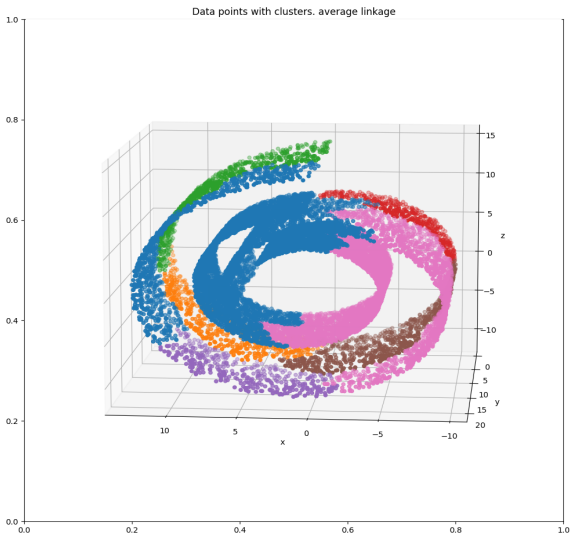

Figure 20. Dataset 2, k = 7

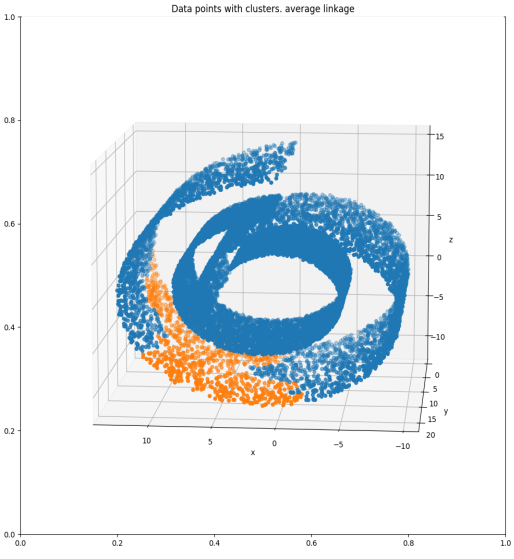

Figure 21. Dataset 2, k = 2

REFERENCES

[1] Baeldung, “The drawbacks of K-means algorithm,” Baeldung on Computer Science, https://www.baeldung.com/cs/k-means-flaws-improvements

[2] P. Sharma, “The Ultimate Guide to K-means clustering: Definition, methods and applications,” Analytics Vidhya, https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

[3] “sklearn.cluster.AgglomerativeClustering,” *scikit-learn*. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html