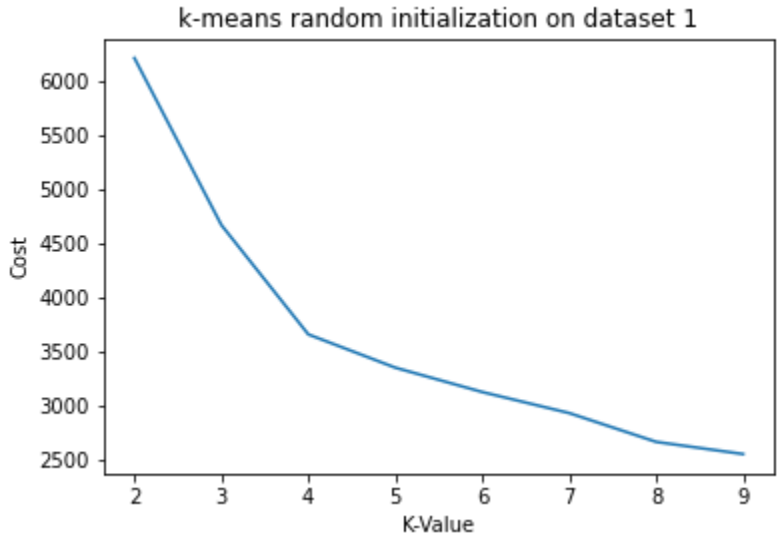
SEng 474 - Assignment 3
Kenil Shah
V00903842

Dataset 1 - 2D dataset

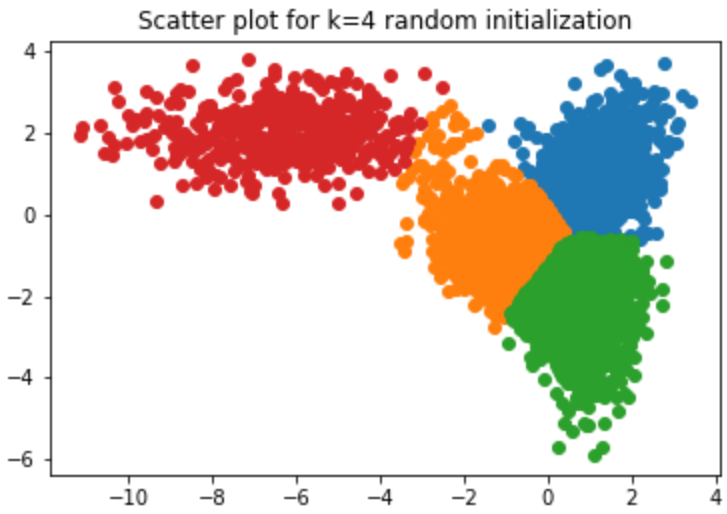
1.1 Lloyd's Algorithm

The first algorithm we implement is the Lloyd's algorithm with two forms of initialization, uniform random initialization and k-means++ initialization. We vary the number of clusters k from 2 to 10, and calculate the cost for both types of initialization to determine the optimal number of clusters to use. We decide the k value when we notice a kink in the graph, which shows that the cost has stopped decreasingly rapidly and has reached the point of diminishing returns.

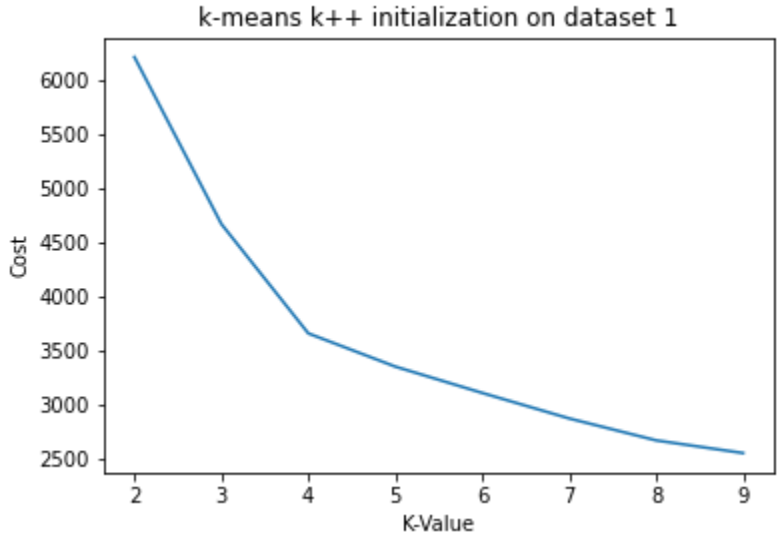
1.1.1 Uniform Random Initialization



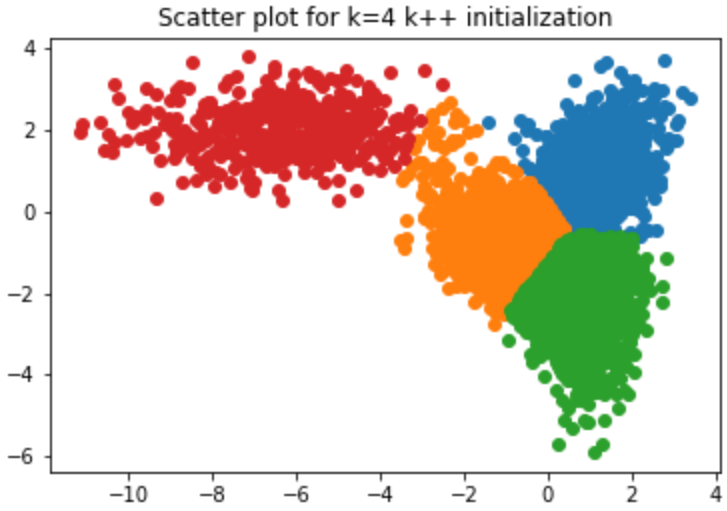
Based on the graph we notice kinks at k=4. The scatter plot for this cluster is shown below.



1.1.2 k-means++ Initialization

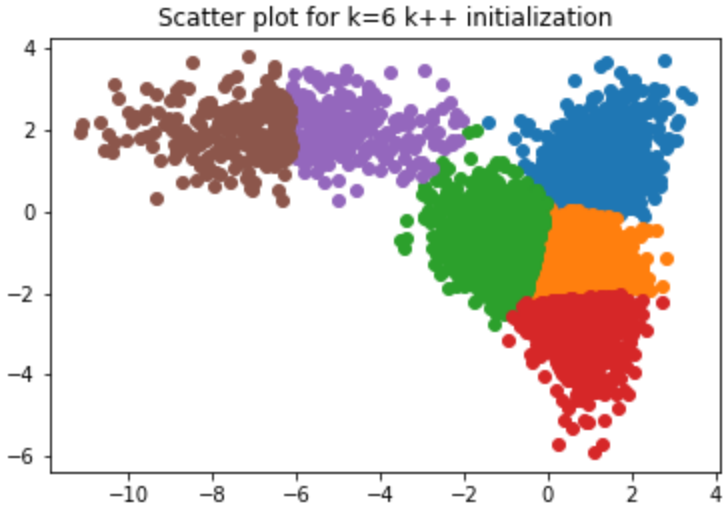


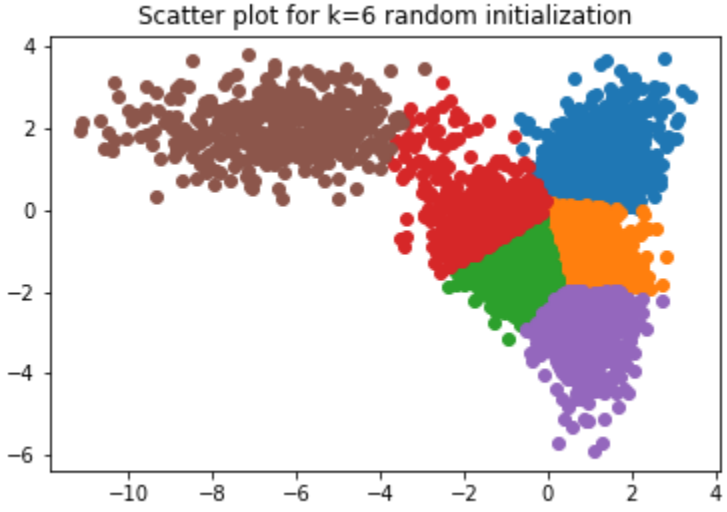
Here we see kinks in the graph at both k=3 and k=4. But there is a less rapid decrease after k =4, thus we believe k=4 to be the optimal number of clusters. The scatter plots for k=4 is shown below.

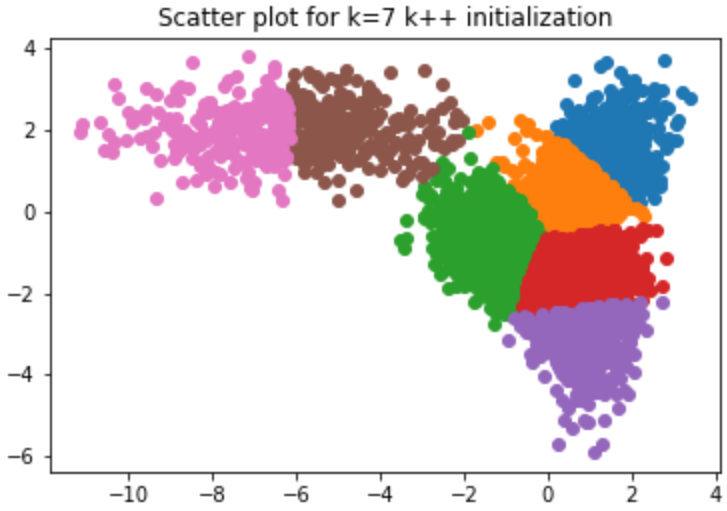


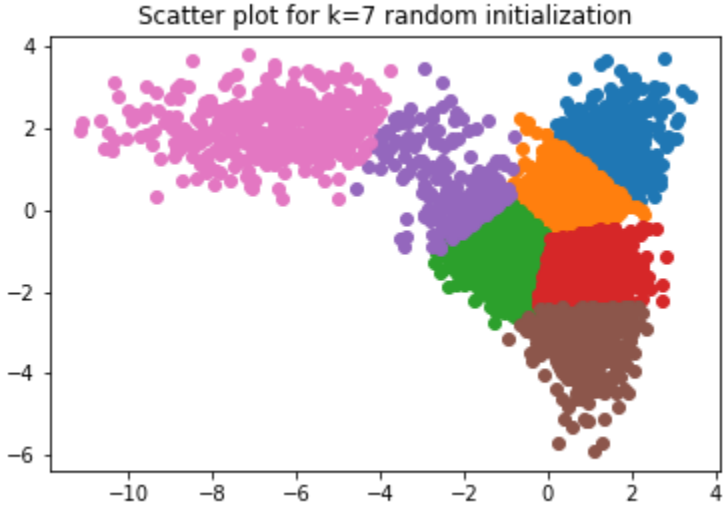
1.1.3 Observations

The above results would lead us to believe there is no difference in these two initialization methods, but when we check the scatter plots for higher numbers of clusters we see that even though the optimal number of clusters may be the same, the clusters are made completely different.


Scatter plot for k=6 k++ initialization
4
2
0
−2
−4
−6
−10
−8
−6
−4
−2
0
2
4


Scatter plot for k=6 random initialization
4
2
0
−2
−4
−6
−10
−8
−6
−4
−2
0
2
4

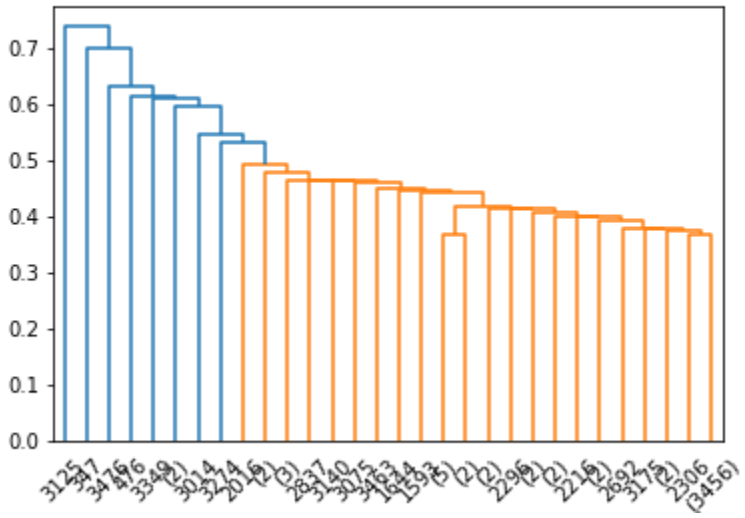




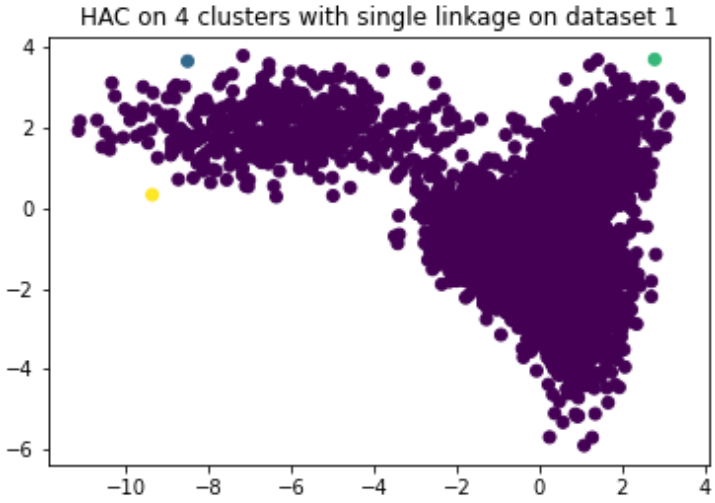
1.2 Hierarchical agglomerative clustering

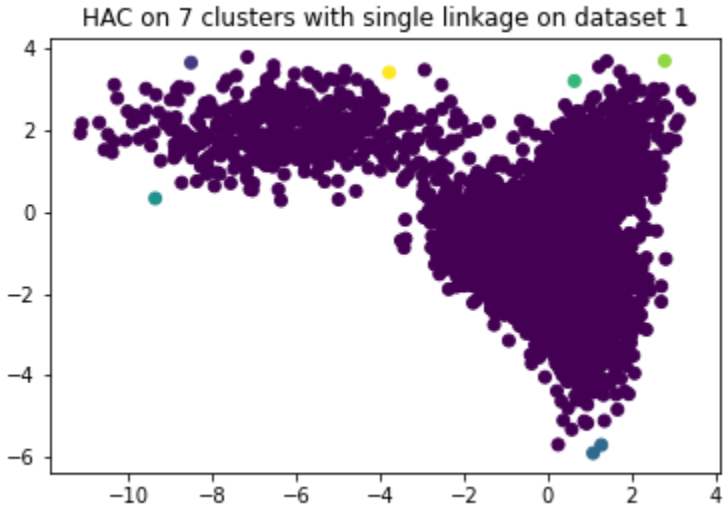
The second algorithm we implement is the HAC, with two types of linkage - single and average. We make dendrograms and select a cut to decide the optimal number clusters to use. The cut is decided based on a reasonable assumption of high dissimilarity between the cluster members.

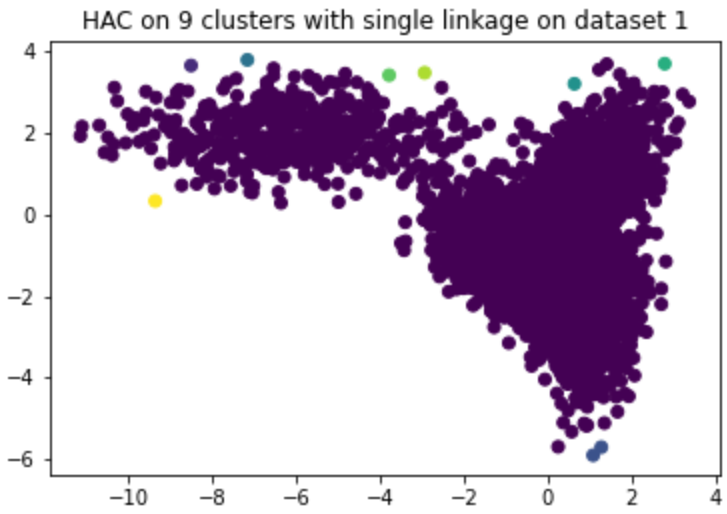
1.2.1 Single Linkage



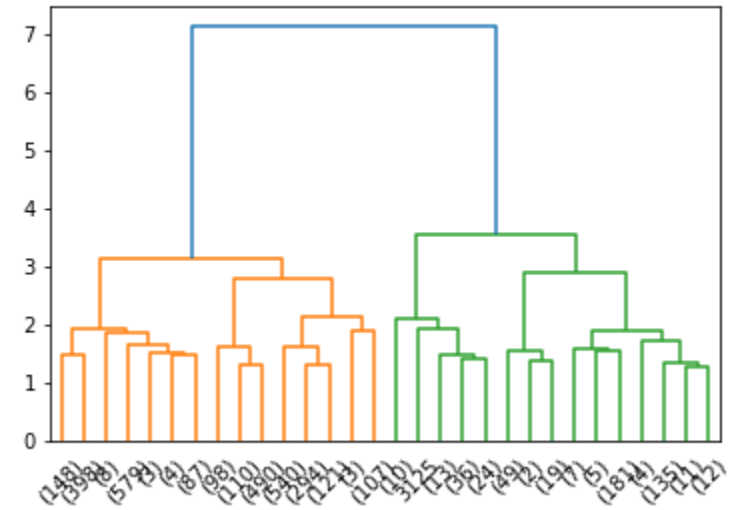
Based on the dendrogram, the reasonable cuts to check would be at k =4, 7, 9. But we found out that using single linkage did not give good clusters as presented by the scatter plots below.


HAC on 4 clusters with single linkage on dataset 1
4
2
0
−2
−4
−6
−10
−8
−6
−4
−2
0
2
4

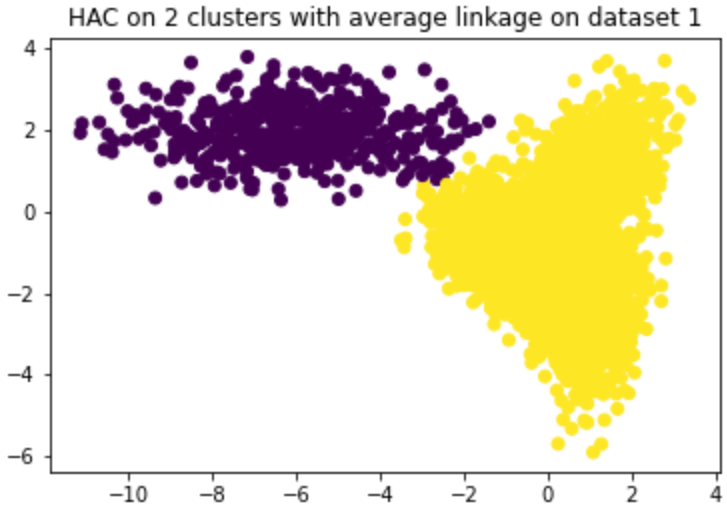

HAC on 7 clusters with single linkage on dataset 1
4
2
0
−2
−4
−6
−10
−8
−6
−4
−2
0
2
4

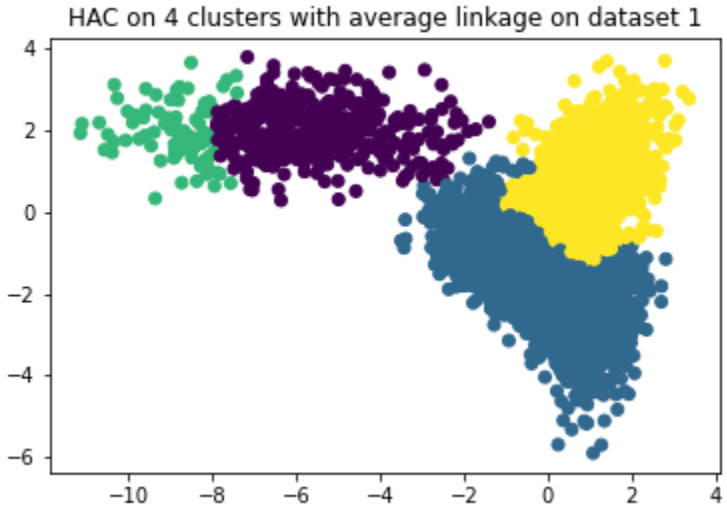


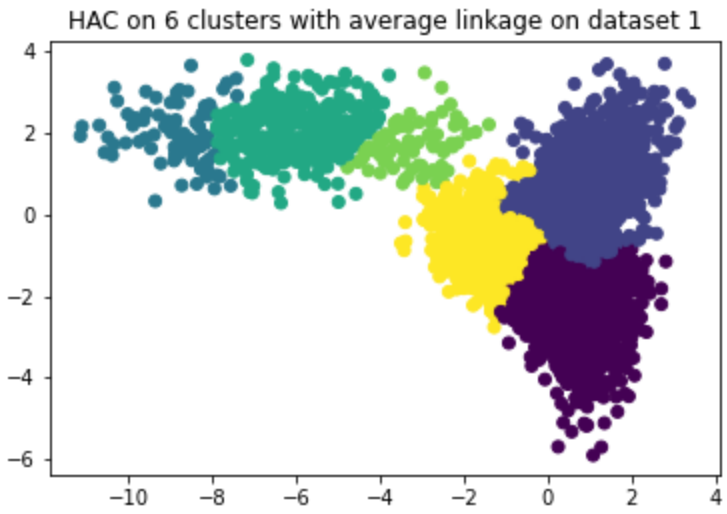
1.2.2 Average Linkage



Based on the dendrogram, the reasonable cuts to check would be at k = 2,4,6. Average linkage provides way better clusters than single linkage and seems to be the better way for clustering than single linkage.


HAC on 2 clusters with average linkage on dataset 1
4
2
0
−2
−4
−6
−10
−8
−6
−4
−2
0
2
4


HAC on 4 clusters with average linkage on dataset 1
4
2
0
−2
−4
−6
−10
−8
−6
−4
−2
0
2
4



Dataset 2 - 3D dataset

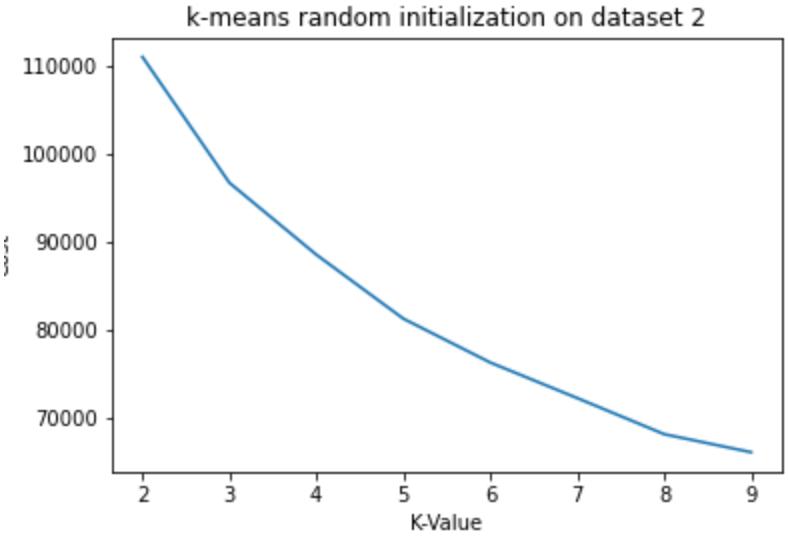
We apply the same algorithms and methods we applied for the 2D dataset to this dataset.

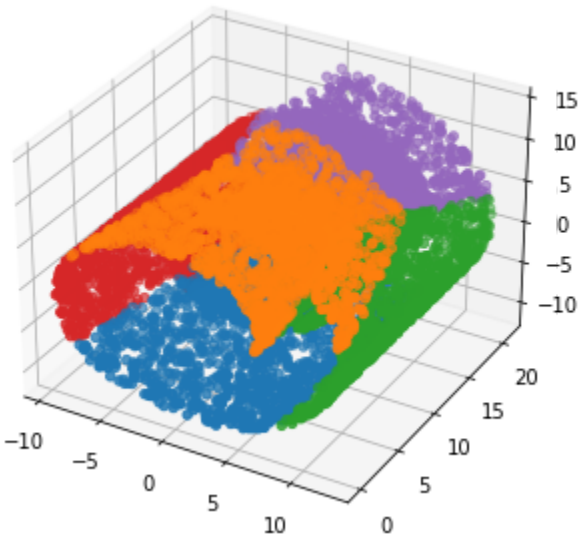
2.1 Lloyd's Algorithm

Similar to the previous method, we employ 2 initialization methods while varying k from 2 to 10, to find the optimal number of clusters based on cost.

2.1.1 Uniform Random Initialization

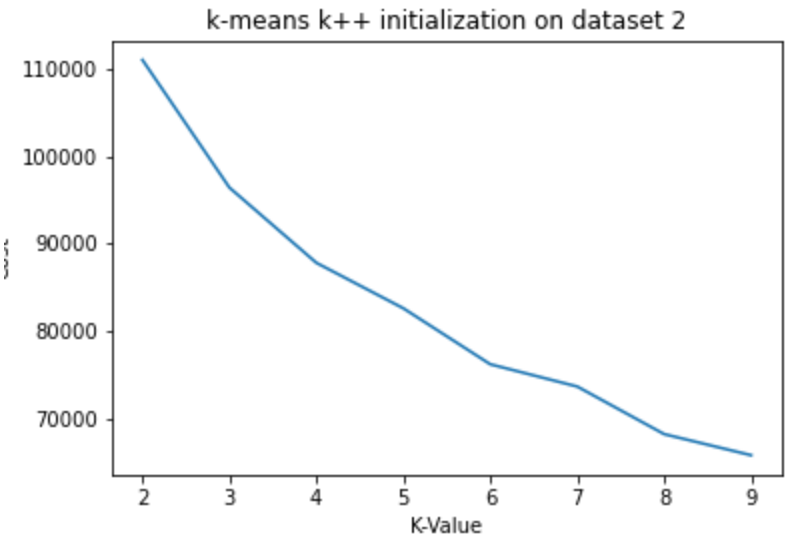
It is harder to check because of the smoother graph, but we see after k = 5 that there is a point of diminishing returns. Thus 5 is the optimal number of clusters.

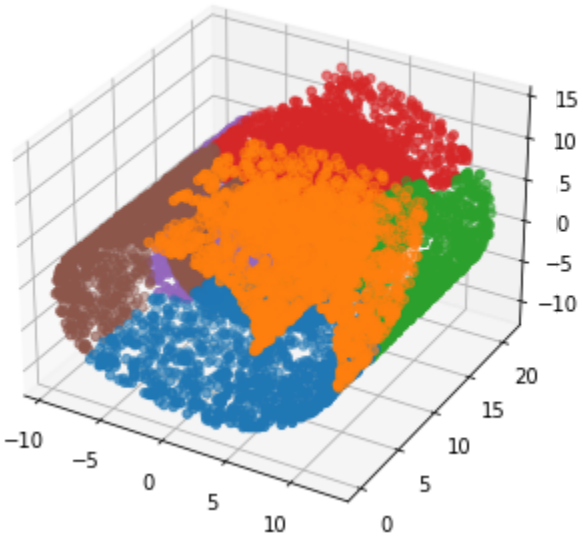




2.1.2 K-means++ initialization

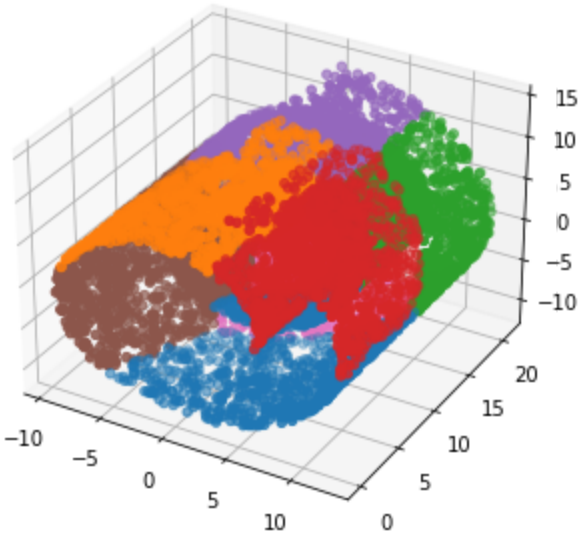
Here we see kinks in the graph at both k=4 and k=6. But there is a less rapid decrease after k =6, thus we believe 6 to be the optimal number of clusters. The scatter plots for k=6 is shown below.

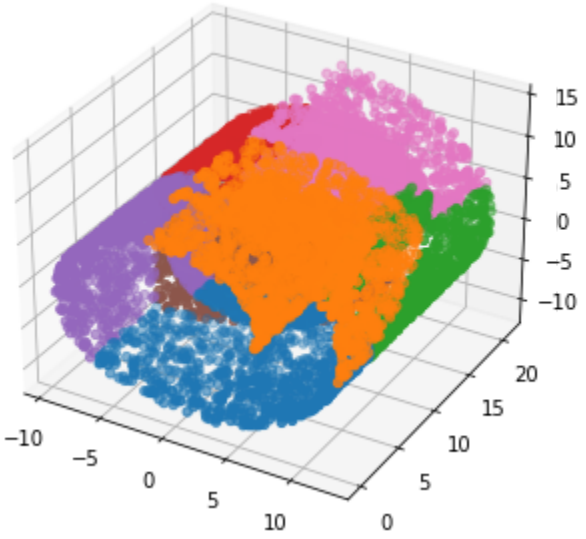


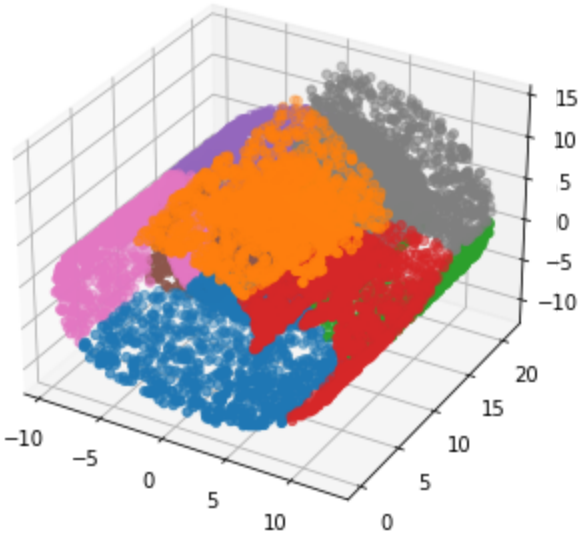


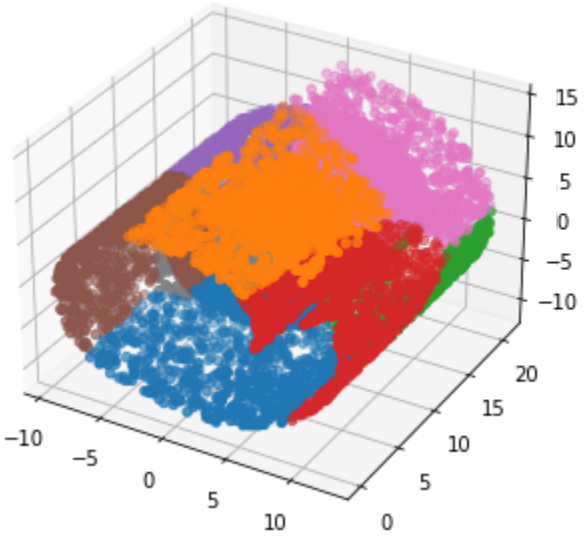
2.1.3 Observations

Similar to the first dataset, we see a difference of cluster types for both initialization methods.


15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10

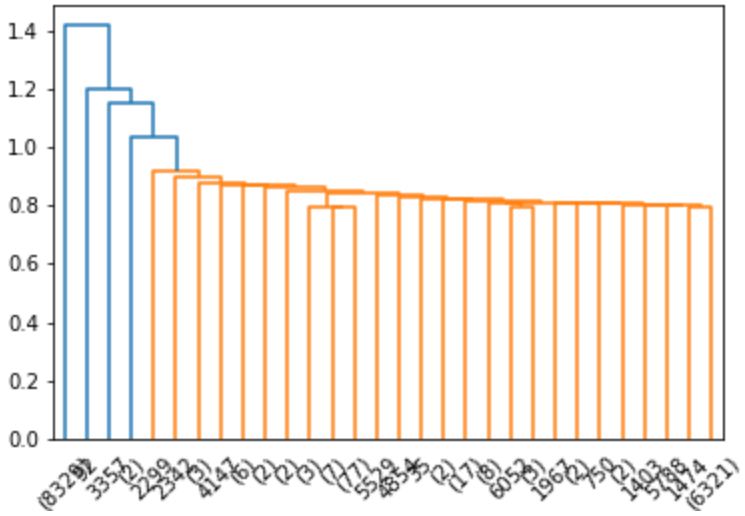

15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10



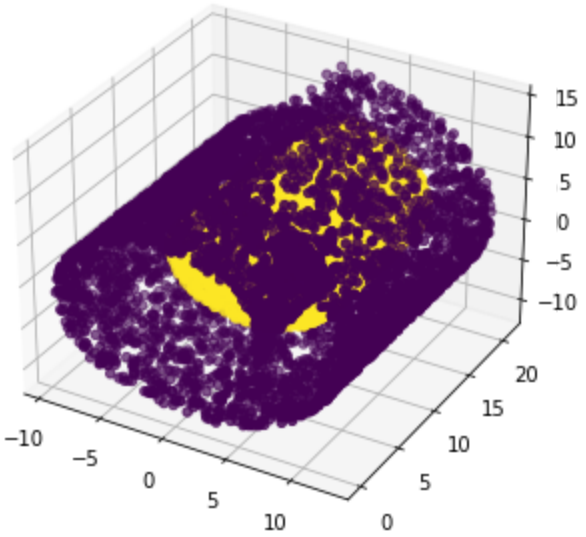


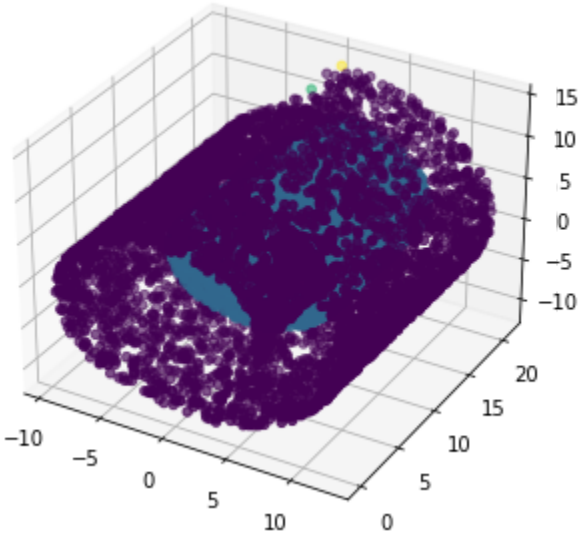
2.2 Hierarchical agglomerative clustering

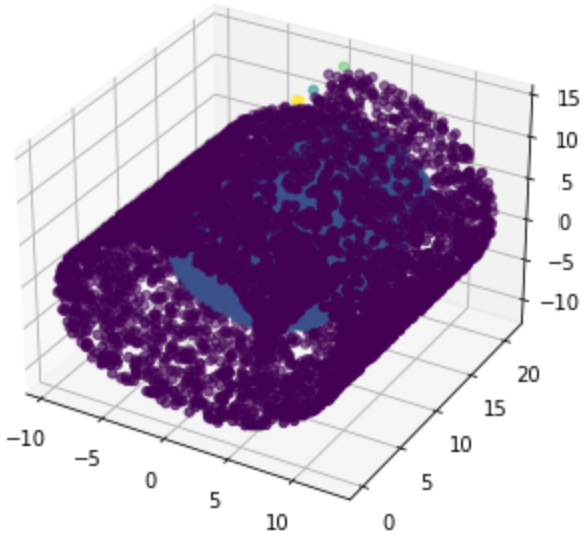
2.2.1 Single Linkage



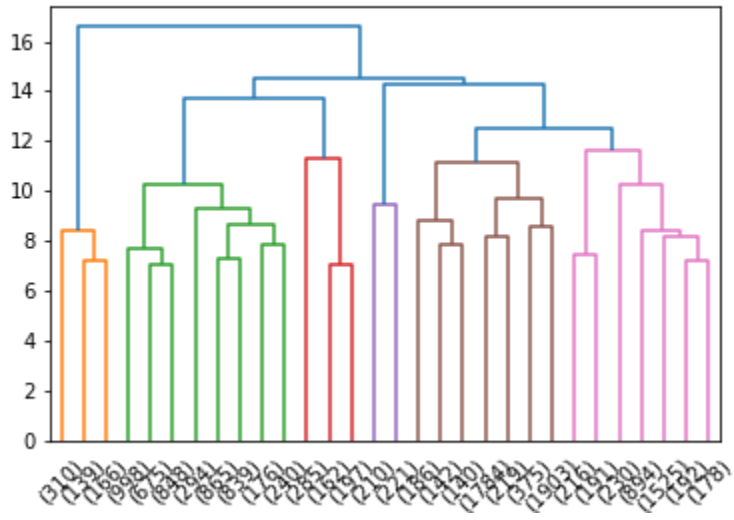
Based on the dendrogram, we make cuts at k =2,4,5. We notice that for 2 clusters, single linkage works the best, but breaks for higher numbers of clusters. This is shown in the scatter plots below.


15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10

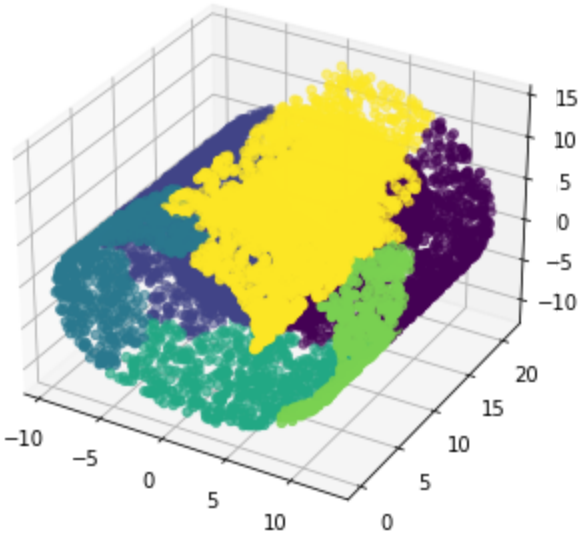

15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10

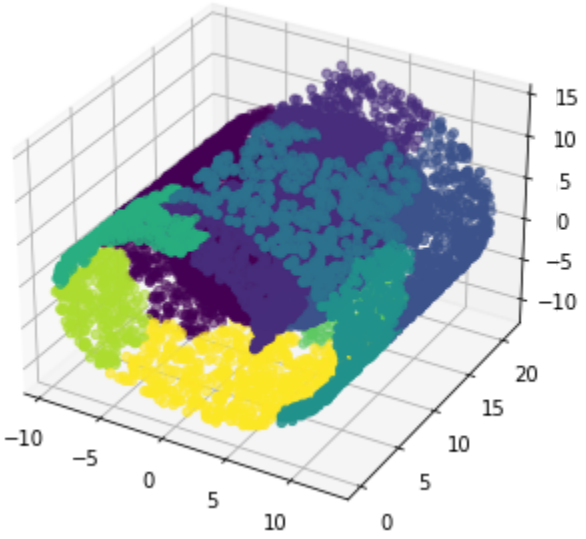


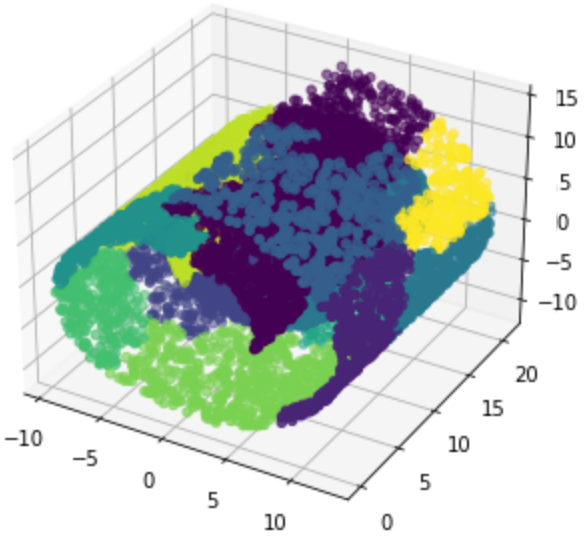
2.2.2 Average Linkage



Based on the dendrogram, we make cuts at k =6,9,11. Average linkage in general provided better visual cuts than single linkage. But compared to 2 cluster in single linkage the performance is lacking.


15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10


15
10
5
0
−5
−10
20
15
10
5
0
−10
−5
0
5
10



References

- https://datasciencelab.wordpress.com/2013/12/12/clustering-with-k-means-in-python/
- https://www.geeksforgeeks.org/ml-k-means-algorithm/
- https://medium.com/geekculture/implementing-k-means-clustering-with-k-means-initialization-in-python-7ca5a859d63a
- https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html