Datasets

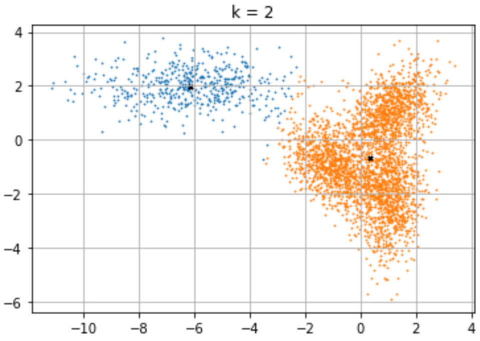
For this report two datasets were used to assess Lloyd's algorithm with uniform random sampling and K-Means++ as well as Hierarchical Agglomerative Clustering. The first dataset consists of 3500 two-dimensional examples generated by a Gaussian matrix model. The second dataset consists of 14801 three-dimensional examples.

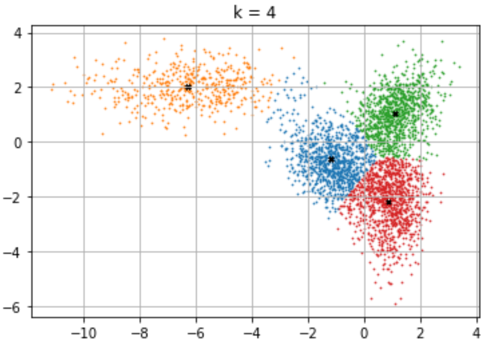
Experiments on Dataset 1

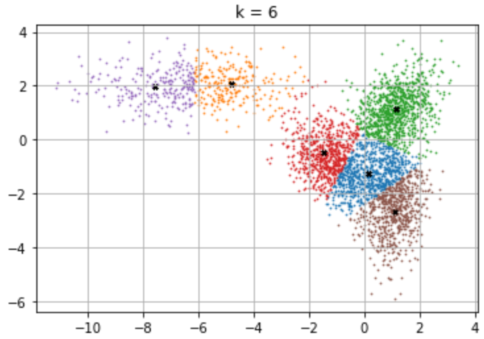
There were four experiments conducted when analyzing dataset 1: random sampling with k-means, k-means++, hierarchical clustering with single linkage and average linkage. This section will briefly describe how the experiments were conducted and highlight some of the results for further analysis.

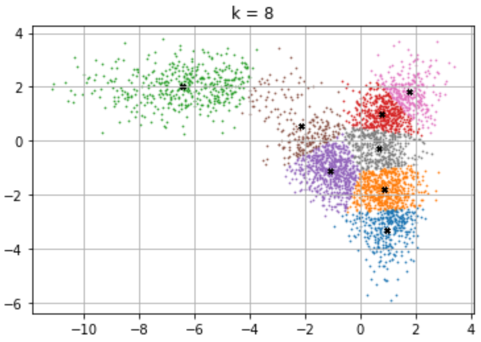
Lloyd's Algorithm with Random Sampling

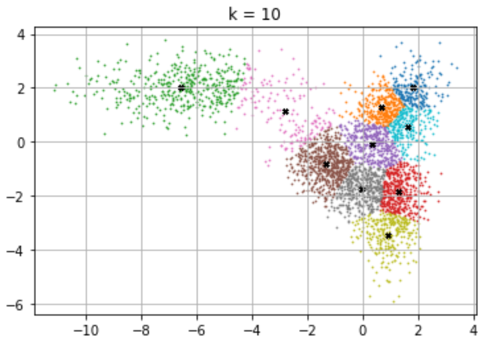
To get a better understanding of how clustering works and to be able to see its performance this method was repeated five times with the value of k (number of clusters) increasing each time in steps of 2 (the number of clusters used k=2,4,6,8,10). Furthermore, to get the best result for each value of k, the algorithm was run for a total of 10 times. Lastly, within each iteration, the cost of the algorithm was recorded and the iteration with the least cost was picked to be plotted. The figures below show how this dataset was clustered using random sampling with different values of k.


k = 2


k = 4


k = 6


k = 8


k = 10

It can be seen from the five figures above that the random sampling method clusters the dataset evenly depending on the number of clusters and generally does a good job. To support this claim and the figures above refer to table 1 and its accompanying graph (figure 1) where the cost for the cluster is shown for the different values of k.

Number of Clusters (k)	Cost
2	14259.705066088092
4	808.2330547290546
6	83.92741335320427
8	435.78965127427307
10	6.159017194332744

It is important to note that these costs were calculated at the end of each iteration and the minimum cost was picked to be the cost for that value of k. However, the cost could have been calculated at the end of the last iteration. The reason for this is because we know that with each iteration of random sampling the cluster centers change and get more accurate and can separate the data better. Hence, eventually, by the end of all the iterations we would have the lowest cost.

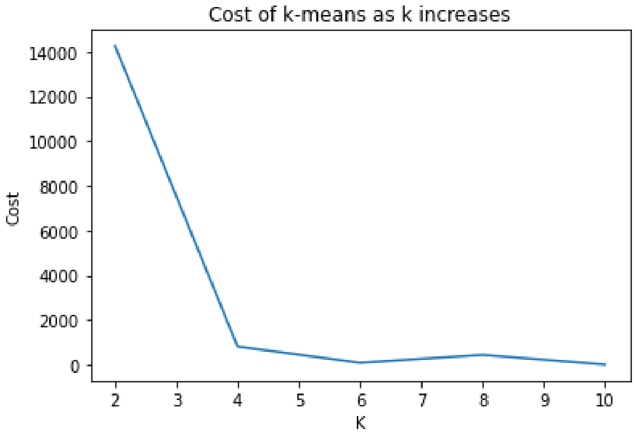


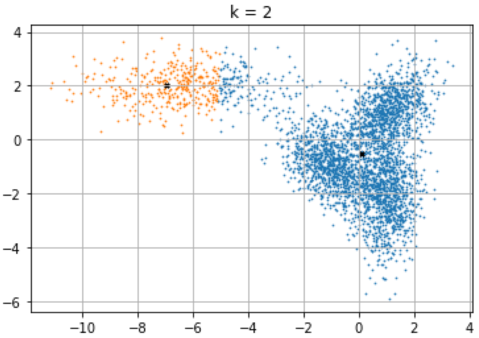
Figure 1: Cost of clustering as k increases

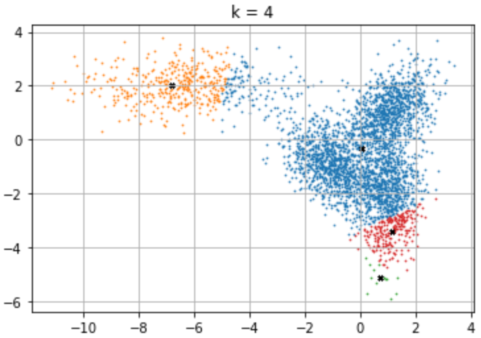
From the table above and figure 1 it is evident that as the number of clusters (k) increases the cost for the algorithm drops drastically. Although, there is an outlier in this graph. If you look at k = 6 and k = 8 the cost of the former is a lot lower than the cost of the later. This may be considered as an error, but we can choose to move past it as the overall cost begins to decline again after k = 8. Furthermore, the random sampling is in fact, random to some extent which could be another reason for such a discrepancy in the results.

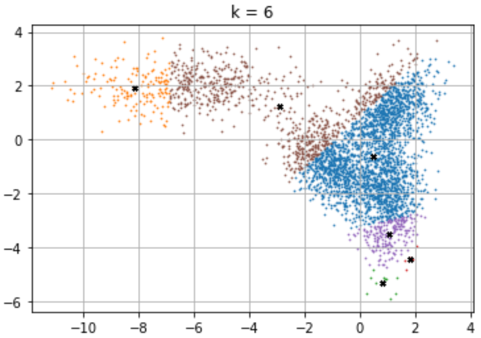
All in all, Lloyd’s algorithm using random sampling to find the cluster means is a very effective way to cluster and segregate the dataset with some labels. However, there may be some errors due to the randomness of the algorithm. The elbow method could be employed to find the optimal value of k for the given dataset; however, it was not used for this report. Despite not finding the optimal value for k, it can be concluded that the more iterations performed for each value of k the lower the overall cost will be for that given value and the higher the value of k the better it performs (lowest cost).

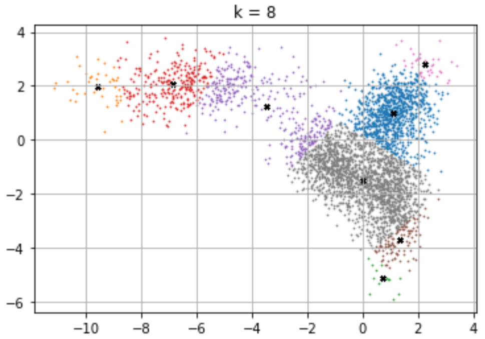
Lloyd’s Algorithm with K-Means++

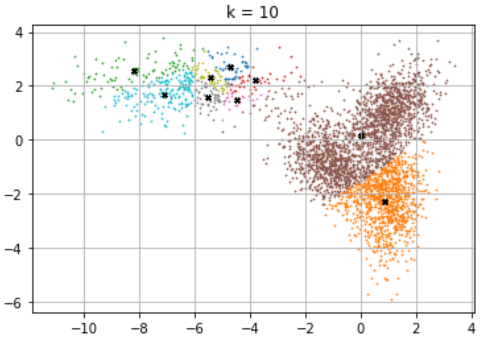
The procedure for this method was the same as when we used random sampling. The only difference is the algorithm is slightly different. This method was repeated five times with the value of k (number of clusters) increasing each time in steps of 2 (the number of clusters used k=2,4,6,8,10). Furthermore, to get the best result for each value of k, the algorithm was run for a total of 10 times. Lastly, within each iteration, the cost of the algorithm was recorded and the iteration with the least cost was picked to be plotted. The figures below show how this dataset was clustered using random sampling with different values of k.











It can be seen from the five figures above that the k-means++ method clusters the dataset evenly to an extent and generally does a good job. Just by comparing the graphs for this method from the graph of random sampling it is evident that k-means++ does not evenly separate the data while random sampling does. To support this claim and the figures above refer to table 2 and its accompanying graph (figure 2) where the cost for the cluster is shown for the different values of k.

Number of Clusters (k)	Cost
2	187139.37327095168
4	119664.08766371295
6	34787.9607617496
8	1711.9193914108844
10	6.710949303712788

Despite the unevenness in the clustering by this method the table above along with figure 2 highlight similar results to previous section. It is evident that as the number of clusters (k) increases the cost for the algorithm drops drastically. However, in this method there is no outlier, and the cost of the algorithm is in a constant decline as the number of clusters increase. The outlier could have been avoided in the previous section by running the entire process again and hoping the randomness does not produce another outlier.

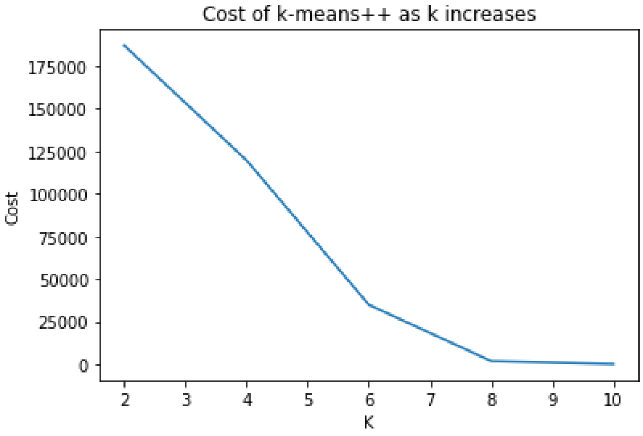


Figure 2: Cost of clustering as k increases

All in all, Lloyd's algorithm using k-means++ to find the clusters is a very effective way to segregate the dataset. This method does not employ as much randomness as random sampling, hence having a smaller probability of outliers. Again, the elbow method could have been used to find the optimal value of k for the given dataset; however, it was not used for this report. Despite not finding the optimal value for k, it can be concluded that the more iterations performed for each value of k the lower the overall cost will be for that given value and the higher the value of k the better it performs (lowest cost). Lastly, it is important to note that the cost of random sampling and k-means++ with 10 clusters are very close to each other, it can be concluded that the two methods are equally powerful but random sampling provides a more even split while k-means++ prevent outliers.

Hierarchical Agglomerative Clustering using Single Linkage

This method was run only once for an "optimal" value of k. To find the optimal value of k, the dataset was converted into a dendrogram, and a horizontal line was placed in that plot. Once the horizontal line was place, the number of intersections were counted, and that number became the "optimal" number of clusters to be used. Figure 3 shows the dendrogram for the first dataset with the horizontal line. The choice for the horizontal line was based on the longest vertical distance without any horizontal line passing through it.

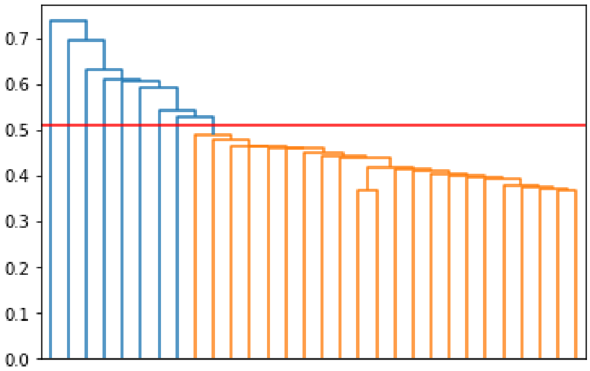


Figure 3: Single Linkage Dendrogram (Dataset1)

Based on figure 3, the optimal number of clusters to use for hierarchical agglomerative clustering for dataset 1 is 9. Now that we have the optimal number of clusters, we can feed this information to sklearns module for this method, and it would generate the cluster for us. The figure below (figure 4) shows the first dataset being clustered using hierarchical agglomerative clustering using single linkage and 9 clusters.

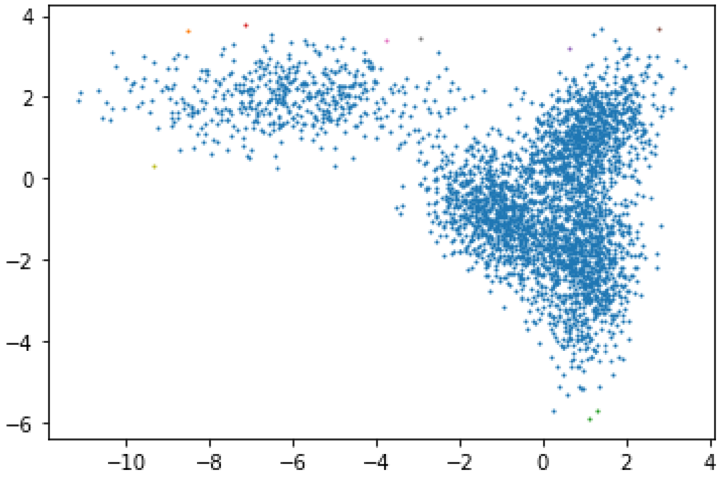


Figure 4: Clustering using Hierarchical with 9 clusters

From figure 4 it is hard to identify the clusters made by this method and only upon very close inspection can we see the points colored differently. This report does not show the figure for this method using different values for k. However, doing further experiments with different number of clusters resulted in a similar graph as figure 4. With this output it can be concluded that single linkage for hierarchical agglomerative clustering is not an effective method to use to cluster this dataset.

Hierarchical Agglomerative Clustering using Average Linkage

This method is the same as the previous section, the only difference in this is that it uses average linkage instead of single linkage. Just to reiterate, this method was run only once for an “optimal” value of k. To find the optimal value of k, the dataset was converted into a dendrogram, and a horizontal line was placed in that plot. Once the horizontal line was place, the number of intersections were counted, and that number became the “optimal” number of clusters to be used. Figure 5 shows the dendrogram for the first dataset with the horizontal line. The choice for the horizontal line was based on the longest vertical distance without any horizontal line passing through it.

Based on figure 5, the optimal number of clusters to use for hierarchical agglomerative clustering for dataset 1 is 4. Now that we have the optimal number of clusters, we can feed this information to sklearns module for this method, and it would generate the cluster for us. The figure below (figure 6) shows the first dataset being clustered using hierarchical agglomerative clustering using average linkage and 4 clusters.

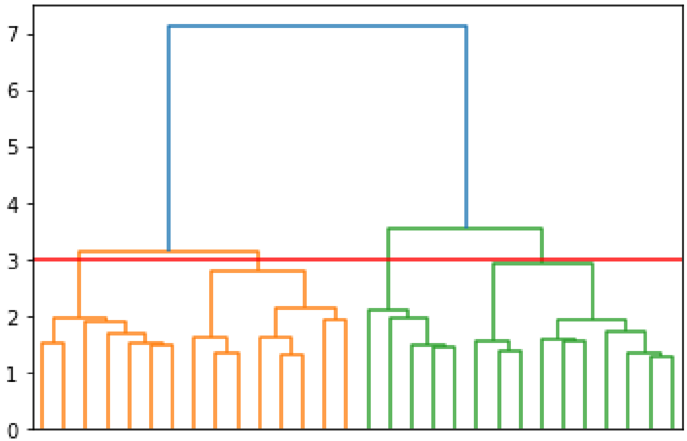


Figure 5: Average Linkage Dendrogram (Dataset1)

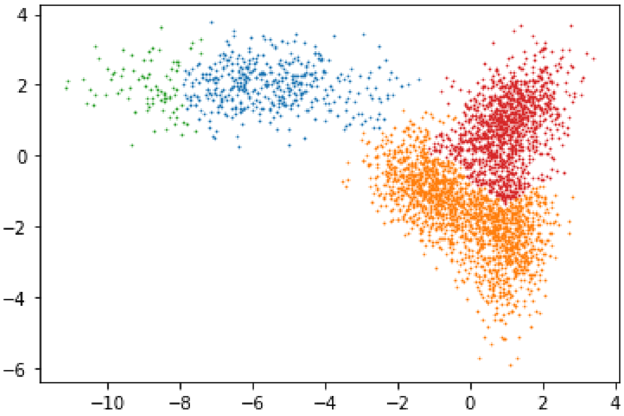


Figure 6: Clustering with Hierarchical using 4 clusters

From figure 6 the clusters made by this method are clearly visible. Upon closer inspection it is evident that the clusters are evenly distributed. From these graphs, it can be concluded that the average linkage method is far better than the single linkage method. This is because the former clearly clusters the data into the right labels while the later struggles to segregate the dataset evenly. All in all, hierarchical agglomerative clustering performs a lot better when using average linkage and it can be said that when using average linkage its performance is as good as Lloyd's algorithm.

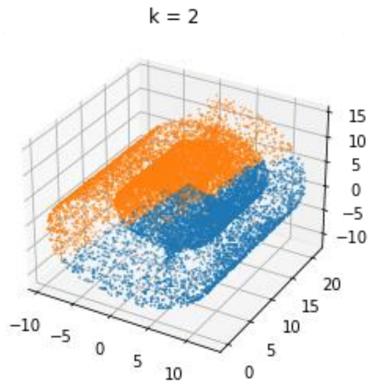
Experiments on Dataset 2

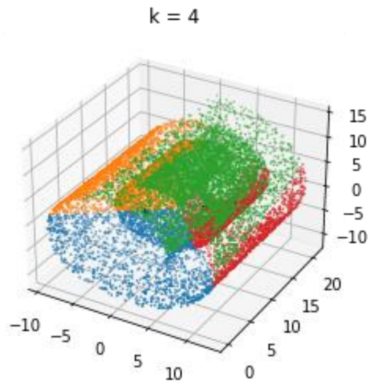
There were four experiments conducted when analyzing dataset 1: random sampling with k-means, k-means++, hierarchical clustering with single linkage and average linkage. This section will briefly describe how the experiments were conducted and highlight some of the results for further analysis.

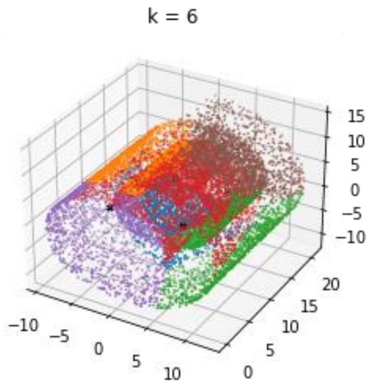
The experiments conducted for this dataset are identical to the ones used for dataset 1 and so I won't be explaining my approach for each experiment. However, the dataset in this section is three-dimensional and hence, the plots are also in three-dimensional to get a better understanding of the dataset as well as how the clustering is done.

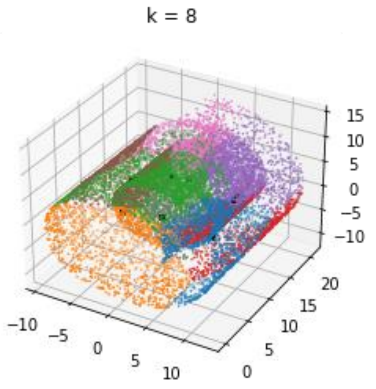
Lloyd's Algorithm with Random Sampling

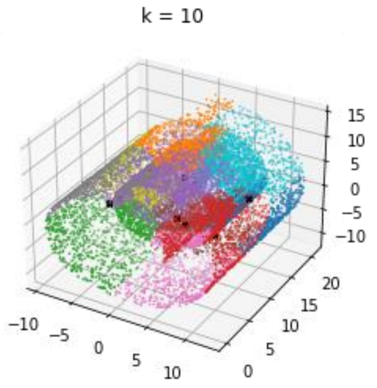
The k-means algorithm using random sampling is the same as mentioned previously. The figures below show how the dataset is clustered and segregated as the number of clusters increase. It is a little difficult to see some of the clusters as this dataset is three-dimensional and the graphing library was not the most ideal to show all the sides clearly. The table below the figures shows how the cost of the random sampling changes as the number of clusters increase, refer to figure 7 for the graph accompanying the table.











Number of Clusters (k)	Cost
2	103884958.70376444
4	66725138.31531093
6	14165443.028644532
8	3257529.6832113764
10	1225767.2893271938

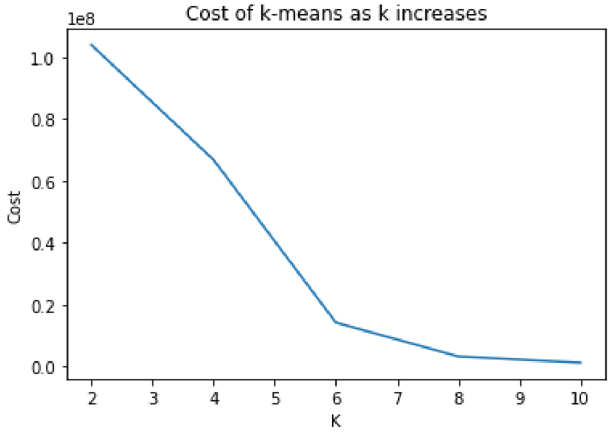


Figure 7: Cost of clustering as k increases

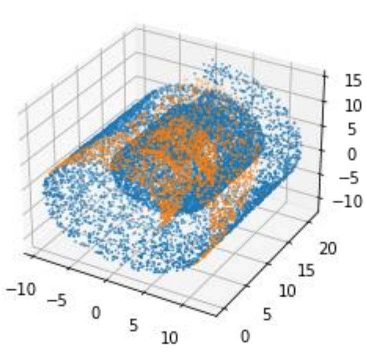
From the table above and figure 7 it is evident that as the number of clusters (k) increases the cost for the algorithm drops drastically. The method was previously seen to do the clustering in an even manner, and it continued to show this trend with this new dataset. The only difference in this dataset was the missing outlier. As mentioned above the outlier could have been avoided by running the entire process again. Luckily enough, the randomness did not produce an outlier for this run of the experiment. Lastly, comparing the of the costs between dataset 1 and dataset 2 the magnitude of the cost of dataset 2 is a lot larger, this could be because it is a three-dimensional dataset as opposed to a two-dimensional dataset.

All in all, Lloyd’s algorithm using random sampling to find the cluster means is a very effective way to cluster and segregate the dataset with some labels. There could have been some outliers due to the randomness but can be avoided (based on luck). Despite not finding the optimal value for k, it can be concluded that the more iterations performed for each value of k the lower the overall cost will be for that given value and the higher the value of k the better it performs (lowest cost).

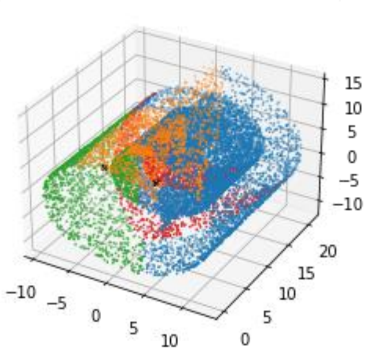
Lloyd’s Algorithm with K-Means++

The k-means algorithm using k-means++ is the same as mentioned previously. The figures below show how the dataset is clustered and segregated as the number of clusters increase. It is a little difficult to see some of the clusters as this dataset is three-dimensional and the graphing library was not the most ideal to show all the sides clearly. The table below the figures shows how the cost of the random sampling changes as the number of clusters increase, refer to figure 8 for the graph accompanying the table.

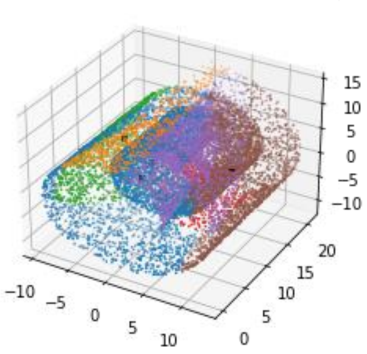
k = 2



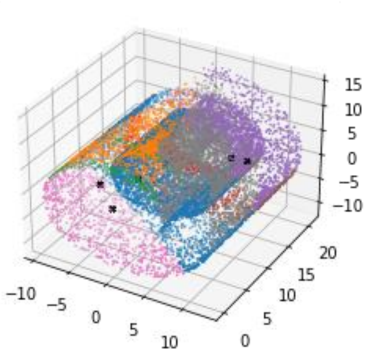
k = 4



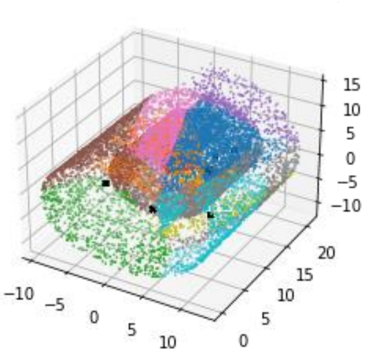
k = 6



k = 8



k = 10



Number of Clusters (k)	Cost
2	128002845.45311695
4	62189055.54468577
6	36241834.33359688
8	32000955.74532771
10	2936198.2983603566

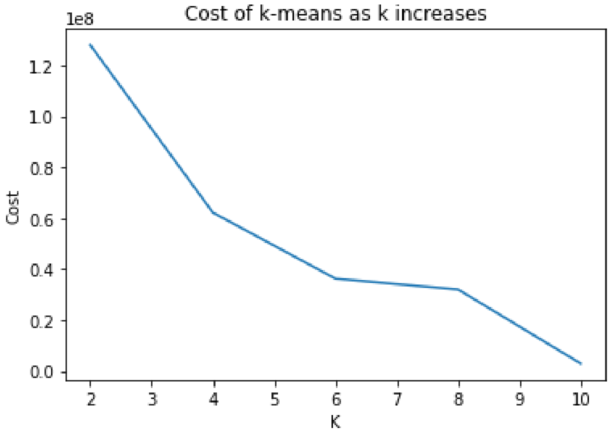


Figure 8: Cost of clustering as k increases

It can be seen from the five figures above that the k-means++ method clusters the dataset evenly to an extent and generally does a good job. Just by comparing the graphs for this method from the graph of random sampling it is evident that k-means++ does not evenly separate the data while random sampling does. This trend was also seen with dataset 1 and is strengthened by seeing it in another dataset.

It is evident that as the number of clusters (k) increases the cost for the algorithm drops drastically. However, in this method there is no outlier, and the cost of the algorithm is in a constant decline as the number of clusters increase. All in all, Lloyd's algorithm using k-means++ to find the clusters is a very effective way to segregate the dataset. This method does not employ as much randomness as random sampling, hence having a smaller probability of outliers. Despite not finding the optimal value for k, it can be concluded that the more iterations performed for each value of k the lower the overall cost will be for that given value and the higher the value of k the better it performs (lowest cost).

Hierarchical Agglomerative Clustering using Single Linkage

The process for this method was the same as when employed with dataset 1: find the "optimal" number of clusters using dendrogram and then feeding this value as well as the single linkage parameter to sklearns built in function to produce the clustered graphs. Figure 9 shows the dendrogram for single linkage of dataset 2 along with the horizontal line that was picked based on the longest vertical distance without any horizontal line passing through it (with this method the value of k is 7). The figure below (figure 10) shows the second dataset being clustered using hierarchical agglomerative clustering using single linkage and 7 clusters.

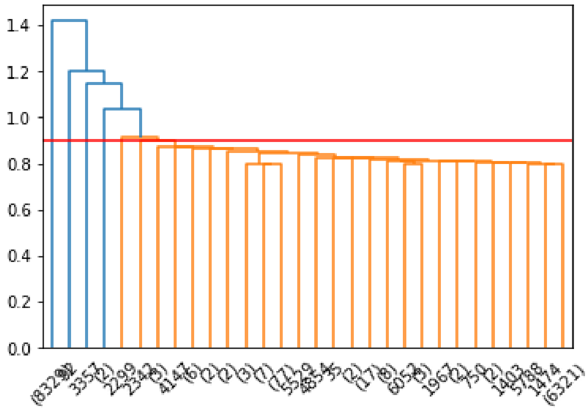


Figure 9: Single Linkage Dendrogram (Dataset2)

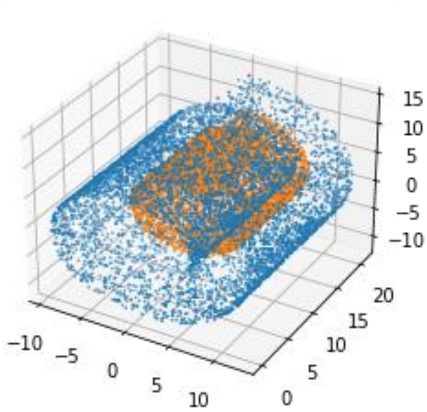


Figure 10: Clustering with Hierarchical using 7 clusters

From figure 10 it is hard to identify the clusters made by this method and only upon very close inspection can we see the points colored differently (closer to the top middle of the plot). This report does not show the figure for this method using different values for k. However, doing further experiments with different number of clusters resulted in a similar graph as figure 10. A similar outcome was seen when using this method for dataset 1 and using it again just strengthens the results from the previous section using single linkage. With this output it can be concluded that single linkage for hierarchical agglomerative clustering is not an effective method to use to cluster this dataset.

Hierarchical Agglomerative Clustering using Average Linkage

The process for this method was the same as when employed with dataset 1: find the "optimal" number of clusters using dendrogram and then feeding this value as well as the average linkage parameter to sklearns built in function to produce the clustered graphs. Figure 11 shows the dendrogram for single linkage of dataset 2 along with the horizontal line that was picked based on the longest vertical distance without any horizontal line passing through it (with this method the value of k is 6). The figure below (figure 12) shows the second dataset being clustered using hierarchical agglomerative clustering using single linkage and 6 clusters.

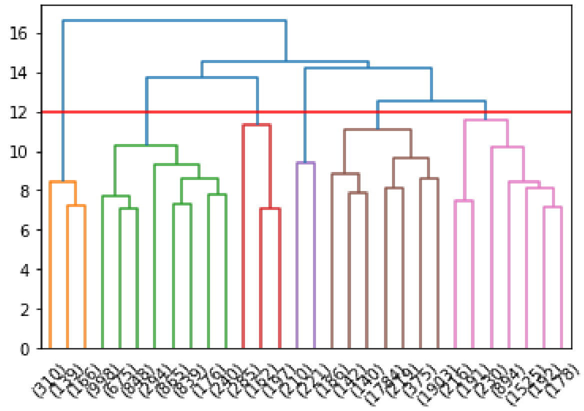


Figure 11: Average Linkage Dendrogram (Dataset2)

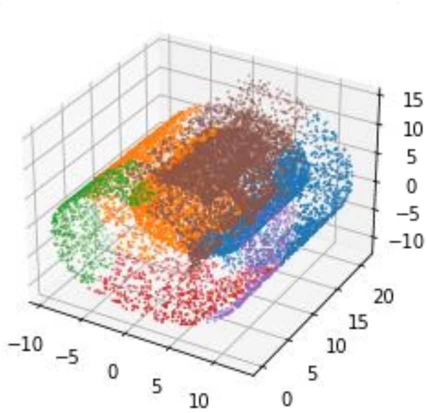


Figure 12: Clustering with Hierarchical using 6 clusters

From figure 12 the clusters made by this method are clearly visible. Upon closer inspection it is evident that the clusters are evenly distributed. From these graphs, it can be concluded that the average linkage method is far better than the single linkage method. This is because the former clearly clusters the data into the right labels while the later struggles to segregate the dataset evenly. This observation was also seen in dataset 1 and so it can be concluded that hierarchical agglomerative clustering performs a lot better when using average linkage and it can be said that when using average linkage its performance is as good as Lloyd’s algorithm (using random sampling or k-means++).