CSC 503

Assignment-3

Data Mining Analysis Report

Submitted to - Dr Nishant Mehta

Submitted by – Divyansh Bhardwaj

V# - (V00949736)

Introduction

The assignment is based on the implementing and exploring one of the unsupervised learning algorithms popularly known as clustering algorithms namely, K-means clustering and Hierarchical clustering algorithm. So, to explain the K-Means algorithm in brief, we are given some data which does not have any labels (which is the whole idea of unsupervised algorithm). The data can be some ‘d’ dimensional data. In this algorithm, nothing is done at the time of training, rather everything is done at the time of testing. During the testing phase, we initialize the first center randomly and compute all the points that share maximum similarity with the above computed centers. We keep on dividing all the points in those k clusters till the convergence is achieved. By convergence I mean that the previous means and the computed or updated means converge or become nearly equal. So, we stop in that scenario and make our final clusters and classify the points in the clusters.

Coming to Hierarchical Clustering, rather I should call it Hierarchical agglomerative clustering as for this assignment I will be referring to this version. The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It is a bottom up approach and the algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all the data points. The result is a tree-based representation of the all the data points, called dendrogram. In this assignment, I have made dendrograms for both the datasets. The basic purpose of dendrograms is to identify or extract out the optimal number of clusters which is further decided by putting a cut in the dendrogram.

For this assignment, we are given two datasets to work on. The first dataset is a two-dimensional dataset consisting of 3500 data points. The second dataset is much larger dataset which is three dimensional and consists of 14801 data points. In this assignment, I have applied the algorithms to both the datasets one by one.

For K-means clustering algorithm, I have implemented the whole algorithm all by myself. The data points are clustered into different ‘k’ clusters and the value of ‘k’ varies from 2 to 15. The cost for the clustering algorithm has been computed and graphs have been plotted for the same. This whole process has been done for both the datasets.

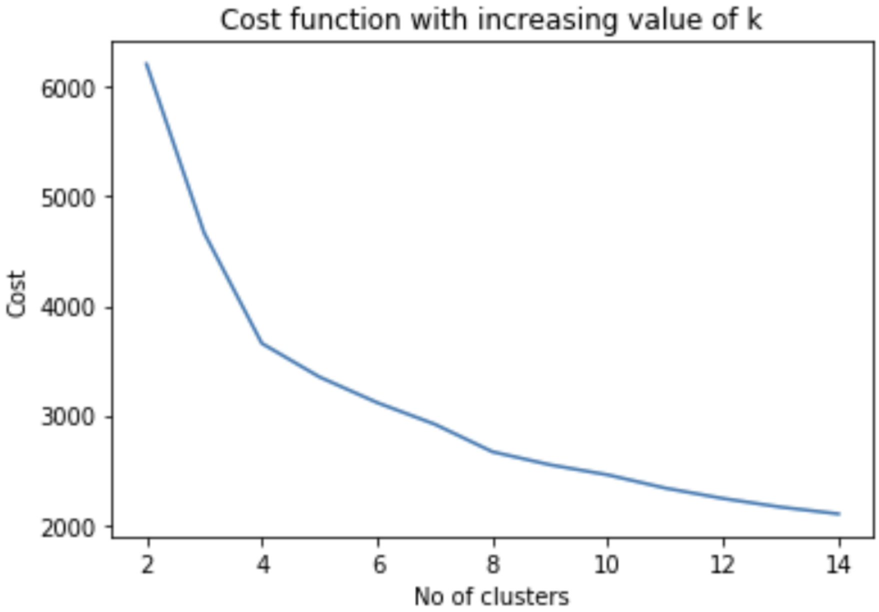
For Hierarchical Agglomerative Clustering, I have executed one of the existing implementation of the same provided by sklearn library in python. First, the dendrogram for the data has been made to identify where the cut should be placed and to decide how many clusters to go with. After having decided the optimal number of clusters, the clusters have been made for the data. The linkage plays an important role here. As per the assignment, I have toggled between the ‘single’ linkage and ‘average’ linkage. Single linkage means that when describing the measure for dis-similarity between the clusters, which here is the Euclidean distance, how we are going to define the distance between cluster. Whether it will be distance between closest points in the clusters (‘single linkage’), or will it be distance between the farthest points between the clusters (‘complete linkage’) or it might be the average distance taken (‘average linkage’). Let us get ahead with the assignment to see the results for the clusterings when run on both the datasets.

1.) K-Means Clustering (Lloyd's Algorithm):

A slight brief introduction of the K-Means clustering has been given in the Introduction section of the report. There are many algorithms available which can be used to implement the K-means clustering on the dataset. Here, as per the guidelines of the assignment, I have implemented the Lloyd's algorithm for K-means clustering. The trick is just in selecting the initial centers for the clustering. The initial centers or the centroids, according to Lloyd's algorithm, are computed by taking any random data points from the dataset and making them as the initial centroids. After that the new centers or the new means are being computed which are then further iterated till the previously computed centroids and the centroids obtained in the current iteration converge or are approximately equal. We stop there the data is said to have been clustered into 'k' clusters.

- **Implementing on 1st Dataset:**

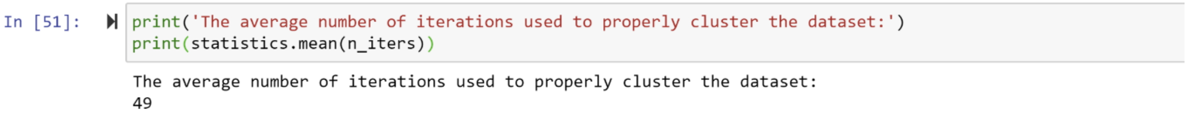
  First the Lloyd's algorithm has been implemented for the 1st dataset. As discussed above, the first dataset is a two-dimensional dataset which consists of 3500 data points. For this dataset, Lloyd's algorithm was implemented. For this assignment, I took the 'k' starting from 2 and increased it till 15 and computed the cost for each clusters. The graph for the following is shown below:



As we can see from the above figure, as the number of clusters go on increasing, the cost for the clusters go on decreasing. The graph can be further sub divided into three phases according to the rate of change of the slope.

From the value of k ranging from 2 to 4, the cost is decreasing at a very high rate. When k reaches the interval 4 to 8, the rate of change decreases going further from the interval 8 to 14, the curve tends to flatten down. The major cost decrease was between 2 to 6. After that the curve seems to be flattening down, hence 6 can be considered as a good value for making the clusters for this dataset.

I also computed and stored the iterations that were needed for each cluster to converge. Since I took the clusters from 2 to 14, I averaged the total number of iterations for all the clusters. Below is shown the average number of iterations for the first dataset:

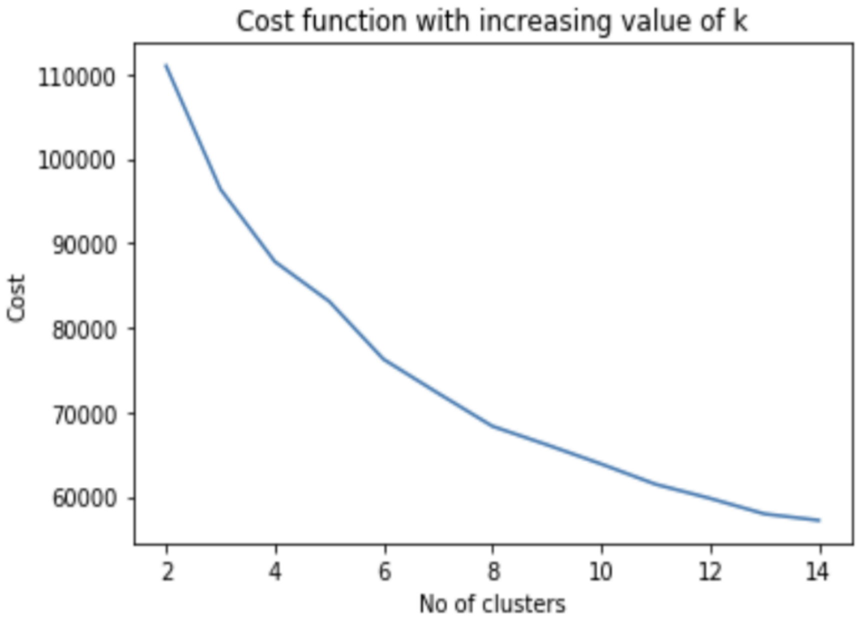


```
In [51]:   print('The average number of iterations used to properly cluster the dataset:')
           print(statistics.mean(n_iters))

           The average number of iterations used to properly cluster the dataset:
           49
```

- **Implementing on the 2nd Dataset:**

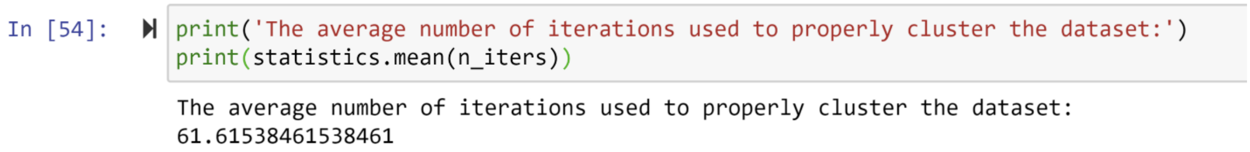
Now, the Lloyd's algorithm was implemented on the 2nd dataset. The second dataset is a three dimensional dataset consisting of 14,801 data points. For this dataset, Lloyd's algorithm was implemented. For this assignment, I took the 'k' starting from 2 and increased it till 15 and computed the cost for each clusters. The graph for the following is shown below:



From the above figure, we can clearly see that, as noticed in the previous graph as well, as the number of clusters increases, the cost of the clustering decreases. For getting the optimal clusters out of it, we should consider the value of 'k' for which the cost is not decreasing at a higher rate or has stopped decreasing.

Looking at the above graph, we can see that after the value of k has reached 8, the value of cost is not decreasing much or tehe rate of change of the cost with respect to k is quite small. Hence, it can be considered the optimal value of k for the 2nd dataset.

I also computed and stored the iterations that were needed for each cluster to converge. Since I took the clusters from 2 to 14, I averaged the total number of iterations for all the clusters. Below is shown the average number of iterations for the second dataset:



```
In [54]:  print('The average number of iterations used to properly cluster the dataset:')
          print(statistics.mean(n_iters))
```

```
The average number of iterations used to properly cluster the dataset:
61.61538461538461
```

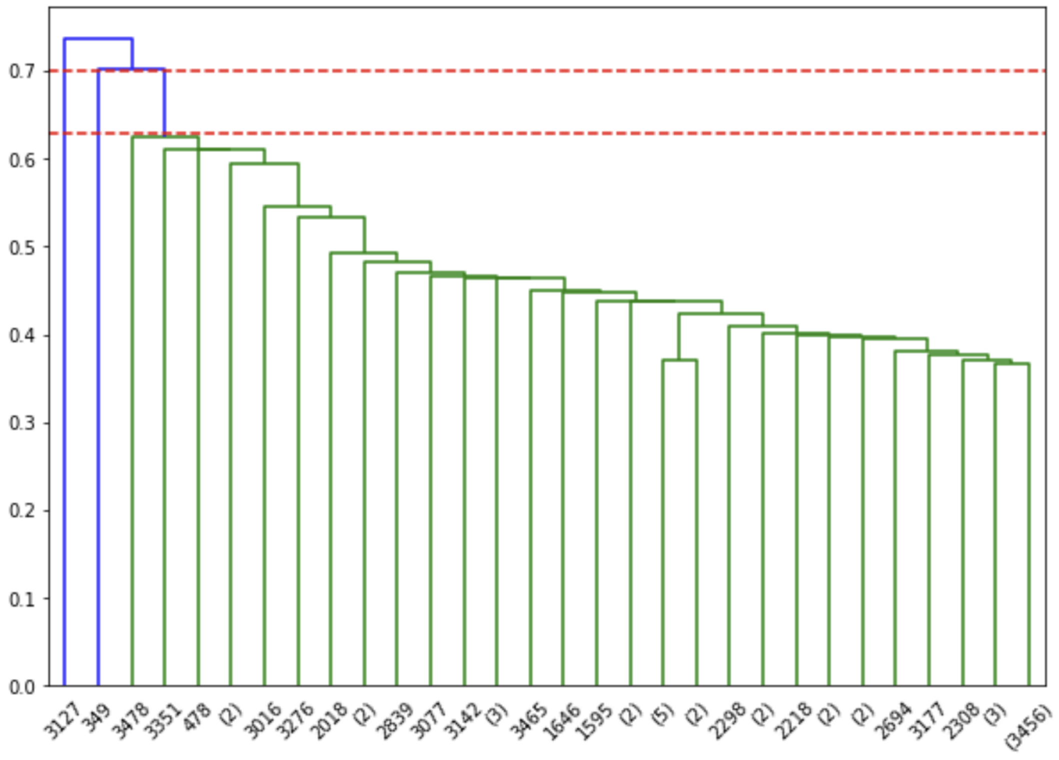
2.) Hierarchical Agglomerative Clustering

Coming to Hierarchical Clustering, rather I should call it Hierarchical agglomerative clustering as for this assignment I will be referring to this version. The agglomerative clustering is the most common type of hierarchical clustering used to group objects in clusters based on their similarity. It is a bottom up approach and the algorithm starts by treating each object as a singleton cluster. Next, pairs of clusters are successively merged until all clusters have been merged into one big cluster containing all the data points. The result is a tree-based representation of the all the data points, called dendrogram. In this assignment, I have made dendrograms for both the datasets. The basic purpose of dendrograms is to identify or extract out the optimal number of clusters which is further decided by putting a cut in the dendrogram.

As per the assignment, first I will be explaining the implementation of Hierarchical Agglomerative clustering with both single and average linkage for both the datasets. Let us first take each dataset one by one.

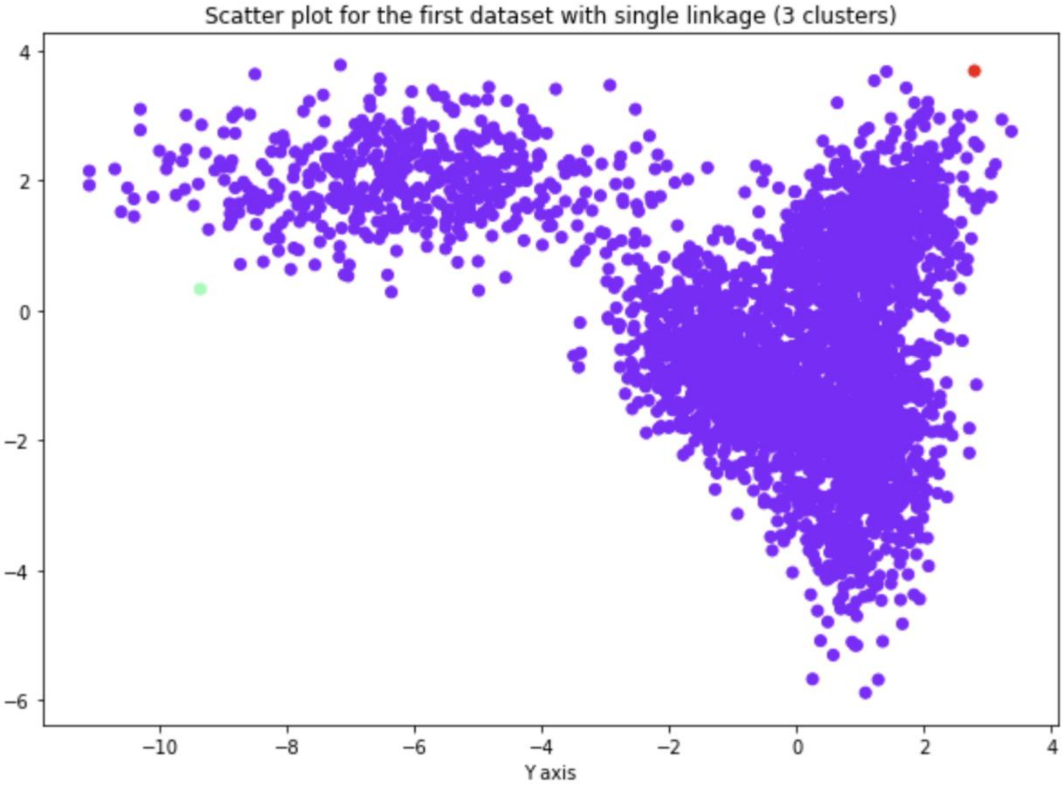
- **Hierarchical Agglomerative Clustering on 1st Dataset ('single' Linkage)**

  First the Hierarchical clustering has been applied on the first dataset with 'single' linkage which was briefly explained in the introduction section of the report. First I made a dendrogram for the same to see where the cut should have been made in order to get the optimal clusters that we should use for this dataset.
  Below is shown the dendrogram for the first dataset.

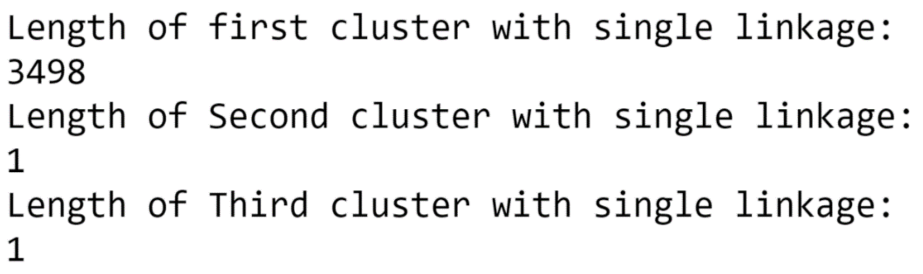


In the above figure, I have shown two red dotted lines. The number of vertical lines between these two red dotted lines gives the optimal cut or the optimal number of clusters. Basically for the optimal cut, we take into consideration the largest vertical distance between the clusters that do not intersect with each other. This criterion was taken from one of the articles that has been referenced in the readme file. Hence for this dataset, we can go with the 3 clusters as the optimal value for k.

After going with 3 clusters, we performed the algorithm on the dataset and clustered all the data points. Below shown is the scatter plot for the same:



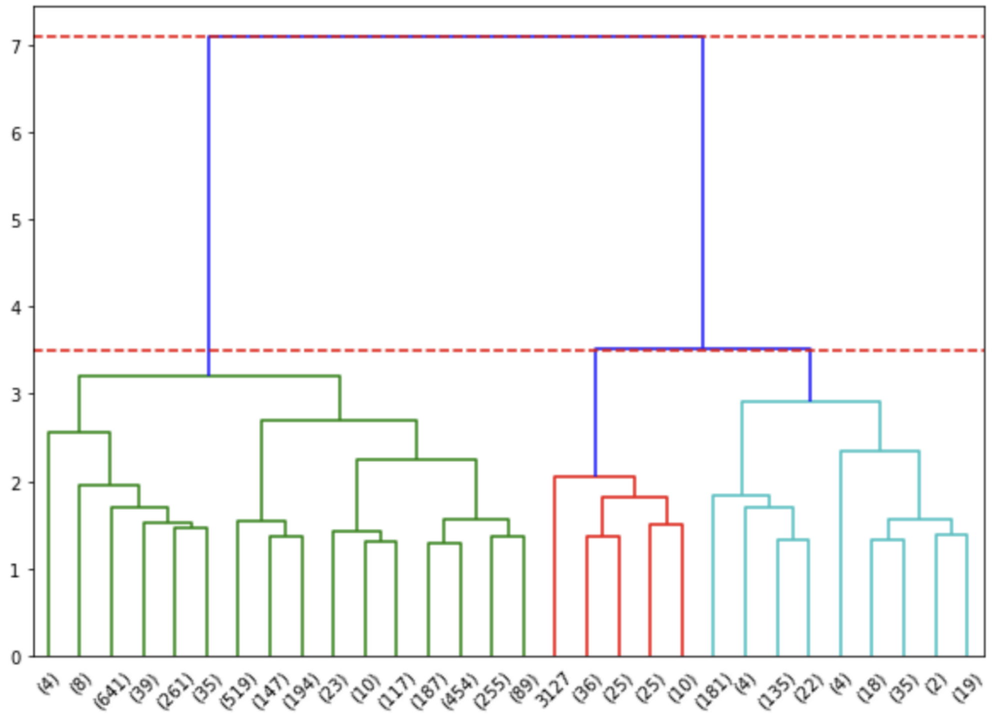
From the above figure, we can clearly see that all the points got clustered into the same cluster and only few of the points got clustered into the other two clusters. Vividly, this linkage choice is not appropriate for this kind of dataset. Now let us have a look at the number of points that got clustered into clusters.



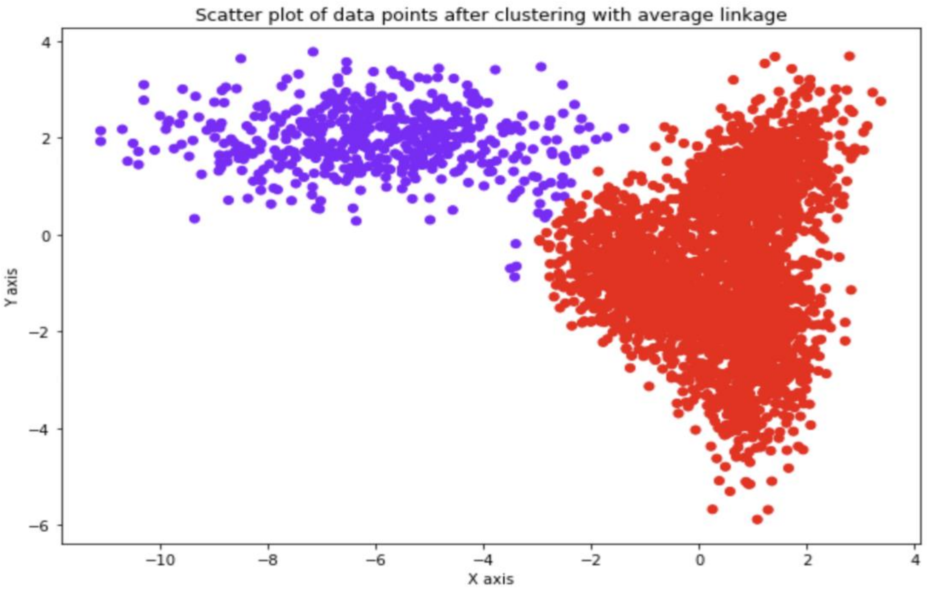
```
Length of first cluster with single linkage:
3498
Length of Second cluster with single linkage:
1
Length of Third cluster with single linkage:
1
```

- **Hierarchical Agglomerative clustering on the 1st Dataset (‘average linkage’)**

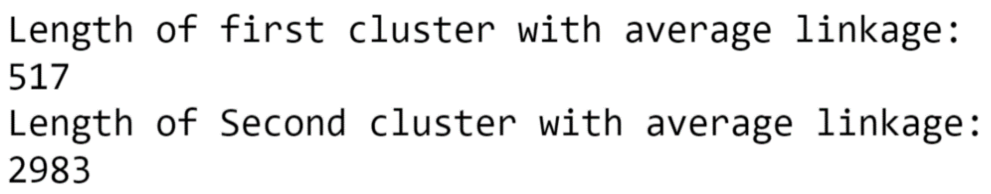
  After implementing the single linkage, I shifted on to the average linkage. Since the theory remains the same, I am just going to show the results for the coming versions of agglomerative clustering. First I drew the dendrogram for the average linkage to see what should be the optimal cut which is produced the same way as described in the previous section. Below is shown the dendrogram for the same:



From the above figure, we can clearly see that there are only two vertical lines passing between the red dotted lines. Hence the optimal number of clusters to be taken here is 2. Hence we apply the algorithm for 2 clusters with average linkage. Below is shown the scatter plot of the points after they are clustered into 2 clusters:



From the above figure, we can clearly see two clusters being formed with the average linkage for the same dataset. The observation that should be made here is the difference between the clustering results when the linkage method is changed. It really depends on problem to problem. For some problems, average linkage might work pretty well whereas for some single linkage might do the work appropriately. Below are shown the number of points that got clustered into different clusters:

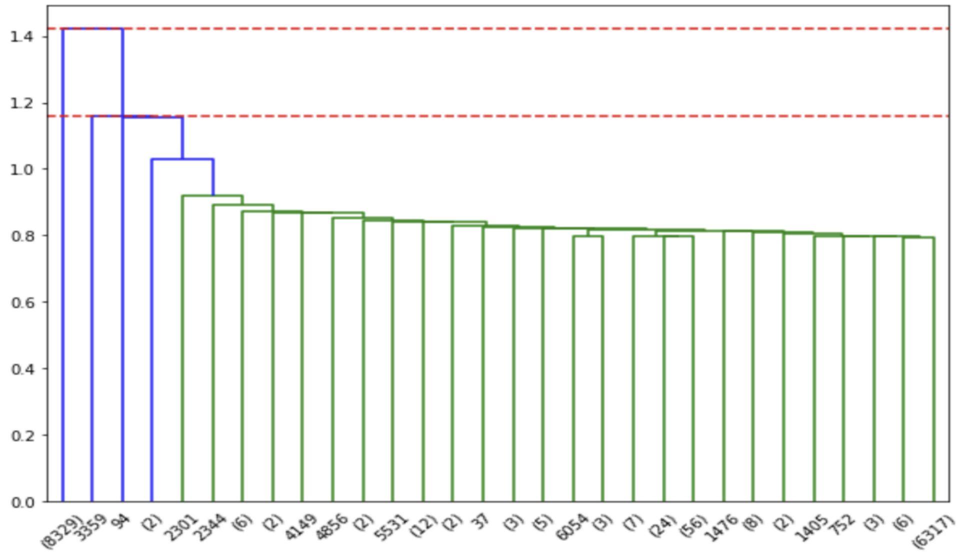


```
Length of first cluster with average linkage:
517
Length of Second cluster with average linkage:
2983
```

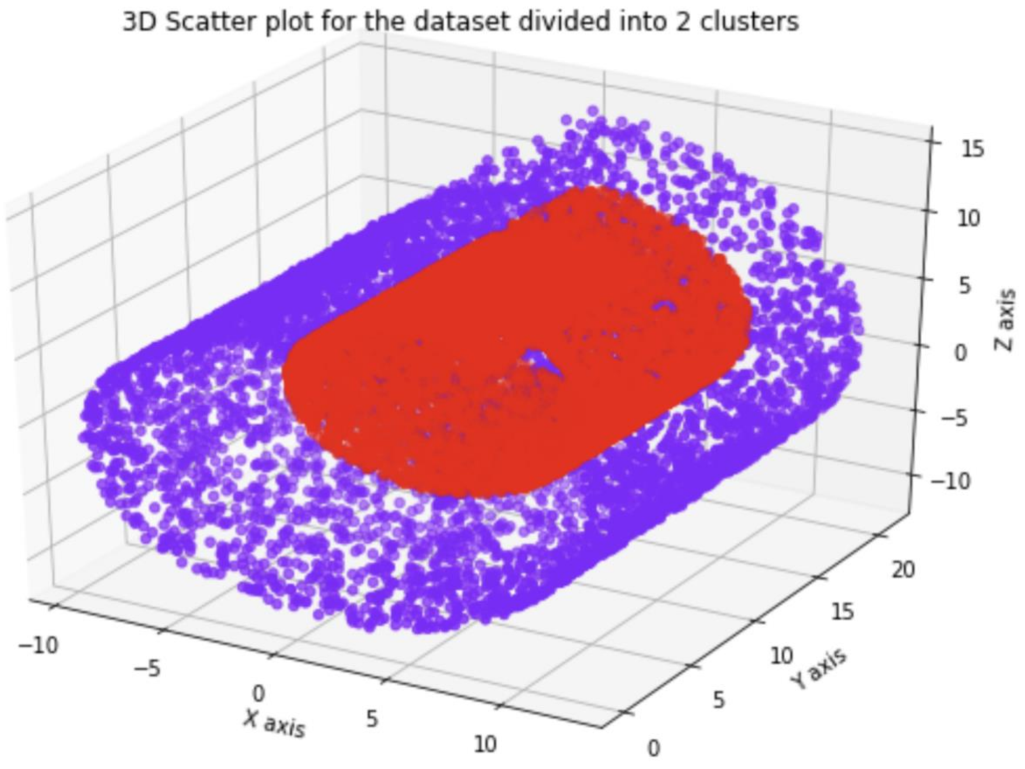
From here, we can notice one slightly common observation by considering both the linkage methods that more than 70 percent of the points got clustered into just one cluster for the second linkage method. Even in the first method, all the points were clustered in one cluster. Hence one cluster is domination over the other or the data has this particular nature where one of the clusters is much larger than the other.

- **Hierarchical Agglomerative Clustering for 2nd Dataset('single linkage')**

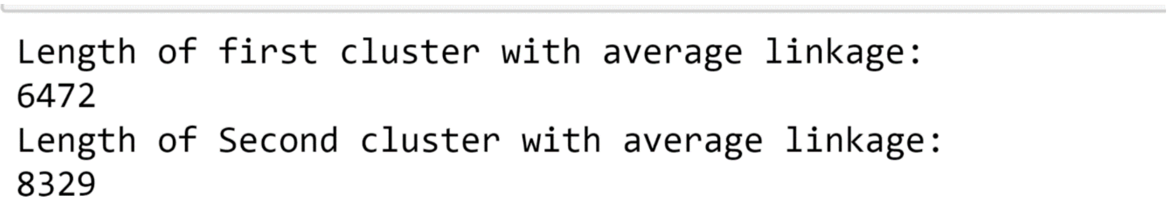
  Now the same procedure was followed for the second dataset as well. The second dataset is three-dimensional dataset consisting of 14,801 data points. First we consider the case of the single linkage method for the hierarchical agglomerative clustering. As usual, I first drew the dendrogram to get the optimal cut for the algorithm. Below is shown the dendrogram for the following algorithm:



From the above figure, we can clearly see that there are two number of vertical lines passing between the dotted red lines. Hence the optimal cut for the following algorithm with single linkage is 2. So we will be dividing the whole dataset into 2 clusters. Below is shown the 3D scatter plot for the same after dividing it into 2 clusters:



The above shown is the 3d scatter plot for the clustering done with the single linkage method. This method, unlike the previous dataset, very neatly clusters the points into different clusters. The data points seems to be a of 2 clusters in the form of concentric cylinders. Unlike the previous dataset, single linkage has very neatly clustered the points into different clusters. Now let us have a look at the number of points which got clustered in each cluster. Below are shown the results:

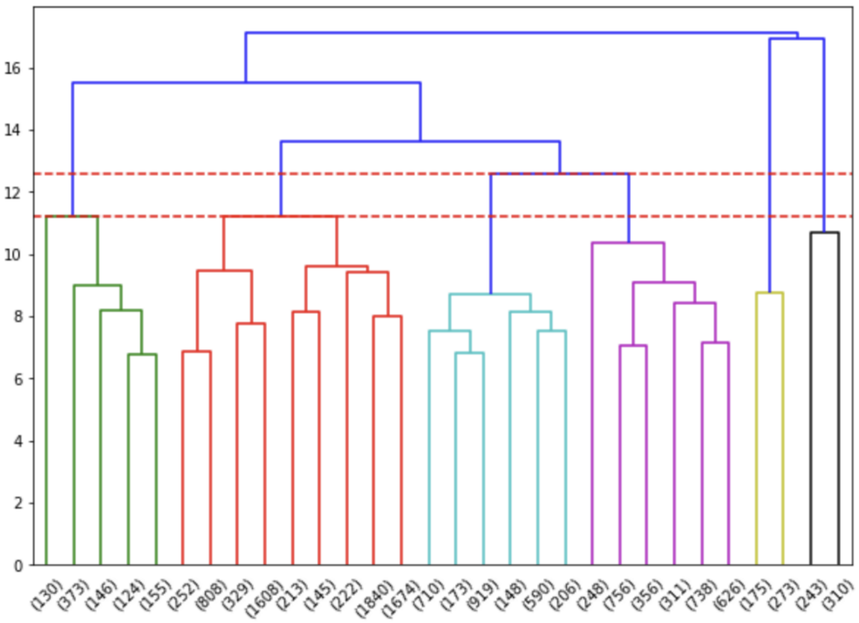


```
Length of first cluster with average linkage:
6472
Length of Second cluster with average linkage:
8329
```

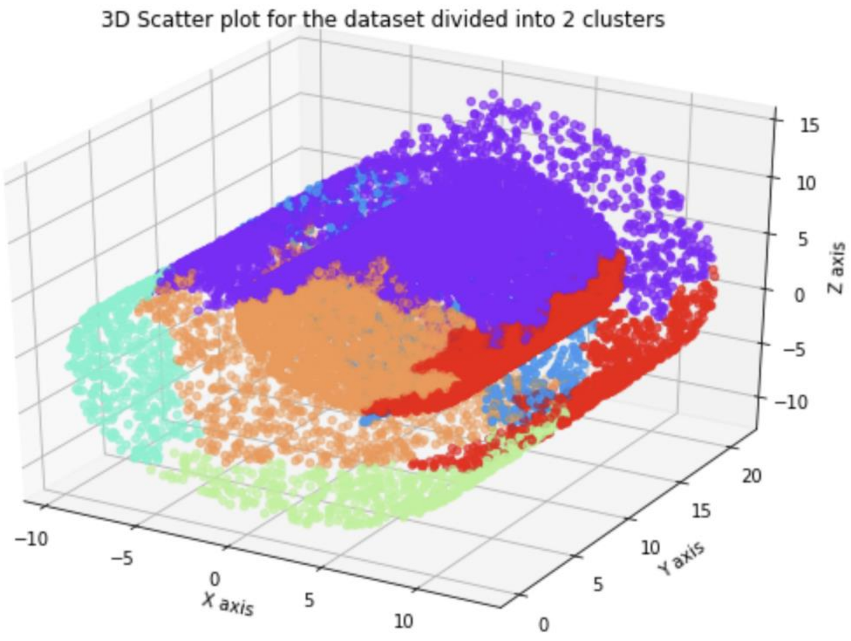
Hence, we can conclude that the number of points clustered are almost in the ratio of 60:40. And looking at the 3d scatter plots, the points have been very neatly clustered into the different clusters.

- **Hierarchical Agglomerative Clustering on 2nd Dataset ('average linkage')**

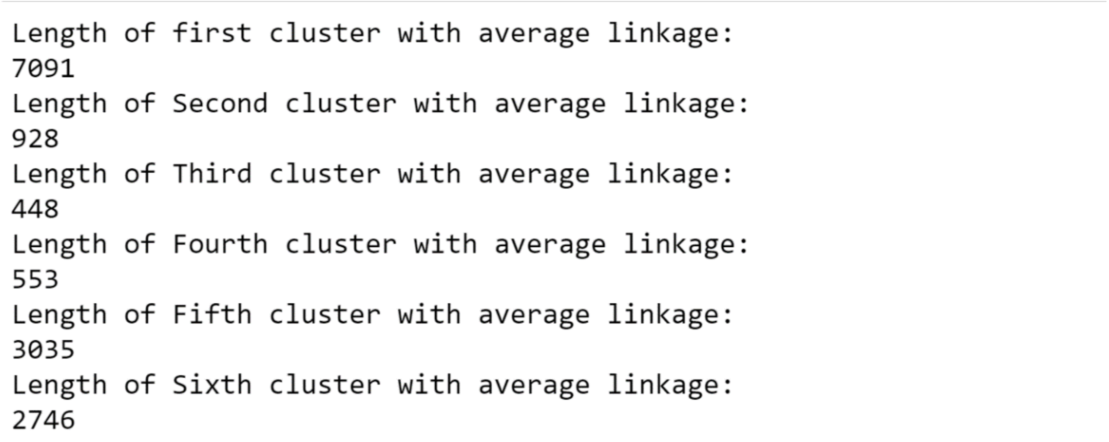
  Finally, we have just the one version left where we applied the average linkage hierarchical agglomerative clustering on the second dataset. Following the procedure as in the previous sections, I drew the dendrogram for the following data points linked via the average linkage method. Below is shown the dendrogram for the same:



From the above figure, it is clear that there are 6 vertical lines passing between the two red dotted lines that I made. Hence the optimal clusters we will be going with will be 6. So we set the cluster number equal to 6. After this, we made a 3d scatter plot for the datapoints after dividing them into their respective clusters. Below is shown the 3D scatter plot for the following dataset after clustering them into 6 clusters:



The above figure shows the 3D scatter plot for the data points of the second dataset after getting clustered into 6 clusters. Since the number of clusters are way too much and it is a 3D representation of the same, it is becoming a bit difficult to comprehend or visualize the different clusters. For a bit more clarity, i am showing below the number of points that got clustered into each of the clusters:



```
Length of first cluster with average linkage:
7091
Length of Second cluster with average linkage:
928
Length of Third cluster with average linkage:
448
Length of Fourth cluster with average linkage:
553
Length of Fifth cluster with average linkage:
3035
Length of Sixth cluster with average linkage:
2746
```

From the above results, we can conclude that the average linkage method seems to be doing a fair job in clustering and the optimal cuts that we received from the cut method discussed above seems to give us good results as the points are getting neatly clustered if the right methodology is used for the problem.