ECE 592: IoT Analytics

PROJECT 4: Clustering

STUDENT ID: 200203773
STUDENT NAME: ESHAAN VENKAT KIRPAL

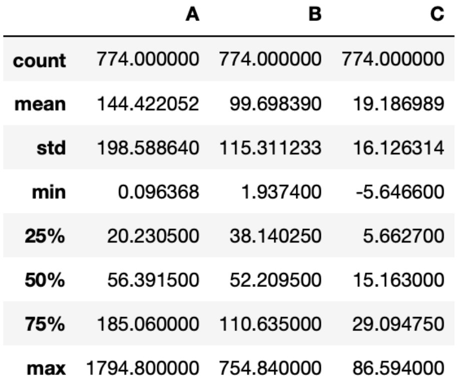
NOTE: There are four tasks in this report and each task is further divided into sub-sections.

SUMMARY OF THE INPUT DATA:

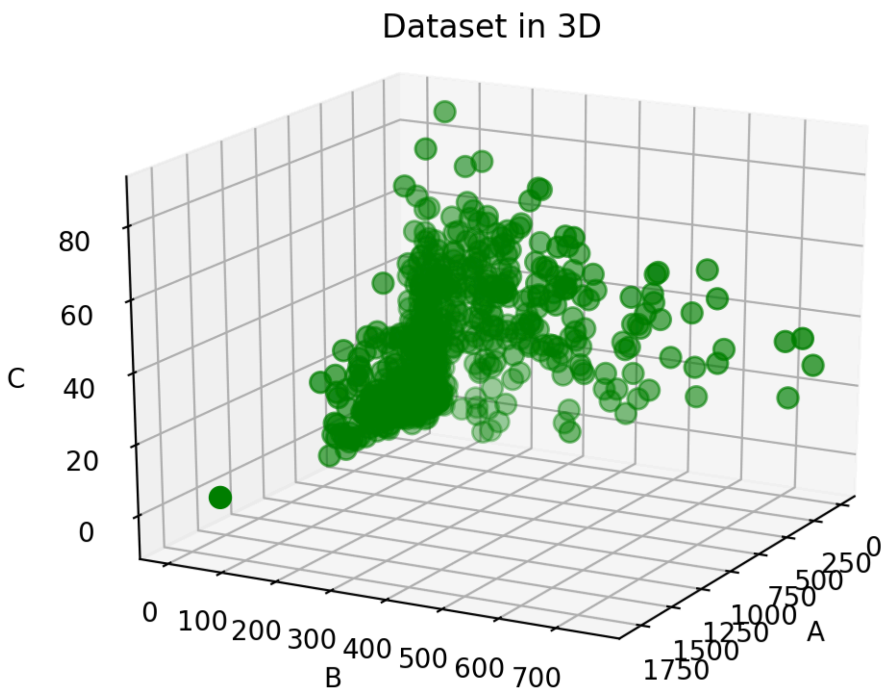
•The input data had 774 3-tuple observations. The data did not seem to have any specific shape, it is centered above the (0,0,0) point and extends across the three dimensions A, B, and C in positive direction. Below is the scatter plot of the data in 3D.

•It looks like the entire dataset is one cluster though we can also have two considering that one set of points extends in the B axis and the other extends int he A axis. We will infer the optimum number of clusters using several clustering techniques.

- The three features (A,B, and C) have different range of values. Thus it seems reasonable that we normalize the data before applying it to our algorithms, so that no feature overpowers the other because of its larger values. We will not be doing so during the hierarchical clustering as original values make it easy to plot the dendogram.



	A	B	C
count	774.000000	774.000000	774.000000
mean	144.422052	99.698390	19.186989
std	198.588640	115.311233	16.126314
min	0.096368	1.937400	-5.646600
25%	20.230500	38.140250	5.662700
50%	56.391500	52.209500	15.163000
75%	185.060000	110.635000	29.094750
max	1794.800000	754.840000	86.594000

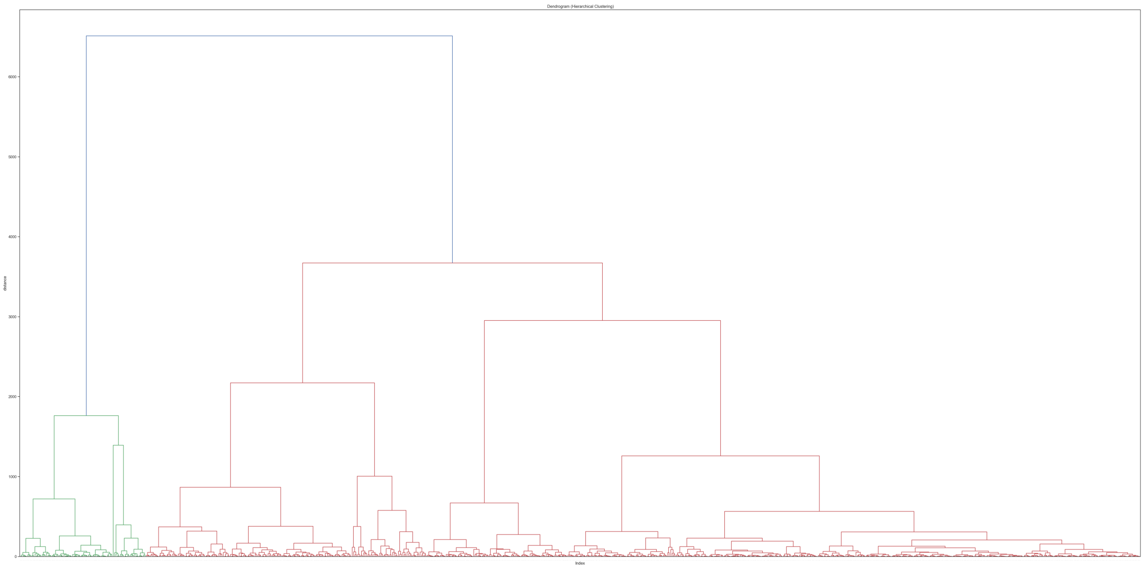


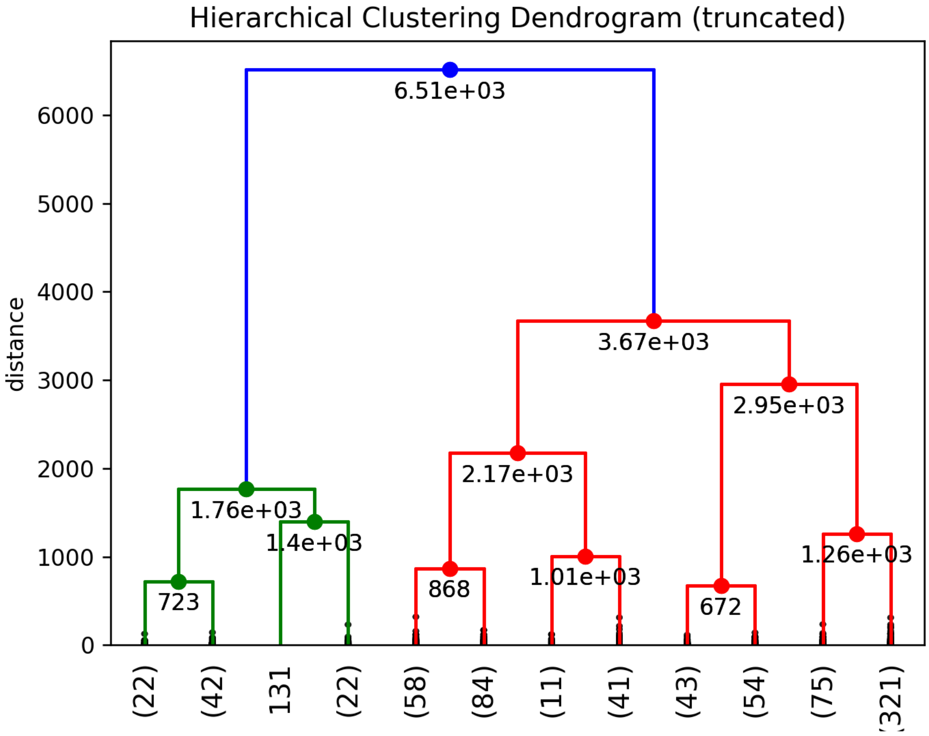
TASK 1: HIERARCHICAL CLUSTERING

SECTION 1.1: PLOT THE DENDROGRAM AND THE DISTANCE GRAPH

We use the euclidean distance as the metric and Ward algorigthm for linkage between the clusters.
The below dendrogram shows the distances at which the clusters merge. We can see from the plot that major clusters are differentiated automatically by color, one is in red while the other is in green, while the top most cluster in blue joins the two.

We can infer better by looking at the truncated graph, which shows the last 12 merged clusters and its distance.



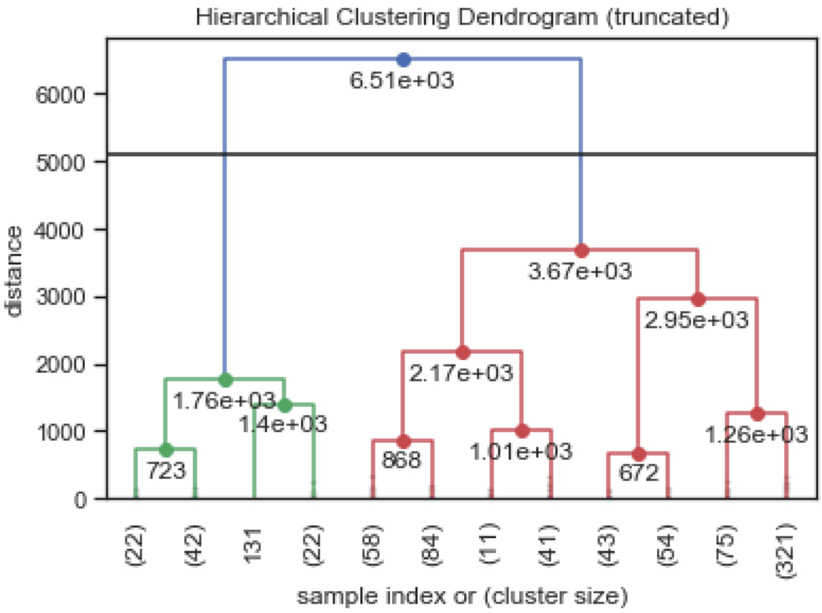


SECTION 1.2: DETERMINE THE NUMBER OF CLUSTERS.

It is evident from the dendrogram that the maximum jump in the cluster merges happen between D1 = 3670
D2 = 6510
So we need to consider the merges above (D1 + D2)/2, which leads us to find the count of clusters.

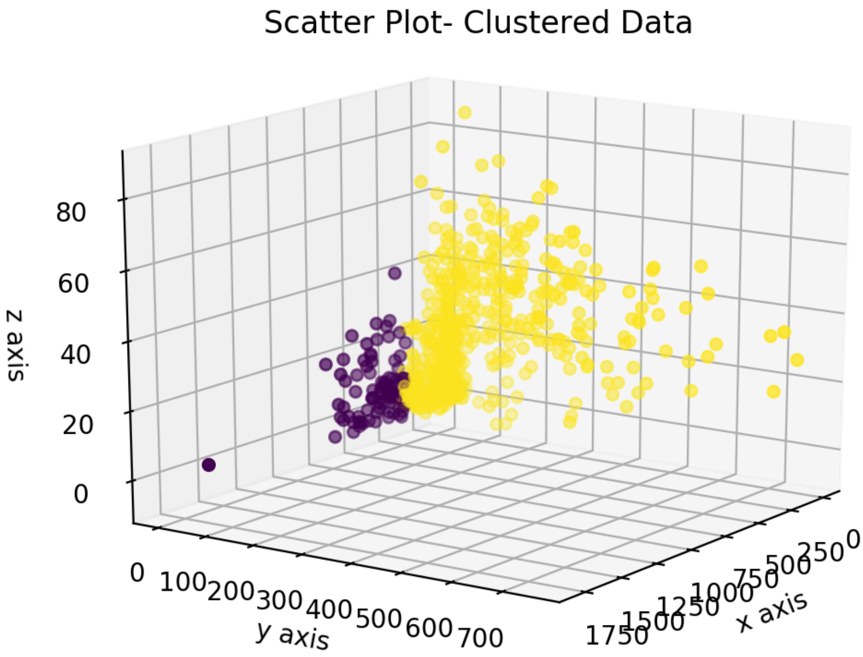
Optimum distance at which we get the best partition = (D1 + D2)/2 = 5092

We next plot the line that splits the dendogram at distance 5092 and creates separate clusters. These two clusters



SECTION 1.3: COLOR THE DATA ACCORDING TO THEIR CLUSTER, AND DO A 3D SCATTER DIAGRAM

Below we see the dataset segregated into two clusters using the hierarchical algroithm. Data points along the y-axis (B) form one cluster while data points along the x-axis(A) form the other cluster.

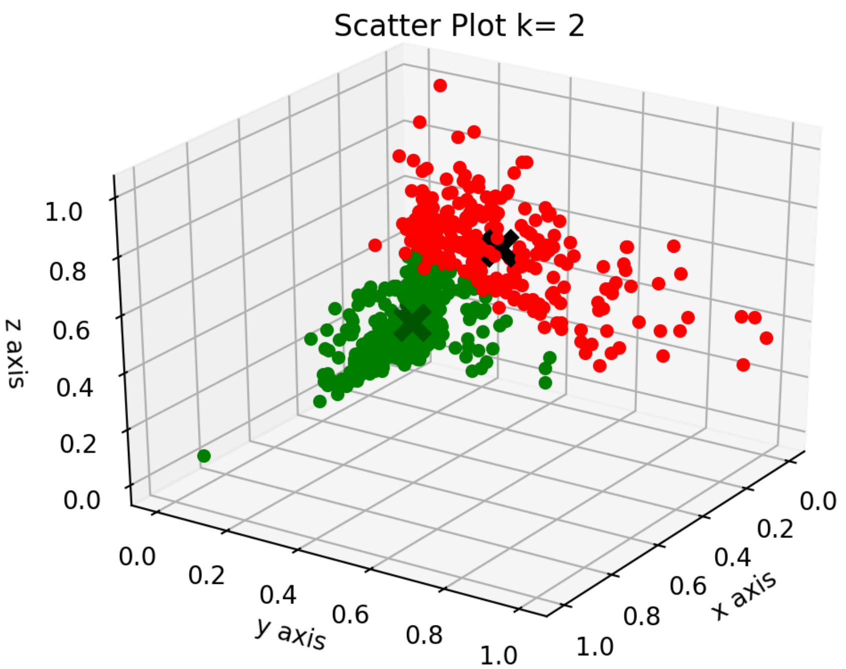


TASK 2: K-MEANS CLUSTERING

SECTION 2.1: APPLY THE ALGORITHM FOR SEVERAL VALUES OF K STARTING WITH K=2.

K-means clustering will take as parameter the number of clusters (K) and accordingly classify the points into those clusters. It uses a distance metric to classify points into clusters and is usually a good algorithm for globular structured data.

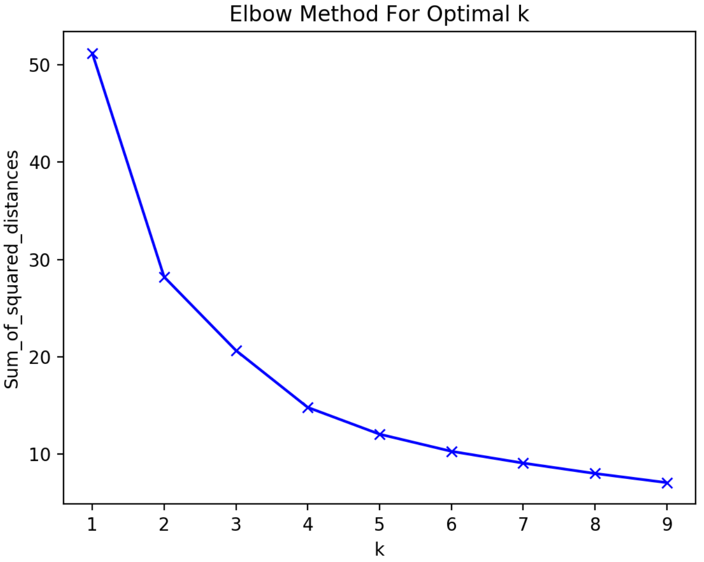
Let's take k =2 and plot the k-means clusters in 3D.



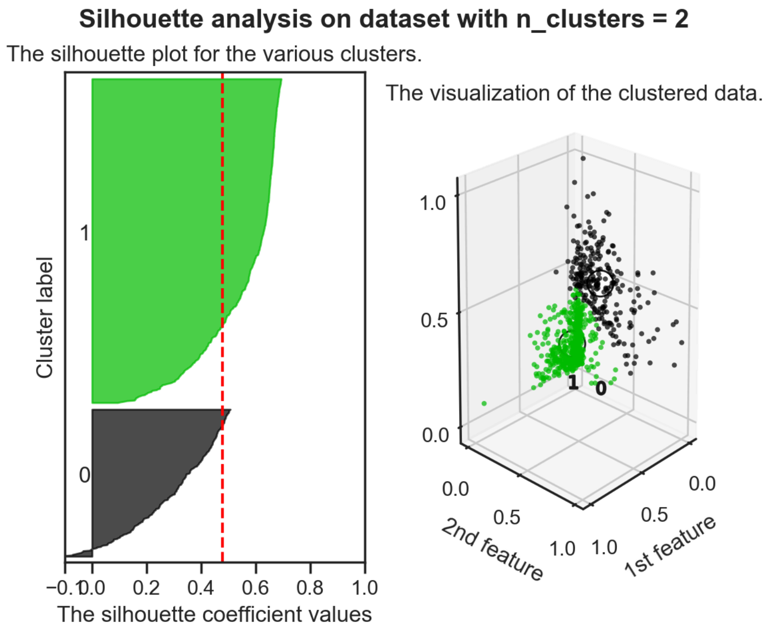
Here the centroids are marked with a black X.
As earlier mentioned, K-means has a tendency to give spherical clusters with an objective to minimize the squared sum of distances from the centroids. So let's change the k –values and see if we can do better.

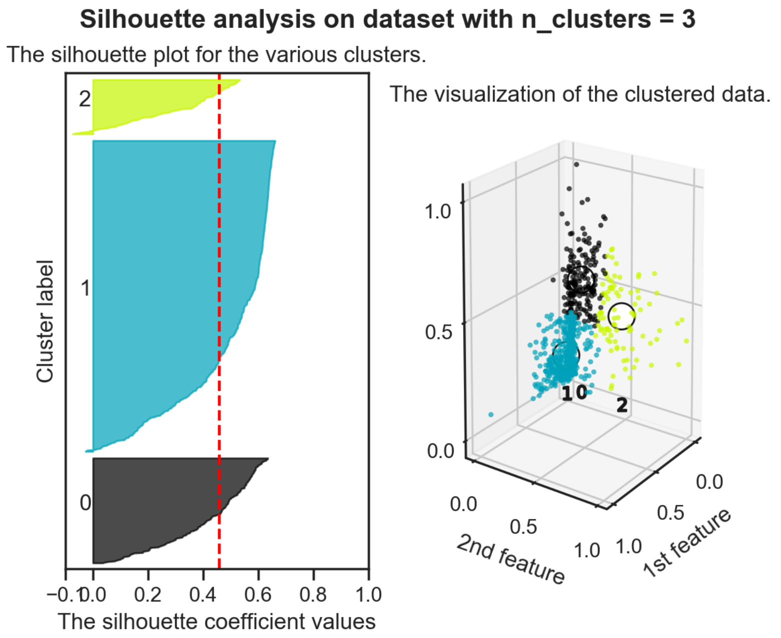
SECTION 2.2: USE THE ELBOW METHOD TO DETERMINE THE BEST VALUE OF K.

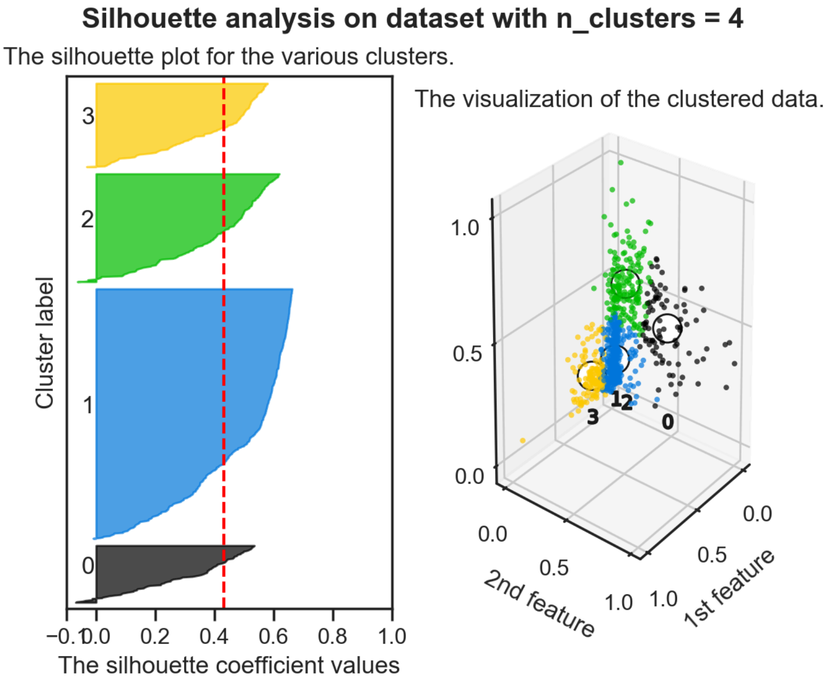
We use the elbow method to find the best value of K. Here, we plot the squared sum of distances from the centroids for the data points against different k values.

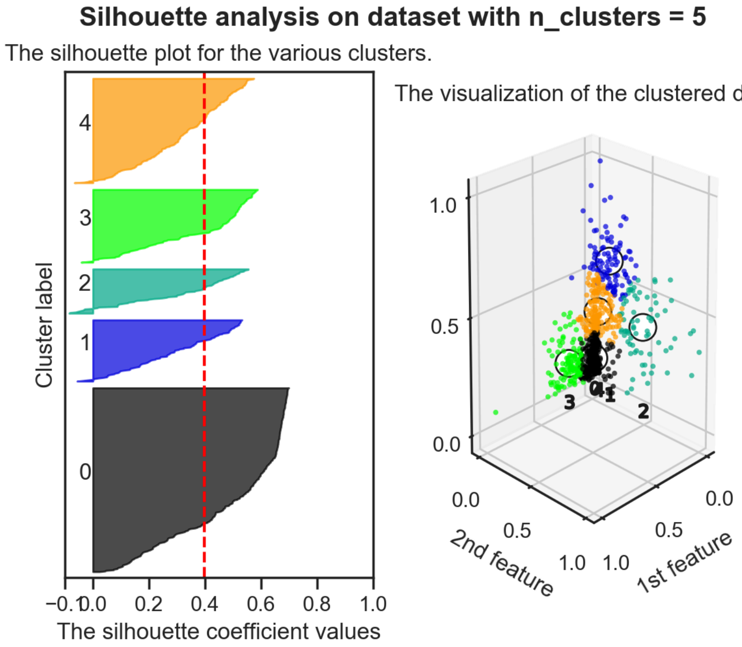


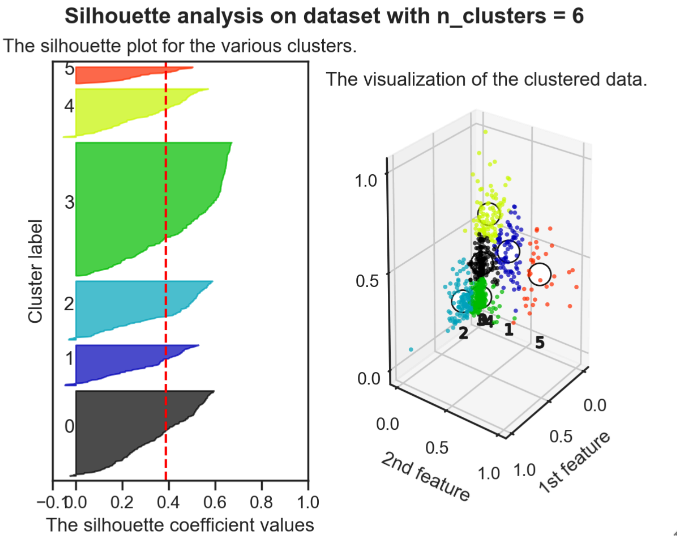
It is evident from the plot that it is a smooth curve and we don't have a clear elbow. This might be indicative that the data does not cluster. Choosing a K from this plot is not correct. Instead we next use the silhouette scores method for determining K.

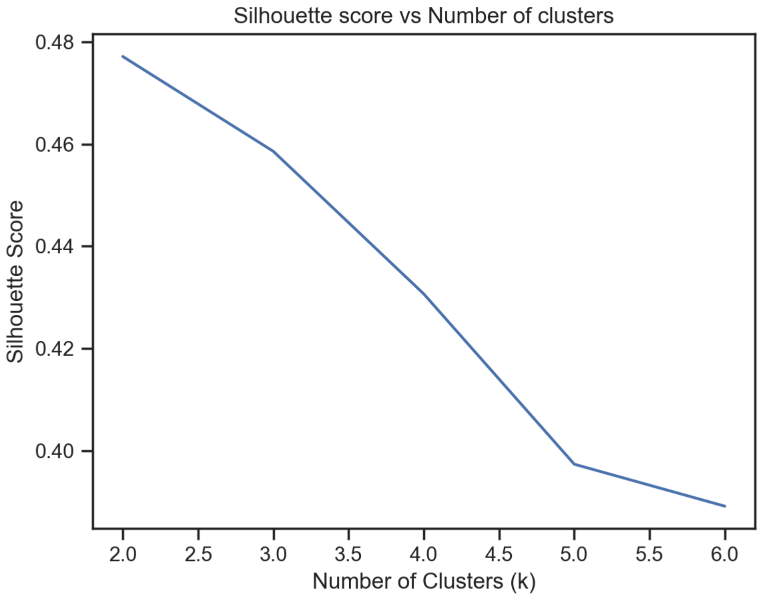








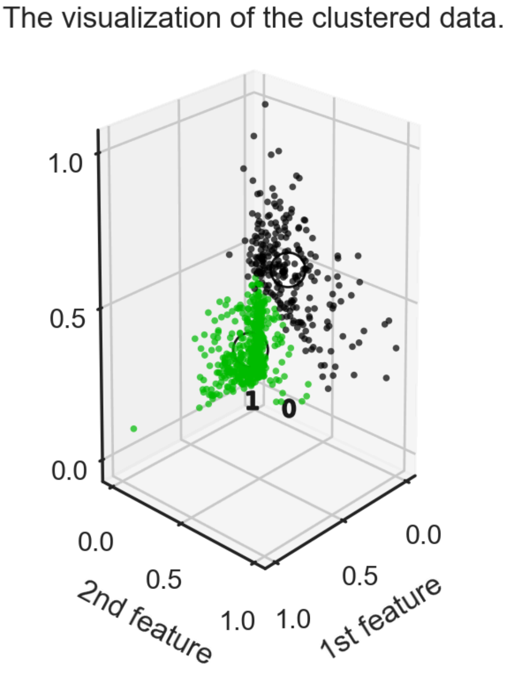




The higher the silhouette score the better the clustering, i.e. a value closer to 1 indicates that the data point is far away from the neighbouring clusters and a value closer to 0 indicates the data point is on or very close to the decision boundary, and negative value indicates the data point has been assigned to the wrong cluster.

In the plots above, on the left are the silhouette plots for the various clusters while on the right the 3D visualization of the clustered points.
We can see from silhouette plots above for different K values, that none of the clustering is perfect and that some points are clustered in the wrong cluster. Clustering with K=2 and K=3 are good as not many points are wrongly clustered.
Moreover, from the average silhouette score line plot above, we can see that at K=2 we get the best silhouette score of 0.478. This suggests that the dataset is best clustered using two clusters. This inference is similar to what we learned after hierarchical clustering.

Now, let's plot the 3D scatter diagram for the best value of K, i.e. K=2. The centroids of each cluster is marked by big black circles.

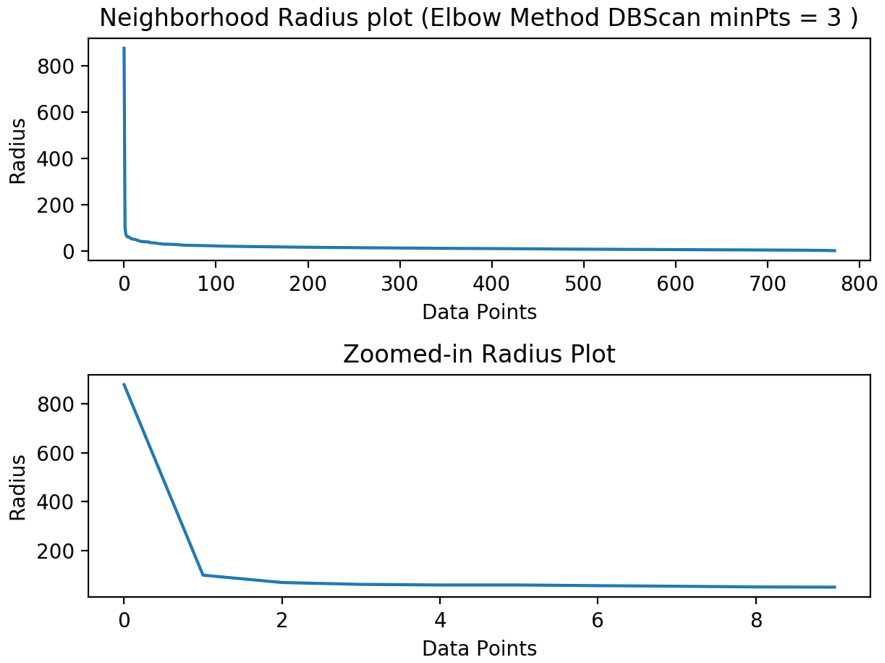


TASK 3: DBSCAN CLUSTERING

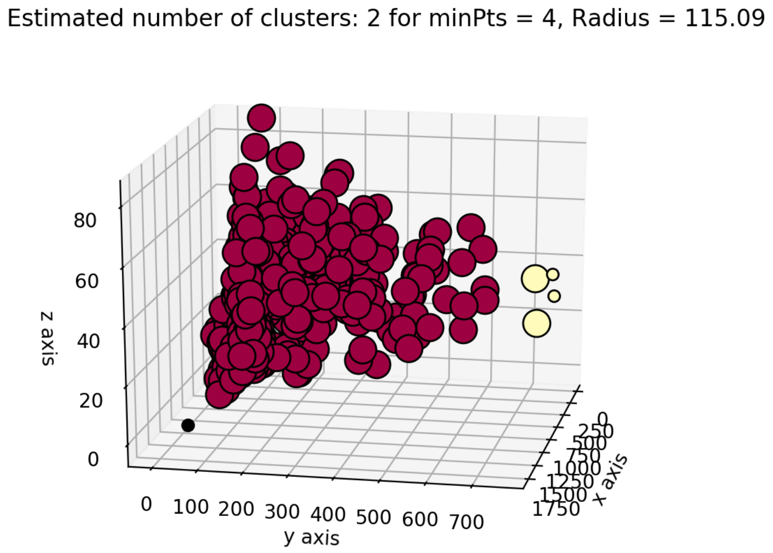
SECTION 3.1: FOR MINPTS=3, USE THE ELBOW METHOD TO DETERMINE THE BEST VALUES OF *RADIUS*.

DBScan does a density based scanning and separates the core points, boundary points and the noise data points. It does not bridge the clusters based on distances but also checks for a threshold on density of the points. It is a good clustering mechanism for data with uneven cluster sizers and those that have specific shapes. Though it is bad when the dataset has varying densities.

We sequentially increase the Minpts value and check manually the classification of the data points. For each Minpts we find the minimum radius r_i. We found from the generated plots that the data is best classified with Minpts=3 and with Minpts =4. The radius is found from the elbow point in the respective radius plots, it is 100.84 and 115.09 for the two cases respectively. [All plots are in a separate file]



We take Minpts as 4 instead of 3 as higher values of Minpts provides us a denser cluster which is more robust to noise points. Thus, taking neighborhood radius = 115.09 and minPts as 4, we perform a DBScan, the result is as below:



We see from the scatter plot above that one noise point is identified in black and two clusters are identified, one in yellow, while the rest of the data in red forms a dense cluster. Beyond minpts=4 the algorithm fails to identify the yellow cluster and creates just one red cluster, as can be seen from the plots in the DBSCAN.pdf file.

This is expected since with increasing value of radius, more and more data points have the opportunity to cluster together. Hence we see this result. But for very high values of neighborhood radius too, the cluster representation will not be that perfect and we see the whole data being represented as a single cluster.

Though from manual inpection of our dataset, we can note that those yellow points aren't as densely clustered as the red points and thus have been separately identified when Minpts is 3 or 4.

TASK 4: COMPARISON OF THE DIFFERENT CLUSTERING METHODS

From manual inspection we can say either the dataset is one big cluster with a few noise points or that there are two clusters, one along the y-axis while the other cluster with data points along the x-axis.

Hierarchical Clustering: It shows that there are two clusters in accordance with the visual inspection. This algorithm is sensitive to noise which can act as bridging points for merging clusters, though here we don't see any bridging points.

K-means: The algorithm clusters data by trying to separate samples in n groups of equal variance. Using the silhouette scores method, we are able to figure out value of k for which the average silhouette score is maximized and it is for k = 2. Based on that we are able to classify and assign cluster labels to the data points and hence we get two clusters. K-means clusters points in a fashion similar to the Hierarchical clustering algorithm.

DBSCAN: Density based scanning on the given data is also indicating that the density for the data points is not the same everywhere and hence the algorithm spots two clusters. With changing minPts and radius too, the algorithm at best distinguishes two clusters. Though the clustering is uneven and not like the previous two methods. This is expected as this algorithm is specialised for uneven cluster sizes. The clustering obtained is in accordance to our visual inference that the data comes from one cluster.

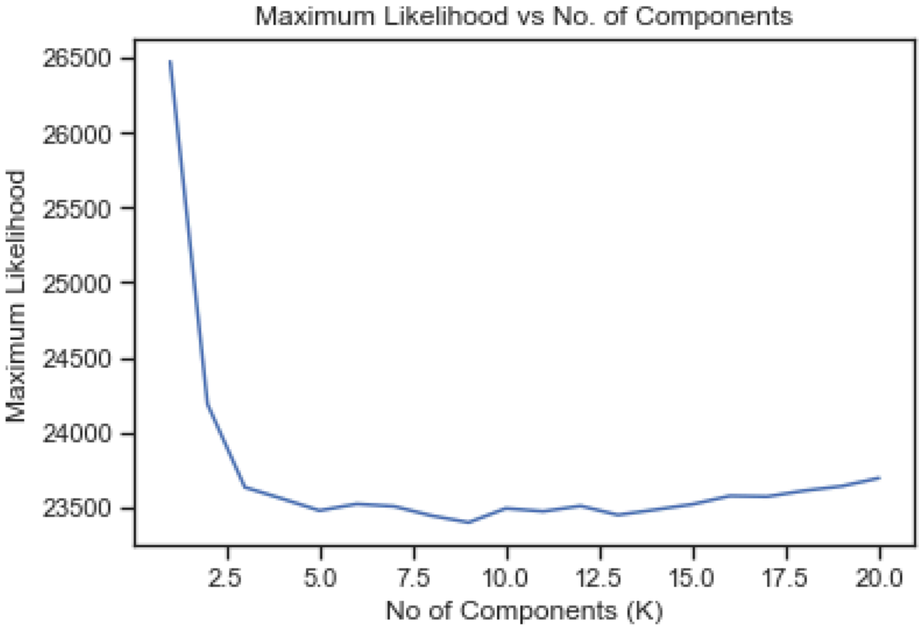
Conclusion:

DB-Scan has done the best job at clustering which supports our visual inspection. It also eliminates noise and identifies non-core points as coming from another cluster.

But we can also say that there are essentially two clusters in the given data points considering distance as the parameters for cluster assignment. K-means and hierarchial both do a similar job in finding two clusters but they both fail to identify the noise points.

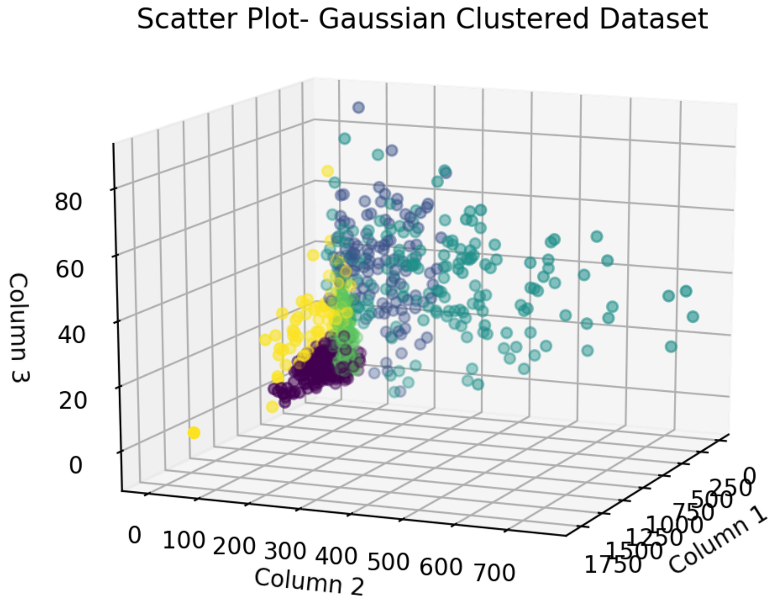
OPTIONAL TASK: GAUSSIAN MIXTURE MODELS (GMM)

SECTION 1: PLOT THE MAXIMUM LIKELIHOOD AGAINST *K* (NUMBER OF CLUSTERS) AND SELECT BEST *K*.

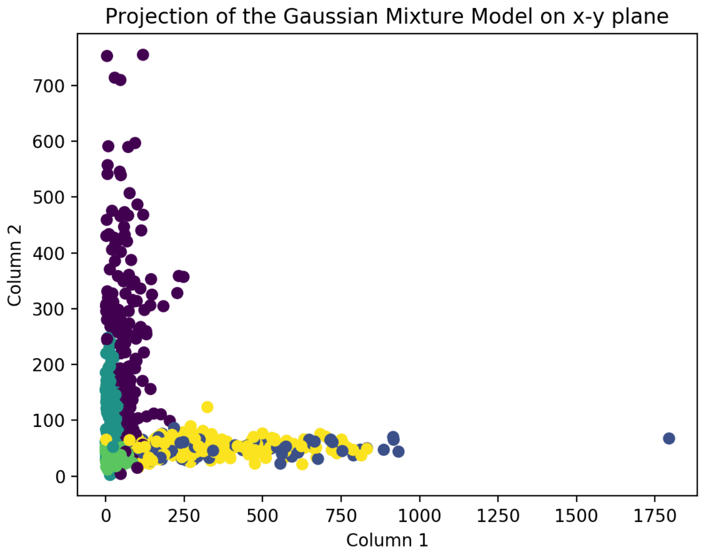


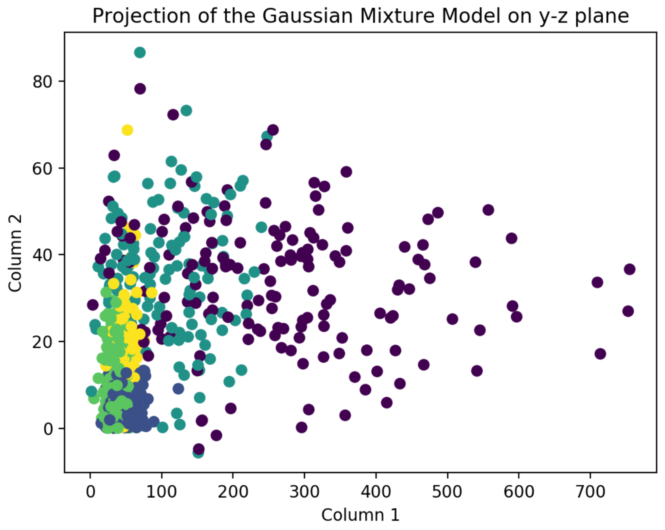
We use the BIC plot to find the optimum no of clusters. Optimum number of clusters =5 as we have a minimum BIC there. We now plot a scatter diagram after fitting GMM with K=5:

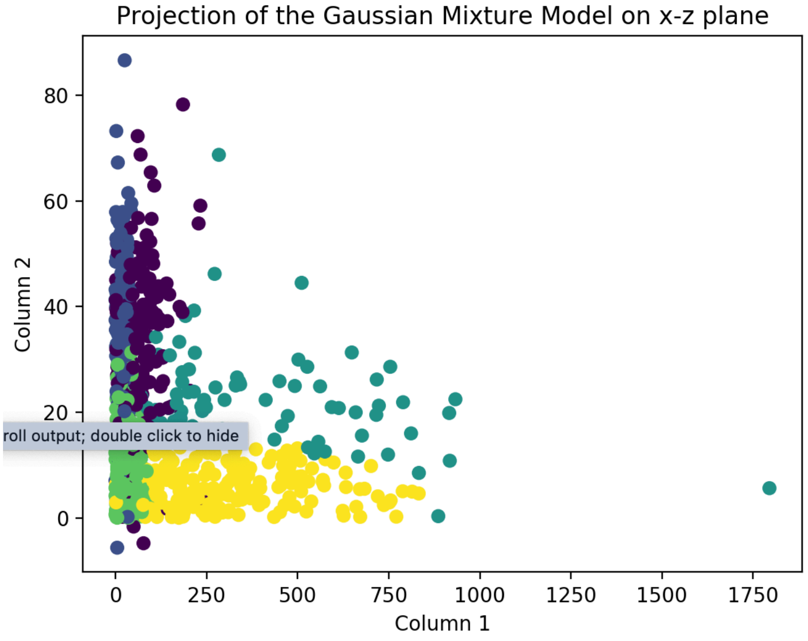
The five clusters are segregated by colors in the figure below.



SECTION 2: OBTAIN A PROJECTION OF THE FITTED MIXTURE OF NORMAL DISTRIBUTIONS ON EACH PLANE


Projection of the Gaussian Mixture Model on x-y plane


Projection of the Gaussian Mixture Model on y-z plane


Projection of the Gaussian Mixture Model on x-z plane

SECTION 3: COMPARE YOUR RESULTS TO THOSE OBTAINED ABOVE IN PROJECT 4 AND DISCUSS YOUR FINDINGS.

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters.

GMM models are good for clustering flat geometric data, i.e. data that don’t have any specific shape. Here with the given dataset, it identifies that the data came from 6 different gaussian distributions and it failed to identify the noise points. Basically, data with different variance seems to have been identified as a separate cluster.

This clustering is not in accordance to what we inferred visually and is totally different to the clusterings obtained using the DBSCan algorithm.
Though the clustering has stark similarities to the K-means clustering when K is selected as 5 as can be seen from the plots in the K-means section.