Project 4: Clustering

Tasks 1. Hierarchical clustering Apply the hierarchical algorithm to the dataset.

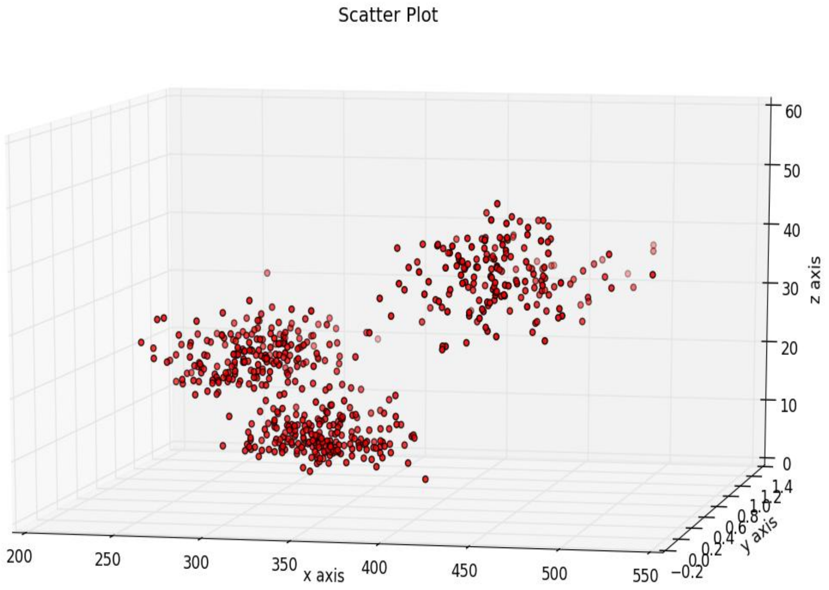
1.1 Plot the dendrogram and the distance graph (if it is given by your package) – **Not supported** (only matrix is supported)

1.2 Determine the number of clusters.

1.3 Color the data according to their cluster, and do a 3D scatter diagram. Rotate the diagram to identify the clusters.

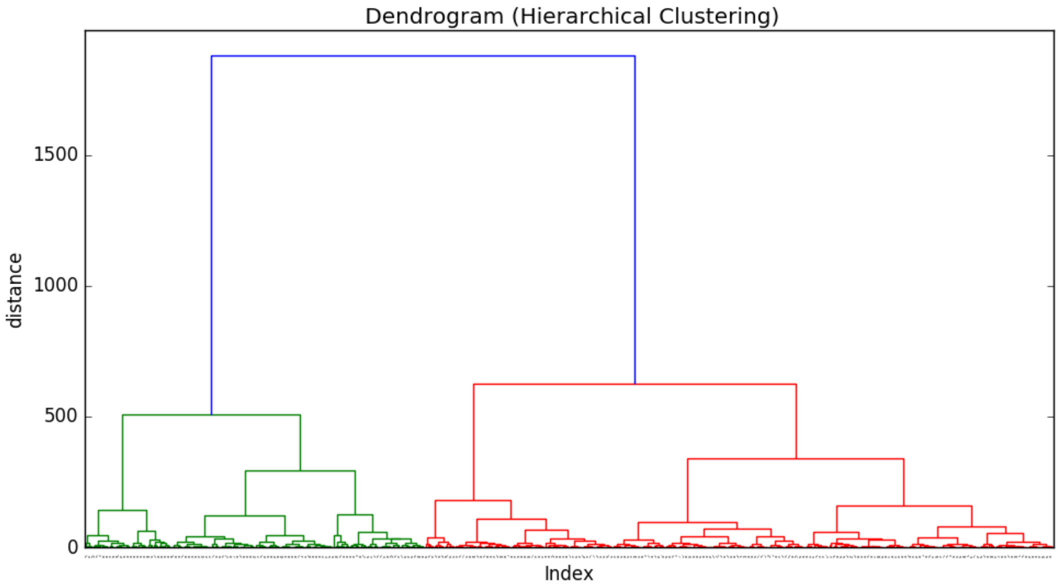
Solution:

Let's make the scatter plot of the data points in 3D:

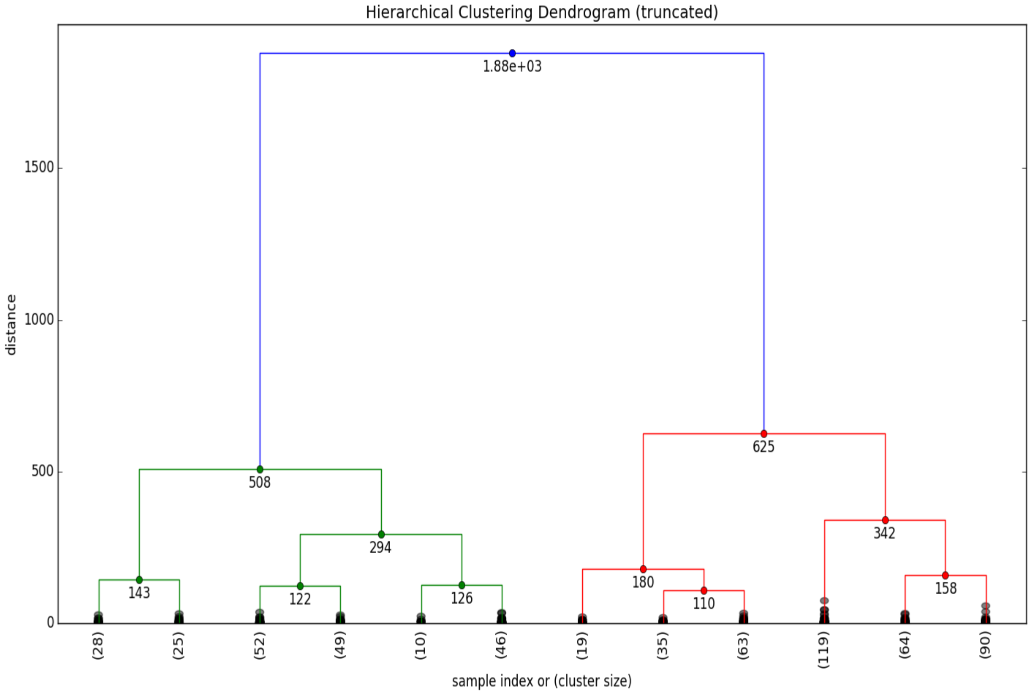


The red dots represent the data points. It looks like there are three clusters. But we need to analyze and infer if that is true. Let's see what results are given by hierarchical clustering.

1. Making a dendrogram for the given data points



This dendrogram shows the distances at which the clusters merge. But let’s truncate the lower merges and mark the distances at which merges happen to help us analyze better



It is evident from the dendrogram that the maximum jump in the cluster merges happen between

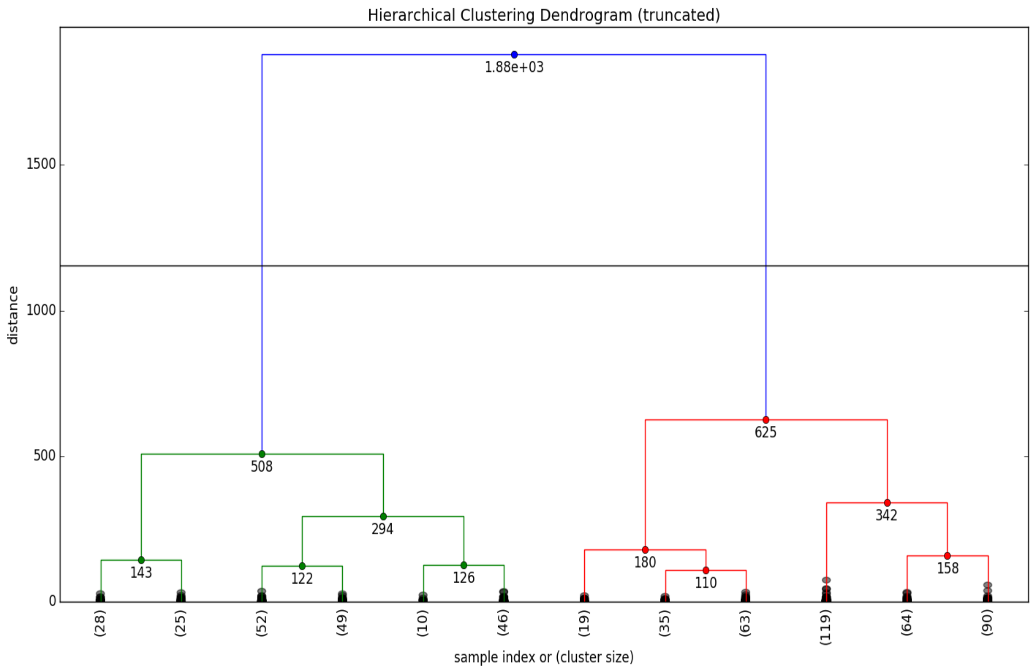
D1 = 625

D2 = 1880

So we need to consider the merges above (D1 + D2)/2, which leads us to find the count of clusters.

D1 + D2/2 = 1155

Marking the line in the dendrogram

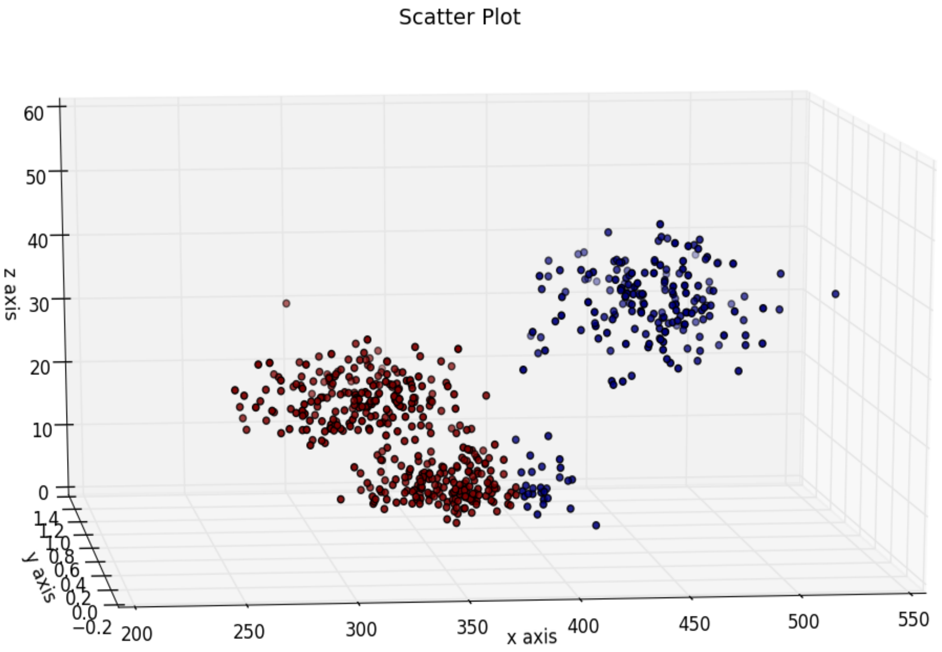


Comments:

Two clusters merge above this distance.

Hence, we have two clusters here according to the dendrogram analysis.

Plotting the clusters in 3D, we get



Two clusters are identified here and marked in brown and blue colors. But some blue bubbles marked at bottom with the brown bubbles look to be assigned incorrectly.

Task 2

k-means clustering

2.1 Apply the algorithm for several values of k starting with k=2.

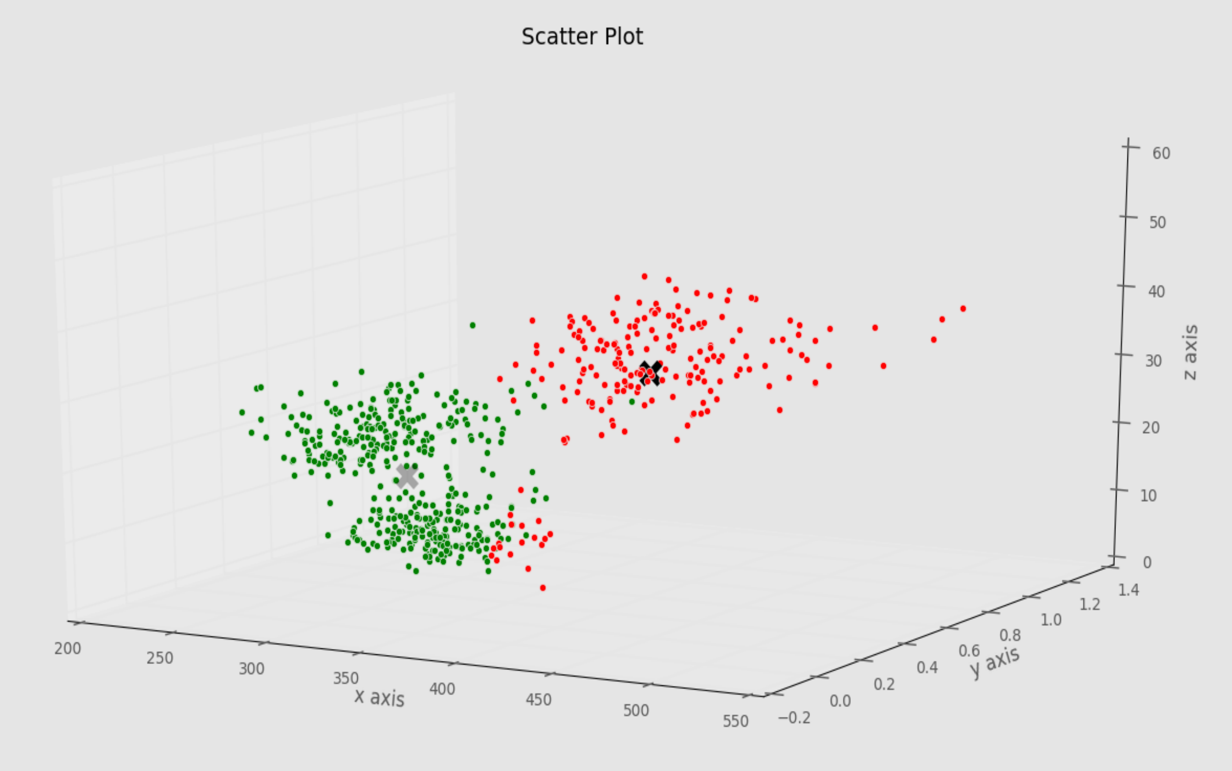
2.2 Use the elbow method to determine the best value of k.

2.3 For the best k value, color the data according to their cluster, and do a 3D scatter diagram. Rotate the diagram to identify visually the clusters.

Solution

K-means clustering will take a k value, the number of clusters and accordingly classify the points in those clusters

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



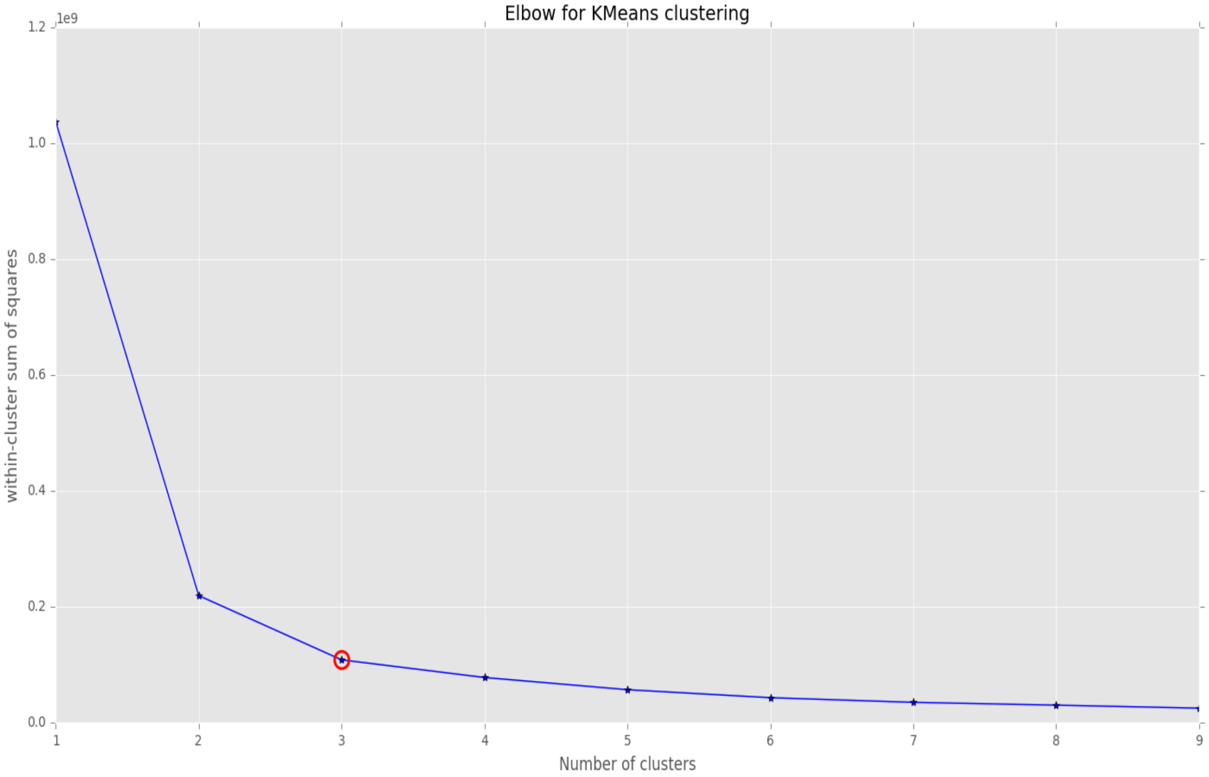
Here the centroids are marked with a black X. In case of green clusters above, it is slightly faded indicating that it is not in the foreground (this is the best representation we can do with 3D plots)

From the submitted code, count of data points in each cluster can be acquired.

401 points are labelled in green cluster and 199 points are labelled in red cluster.

The 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 look at different k –values and see if we can do better.

We plot the squared sum of distances from the centroids for the data points against different k values. Let’s analyze using elbow method

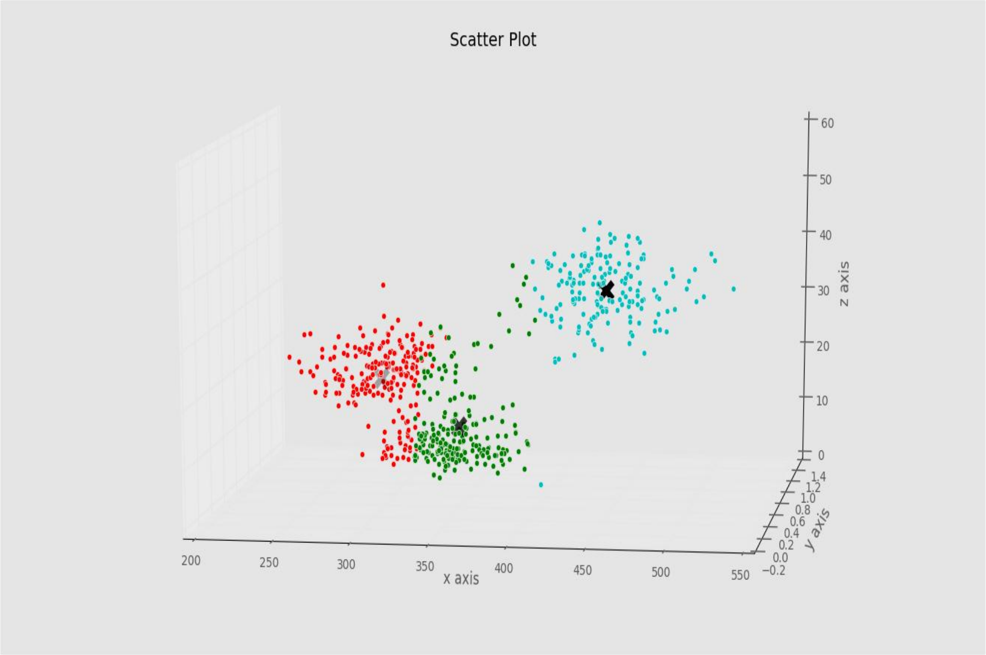


Comments & Analysis

Let SSE = squared sum of distances from the centroid

From the plot, we can see that there arises a big difference in SSE value when we switch k from 1-2. But switching k from 2-3 also has a significant change in the SSE value. Later it doesn’t change much. If we take k = 3 and find the cluster labels, we find that that the data points are divided in spherical clusters of about same size (of sizes 201, 206,173).

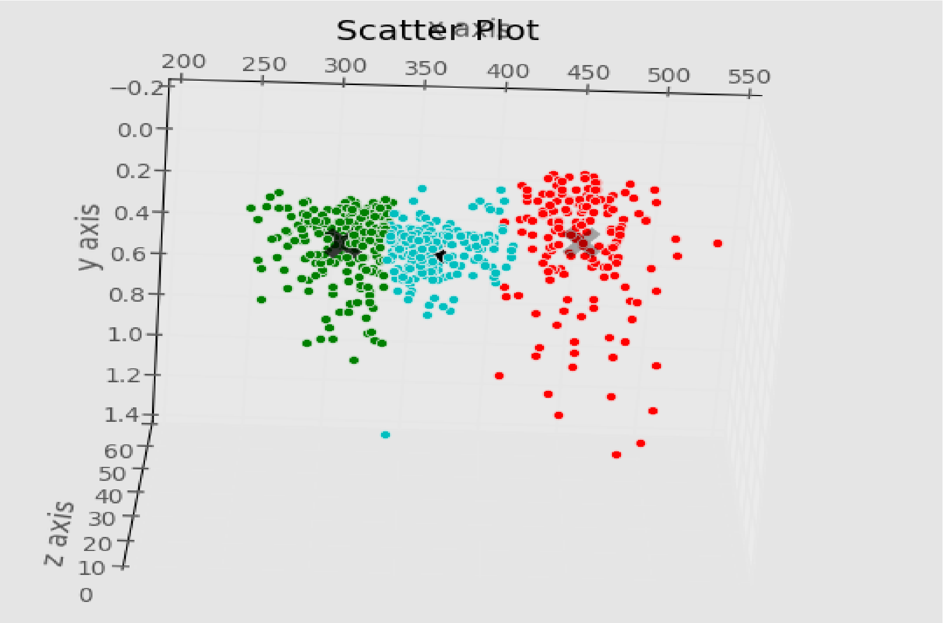
So according to the elbow methods and general property of k-means to divide the data set in spherical clusters of about same size, we choose, k =3. So, three clusters can be seen here. They are plotted in 3d below.



The three clusters here are marked in red, green and blue. But the k-means has not done a very good job of separating the data points. Since some red bubbles and green bubbles are not assigned correctly (visually)

Nevertheless, it has the SSE value lesser than that for k=2 and shows a sharp elbow in elbow method curve.

Here is a plot of the three clusters from a different angle



Task 3

DBSCAN clustering, Apply the DBSCAN algorithm to the dataset to determine the number of clusters.

3.1 For Minpts=3, use the elbow method to determine the best values of ε. Run the DBSCAN algorithm for the best value of ε and Minpts=3. Color the data according to their cluster, and do a 3D scatter diagram.

Rotate the diagram to identify visually the clusters.

3.2 Repeat the above step for Minpts=4,5,6. (use more values if necessary).

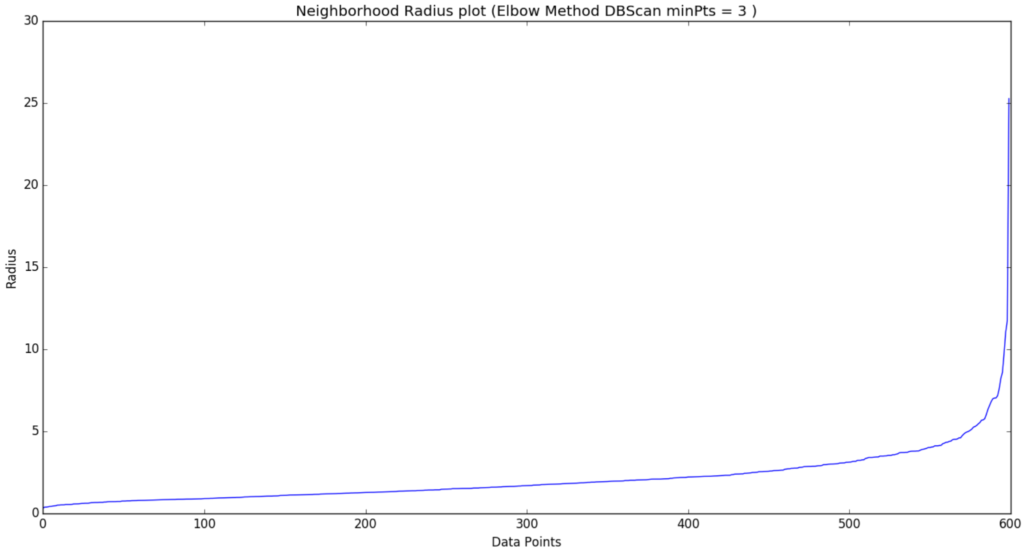
3.3 In your report provide only your best clustering and its 3D scatter diagram. Provide the remaining resuts of your investigation in a separate file

Solution:

DBScan does a density based scanning and separates the core 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.

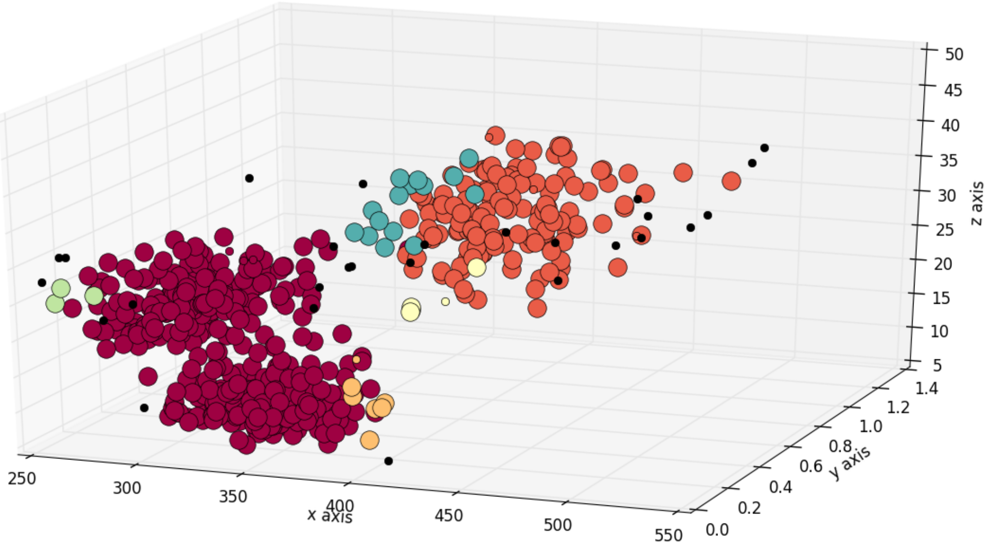
3.1

For Minpts = 3, we will first use k nearest neighbors method to identify the radius for the neighborhood detection. The radii are sorted in increasing order here to identify the knee


Neighborhood Radius plot (Elbow Method DBScan minPts = 3)

From this plot, we observe a knee at radius =4.5. Taking neighborhood radius = 4.5 and minPts as 3, we perform a DBScan, the result is as follows:

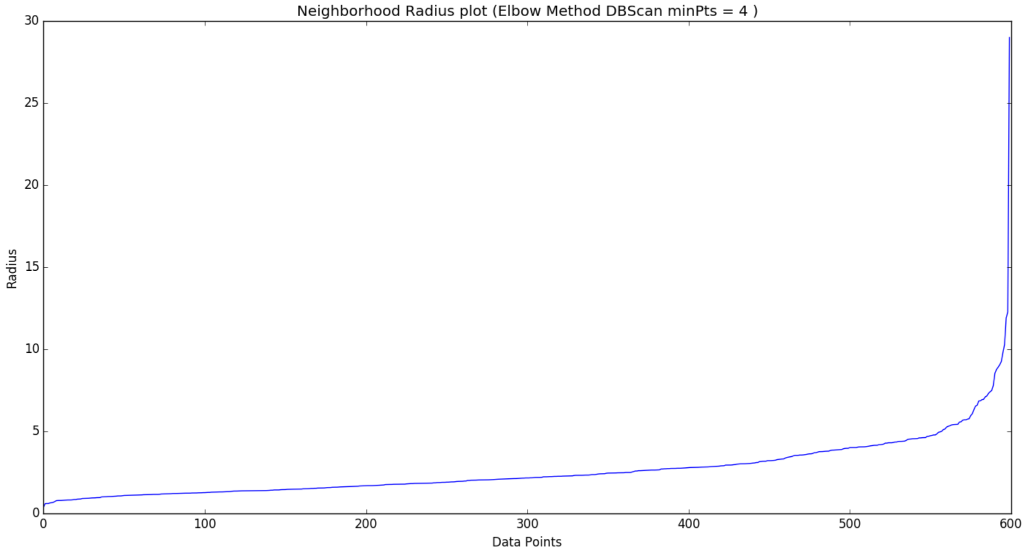
Estimated number of clusters: 6 for minPts = 3, Radius = 4.50



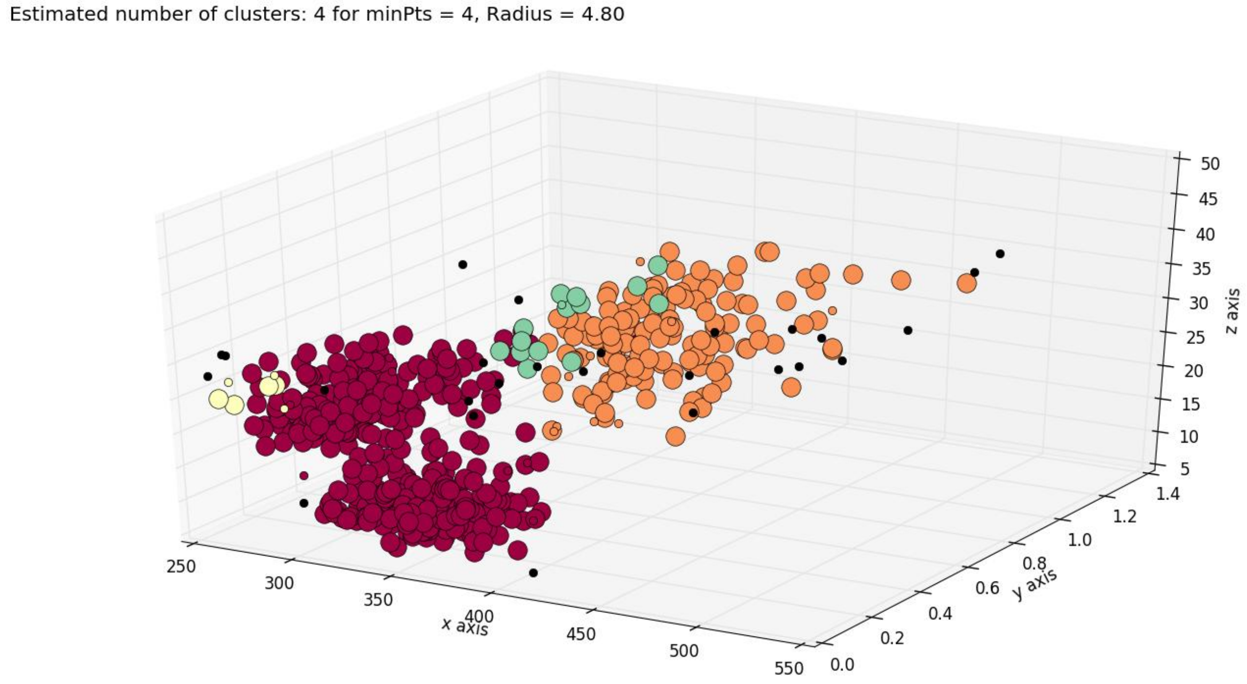
Estimated no. of clusters = 6 using the density based scanning. Since the minPts is less, we can expect cluster count to be more

3.2

Now, we check the radius for minPts =4

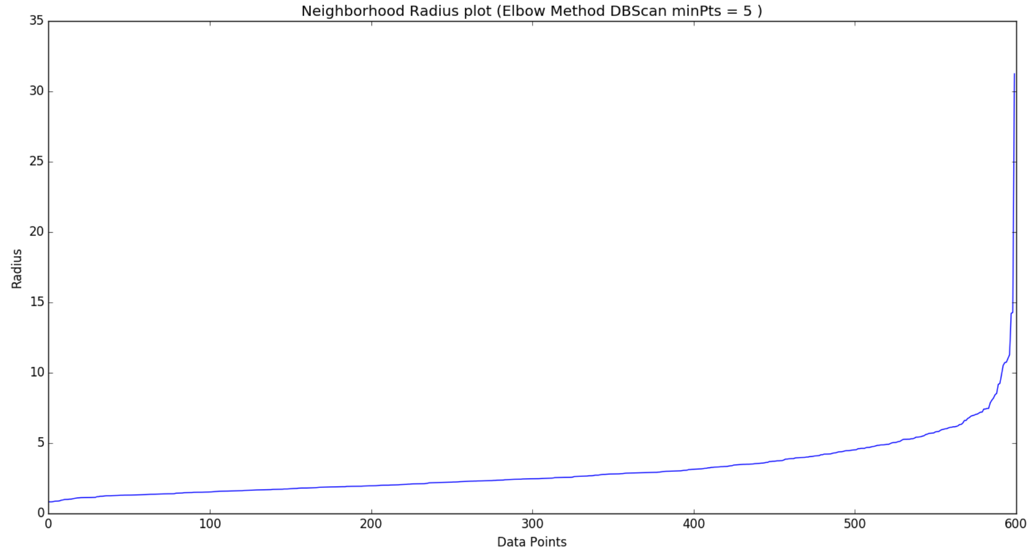


Here the knee falls at 4.8 which is almost same as for minPts = 3
Plotting the clusters for minPts = 4



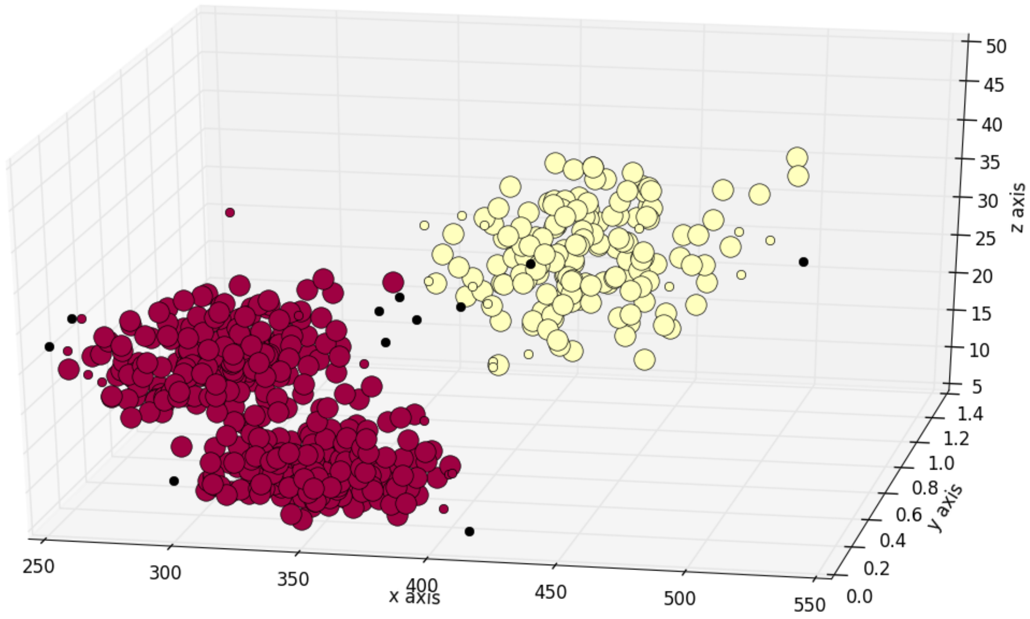
We observe a change in the clustering pattern by changing the minPts to 4. The no. of clusters is reduced now considering increase in the minPts.
But the cluster identification is not very uniform, the sizes of the clusters is varying a lot. So it is not a very good representation of the clusters.

Let's check for minPts = 5



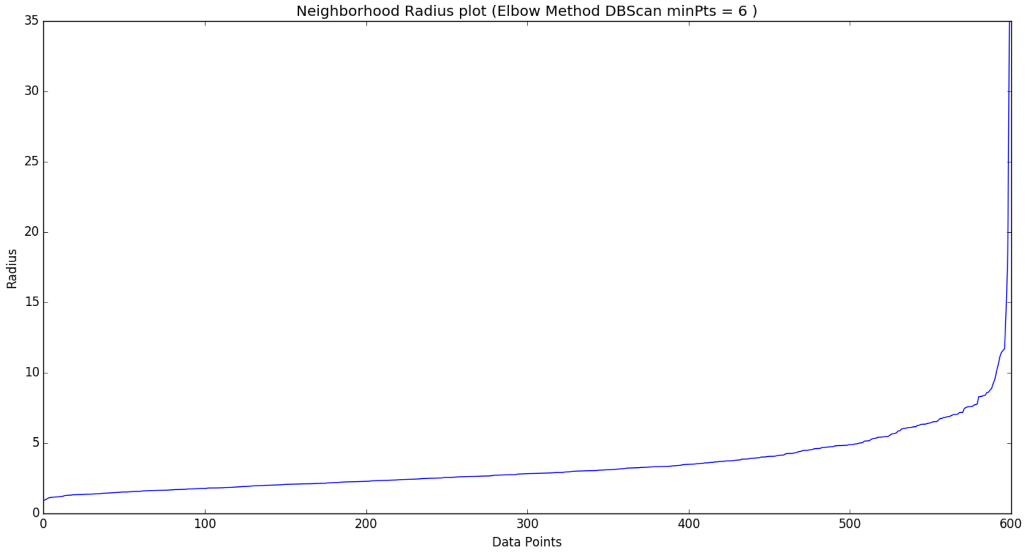
For minPts =5, the neighborhood radius falls at 6.2. Run DBScan on the data with minPts =5 and neighborhood radius = 6.2, we get

Estimated number of clusters: 2 for minPts = 5, Radius = 6.20

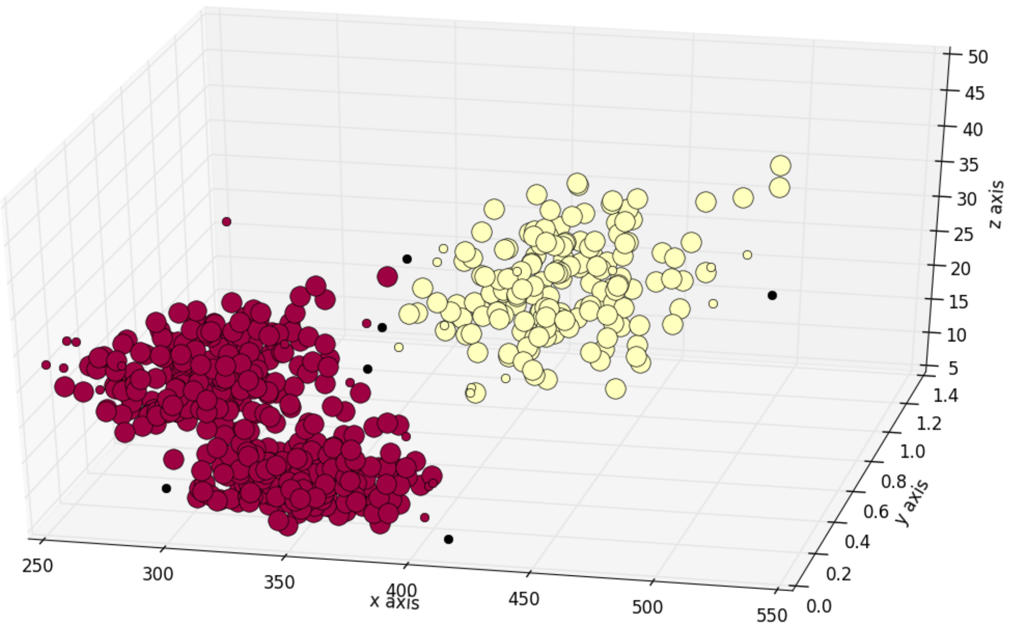


The estimated no. of clusters turnout to be 2 in this case.
Also the clustered core points, the noise values can be seen distinctly. It looks like the correct representation of the clusters

Let's give a try with minPts = 6



Here, the neighborhood radius is identified as 7.1.
Plotting clusters

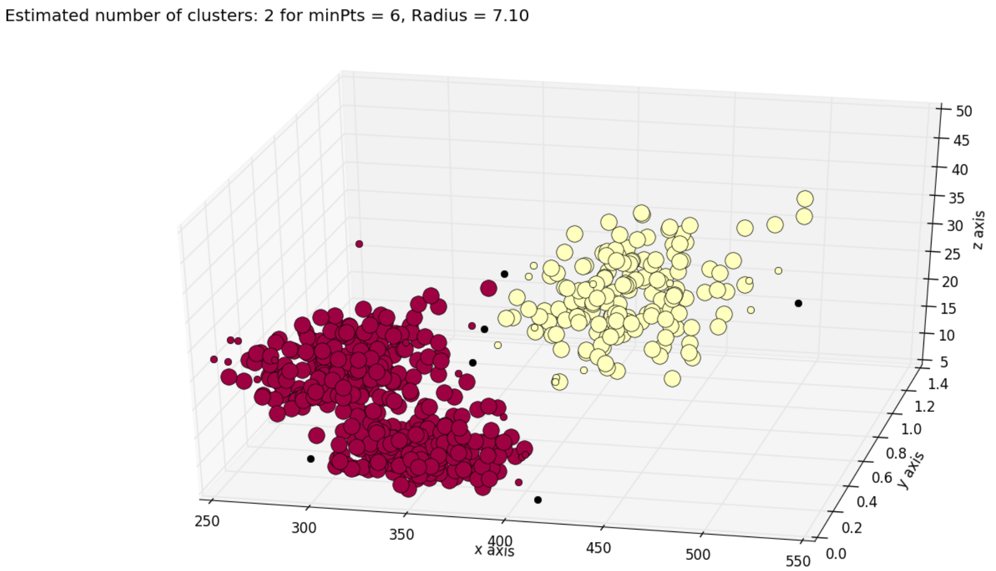

Estimated number of clusters: 2 for minPts = 6, Radius = 7.10

With minPts =6 too, we can identify two clusters from the given dataset using DBScan.

The results for all the dbScan runs (3-8) is written in a separate file as the question says

3.3

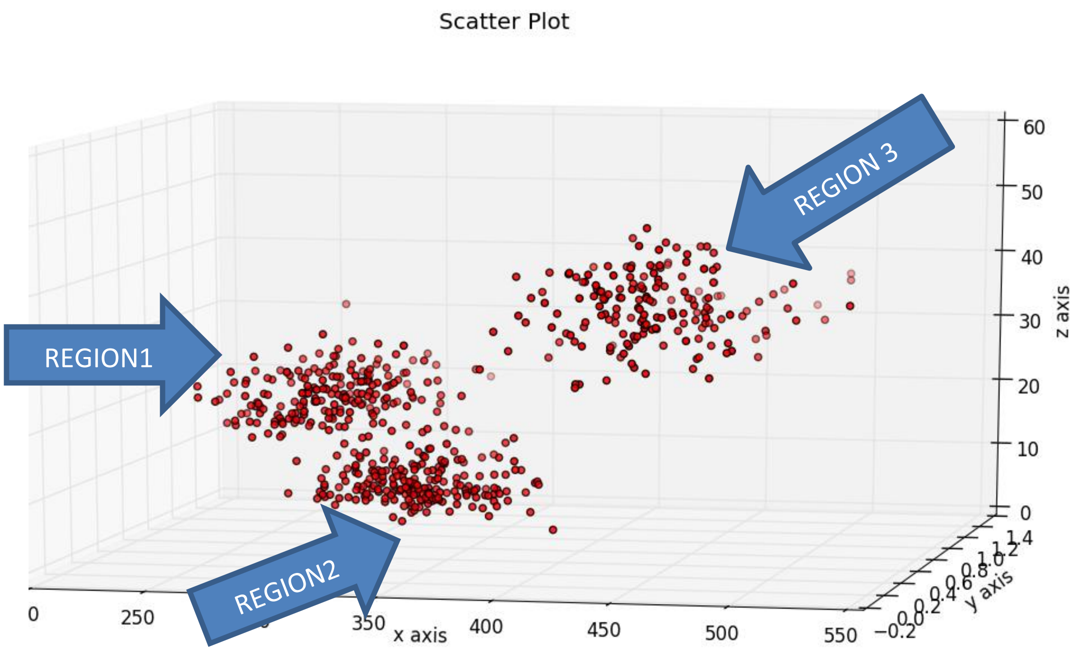
So with minPts = 5 or 6 and further for few other higher values as well, we have identified the best clustering for the given data set and the no. of clusters identified in density based scanning is 2.



This is expected too 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 can see the whole data being represented as a single cluster.

Task 4: Analysis

Let's again plot the data points



Comparing the data set and giving the best clustering from the above analysis

Hierarchical Clustering: It shows that there are two clusters which have merged due to the biggest difference in inter-cluster distance. Visually, there look to be three candidate clusters here (REGION 1-3), but since there are some data points acting as bridging points between REGION1 and REGION2,, the clusters are assigned same labels and hierarchical clustering gives two clusters

K-means: Using the elbow method, we are able to figure out value of k for which SSE is minimized and it is for k = 3. Based on that we are able to classify and assign cluster labels to the data points and hence we get three clusters. But the K-means clustering gives a result where some values seem to be assigned incorrectly.

DBSCAN: Density based scanning on the given data is also indicating that the density for the data points REGION1 and REGION2 is within threshold and hence it has spotted two clusters. With changing minPts and radius too, the algorithm at best distinguishes two clusters.

Conclusion:

K-means is able to spot the three regions as observed visually by minimizing the SSE for k=3, but there are some incorrect assignments. So, k-means has done the best job of clustering which supports visual inspection.

But we can also say that there are essentially two clusters in the given data points considering proximity (density) and distance as the parameters for cluster assignment. DBSCAN is not able to eliminate the points between region 1 and 2 considering them as noise. It implies that they satisfy the threshold constraints. Hence, there are two major clusters in the dataset.