ECE 592 005 – IOT Analytics

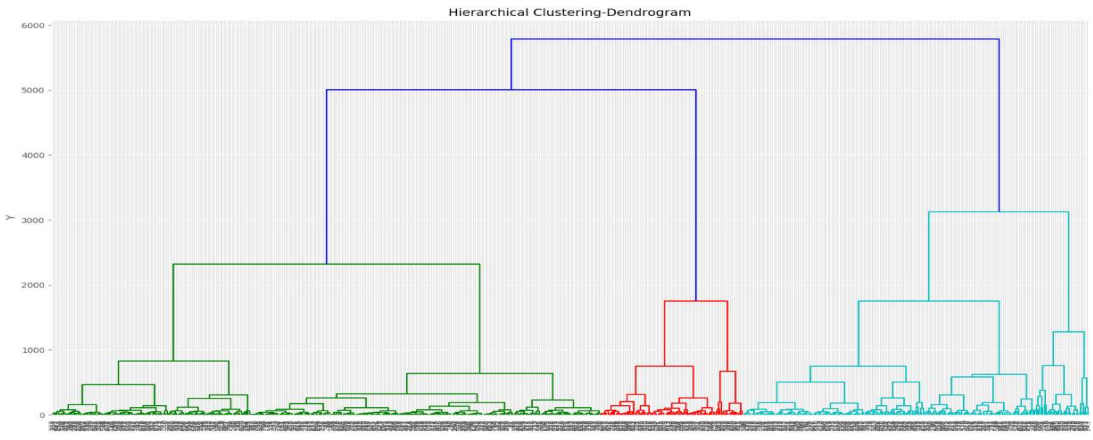
Project 4 – Clustering

Plain scatter Plot of the three-dimensional dataset without clustering:

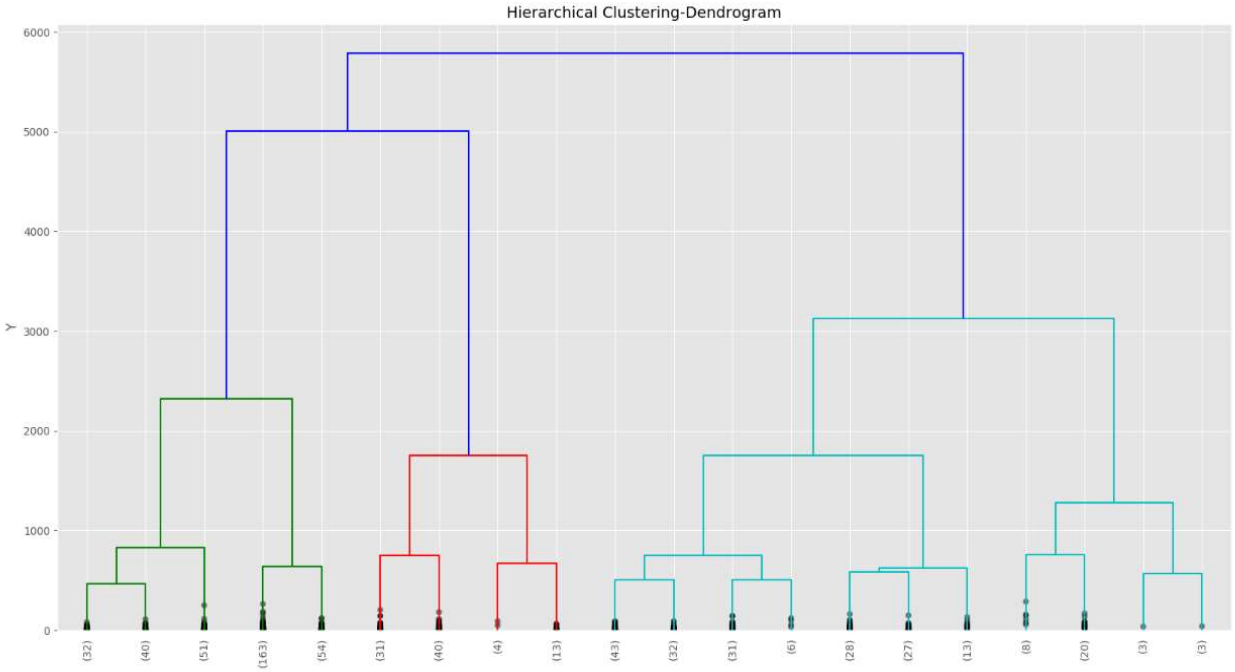
Task1: Hierarchical Clustering:

We apply the hierarchical algorithm to the three-dimensional dataset and plot a dendrogram.

The full dendrogram is as follows:



We truncate the dendrogram to observe the top 20 clustering linkages alone:



The values within the brackets denote the number of cluster linkages that are further below.

To determine the number of clusters from the dendrogram we use the following method

(i)Take the longest vertical distance that has no other horizontal line passing through it

(ii)A horizontal line is imagined cutting that distance

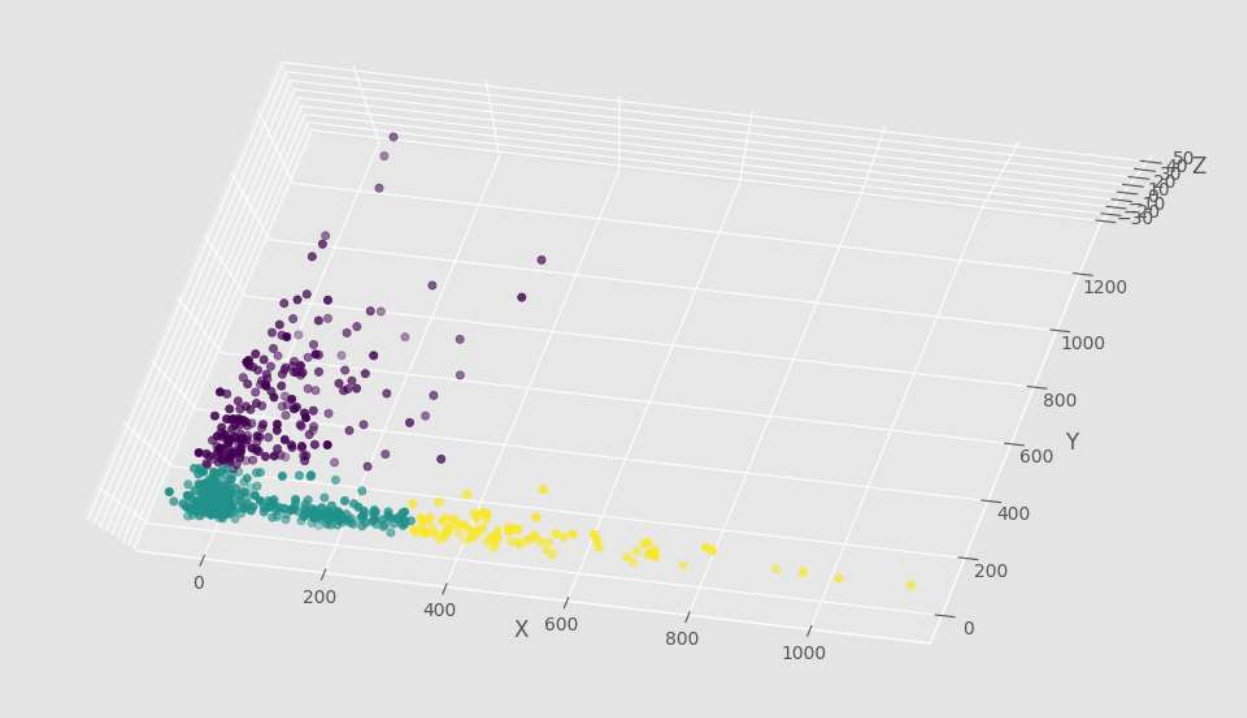
(iii)The number of vertical lines this imaginary horizontal line cuts is taken to be optimal number of clusters.

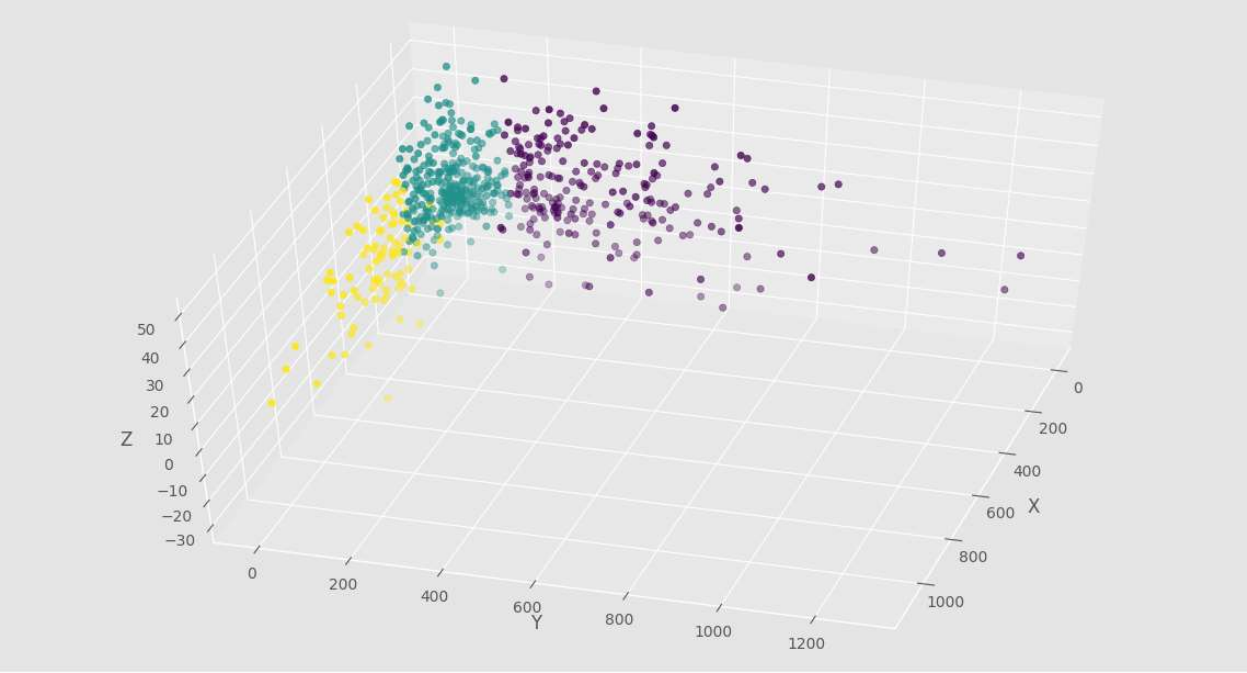
From the dendrogram, we take the number of clusters to be 3 as the imaginary horizontal line the longest vertical distance cuts 3 vertical lines.

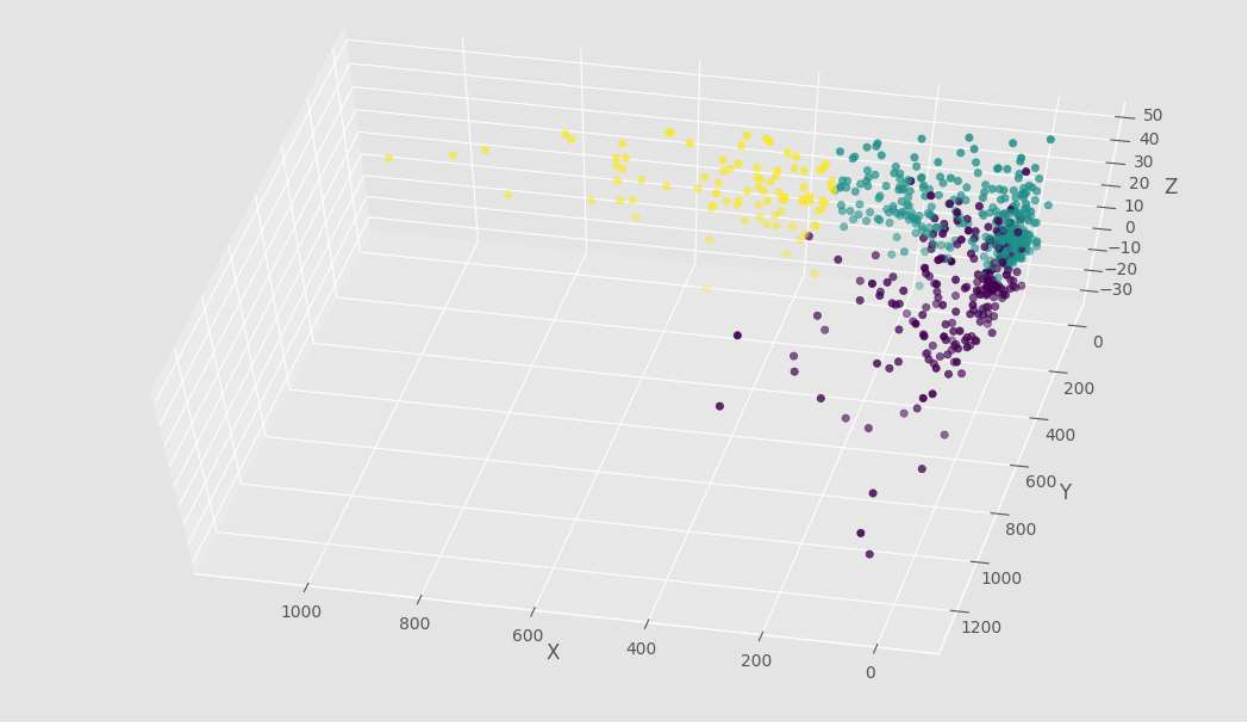
Now we perform Agglomerative Clustering using no.of clusters as 3 and the distance metric to be Euclidean.

```
ragavi@ubuntu:~/workspace/IOT_Project4/task1$ python -W ignore hierarchial.py
No of points in each Cluster: 0 to n are the clusters
{0: 214, 1: 340, 2: 88}
```

A 3d scatter plot for the clusters formed is as follows, (Figures give various projections)


Z
50
40
30
1200
1000
800
Y
600
400
200
0
0
200
400
X
600
800
1000


50
40
30
20
Z
10
0
−10
−20
−30
0
200
400
600
Y
800
1000
1200
0
200
400
600
X
800
1000

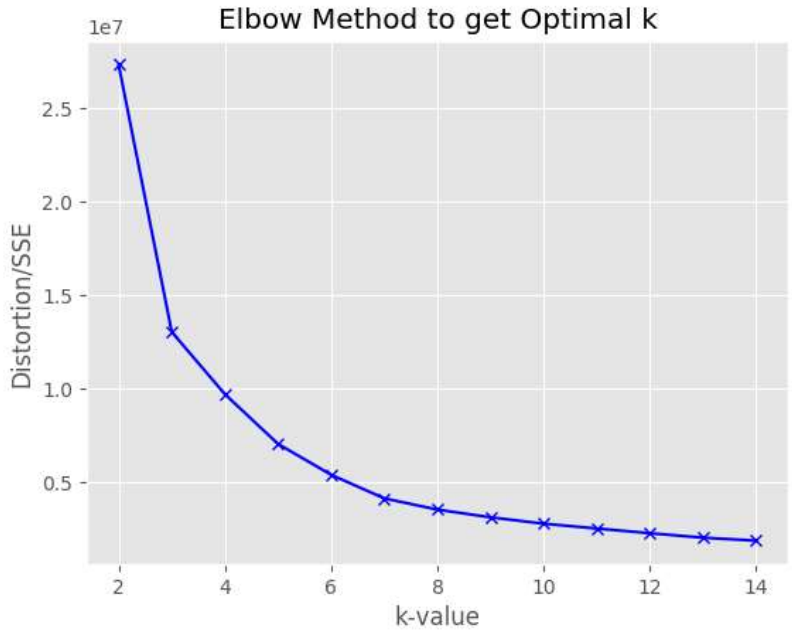


We identify three clusters in 3 different clusters, and they are closely packed to each other. Hierarchical clustering does not seem to identify points that are noisy even though the scatter plot clearly shows some data points that are very far away.

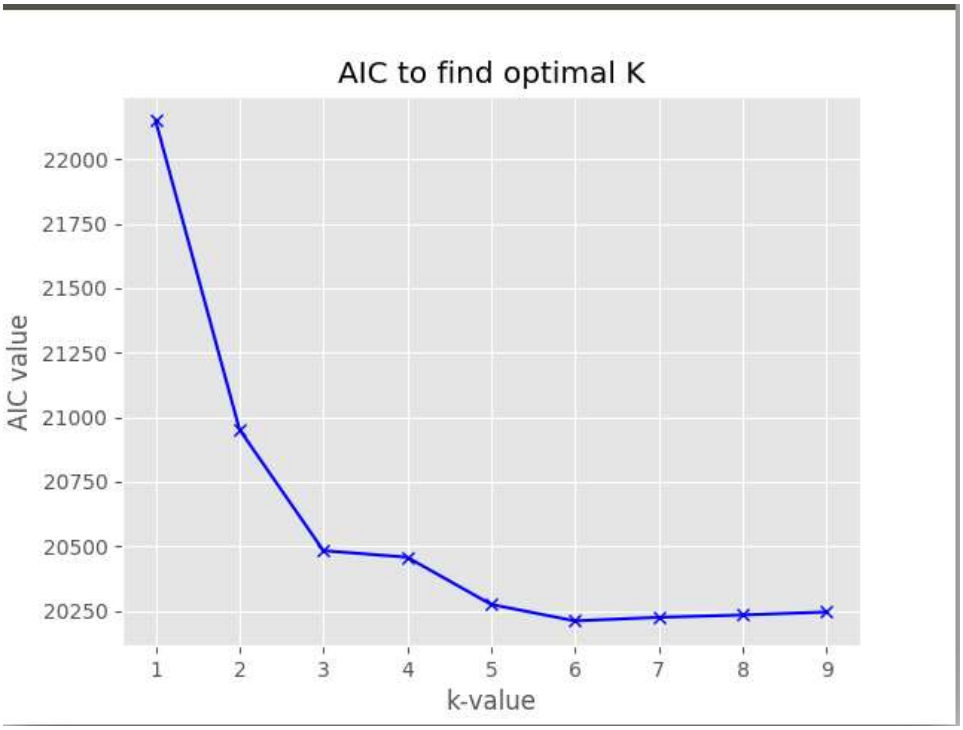
Task2: K-Means Clustering:

K-means clustering focuses on clustering data points in the given dataset based on a given number of groups. The algorithm iteratively assigns every data point in the data set in to one of the given groups (K) that matches its similarity. At the end of the algorithm, K-means returns the labels (group) to which every data point belongs to and the centroids of the K clusters. These centroids can be used to predict the labels of unknown data in the future.

To find the best K value, we find the SSE/distortions values for a range of K values (2,15) and plot a graph of the distortion vs K.

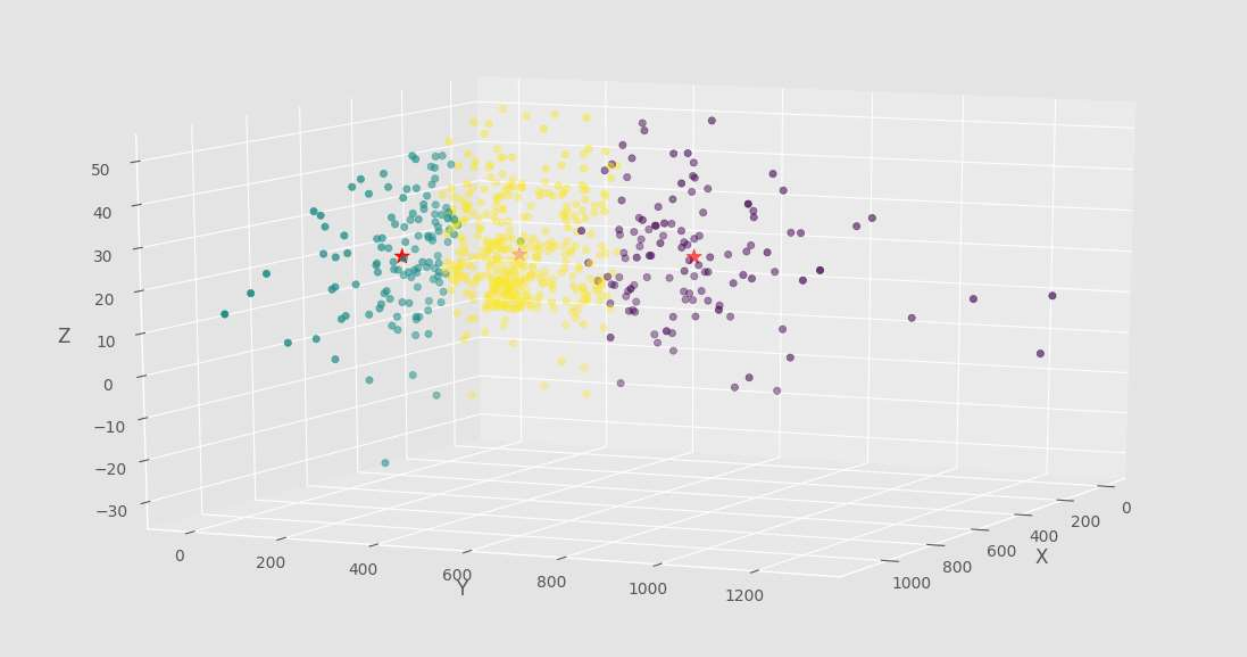


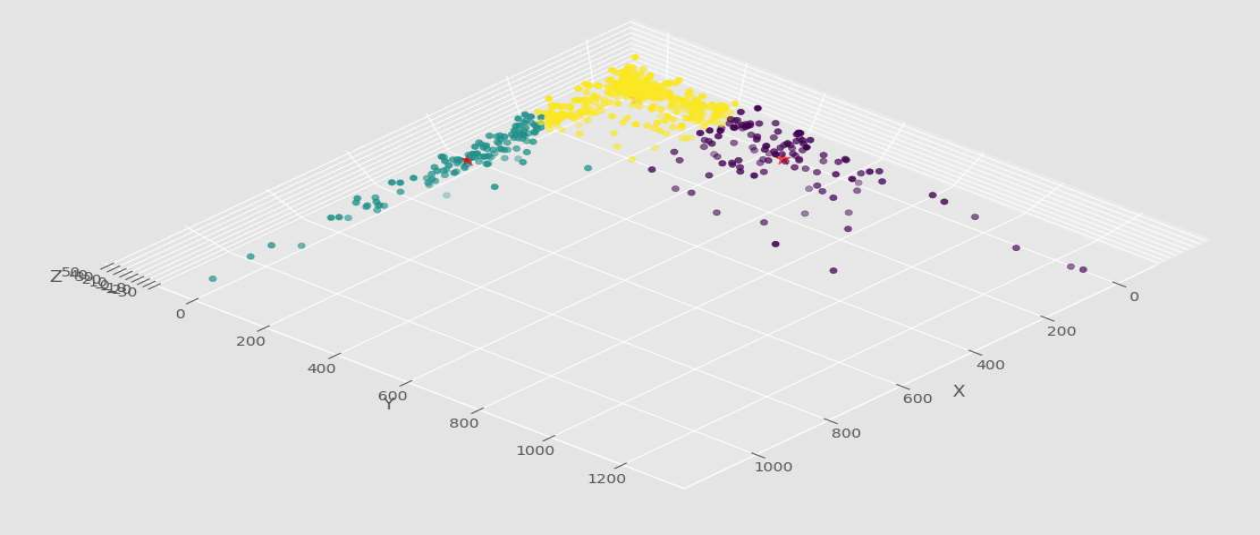
The plot does not seem to give a clear indication of the valley and hence we calculate the AIC (Akaike's criterion) values for a range of K, plot it and find the valley.



From the above plot, we take the number of clusters to be 3 and the same as the best K value. We use the K-means algorithm on the best K value, obtain the 3D scatter plot and color each cluster in a selective color. We also mark the centroids for each cluster.

```
ragavi@ubuntu:~/workspace/IOT_Project4/task2$ python -W ignore kmeans.py
('centroids:', array([[ 80.4225206 , 499.0962931 ,  16.34585707],
       [457.18391667,  44.95563417,  18.99149717],
       [ 64.91852158,  94.05093793,  14.57683089]]))
No of points in each Cluster: 0 to n are the clusters, -1 denotes noise
{0: 116, 1: 120, 2: 406}
```





We identify three clusters in 3 different clusters and their centroids, and they are closely packed to each other. K-means clustering does identify points that are noisy even though the scatter plot clearly shows some data points that are very far away. It has to allocate the data point to one or the other group according to K to whichever group seems to be similar to the data point or closest to the group's centroid location.

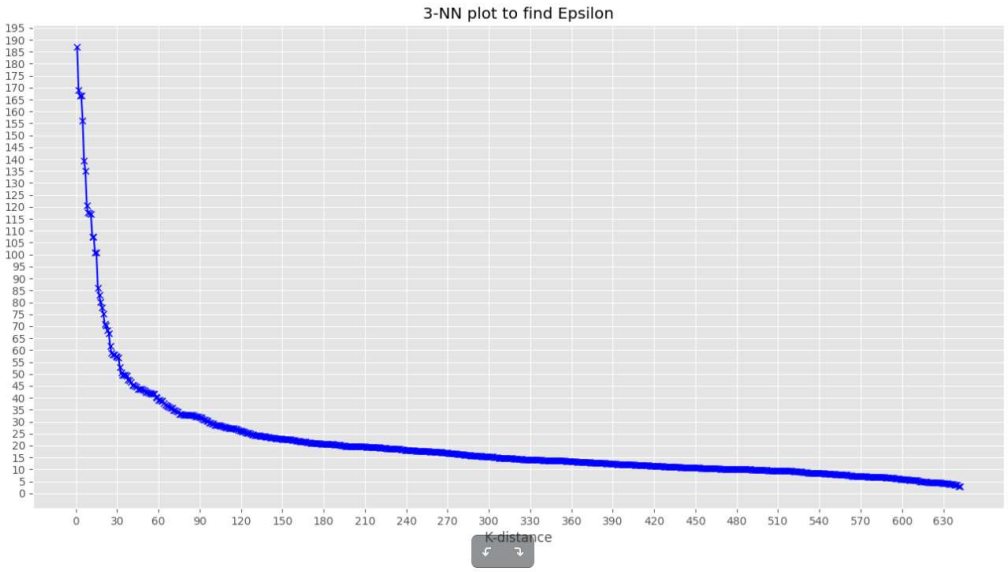
Task3: DBSCAN Clustering:

DBSCAN is density based and requires two parameters, the radius ε of a hypersperical neighborhood and the minpts, the minimum number of points that have to be within the radius mentioned.

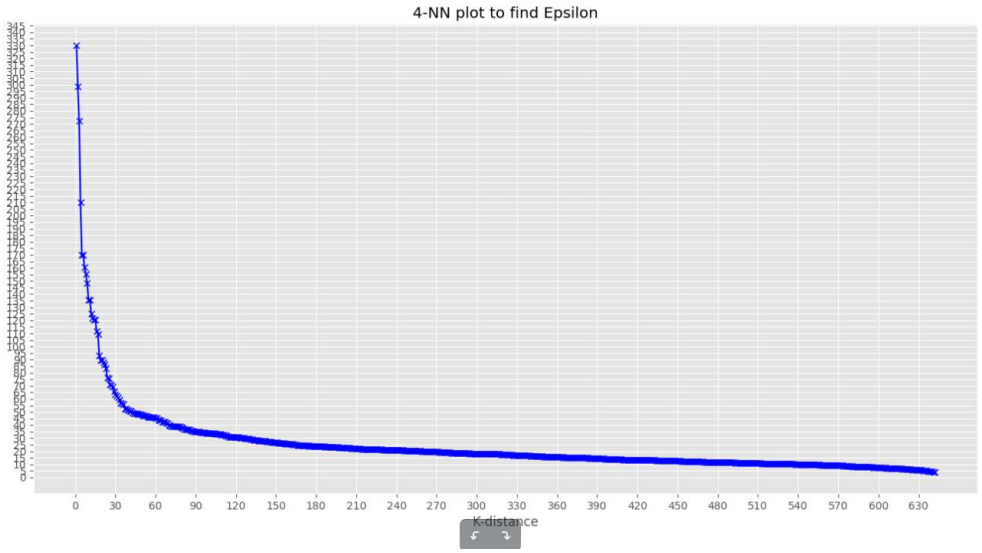
We start by taking the minpts value to be 3. We use the nearest neighbors (K-NN) ie)3-NN distance measures and find the average of distances of every data point in the data set to three of its nearest neighbors. In general, for K-NN, we find the average of distances for K-nearest neighbors for each data point in the three-dimensional dataset.

The k-distances are sorted in a descending order and plotted along with the index of the data points. Ie)1 to 642. The knee in the plot corresponds to the best ε value.

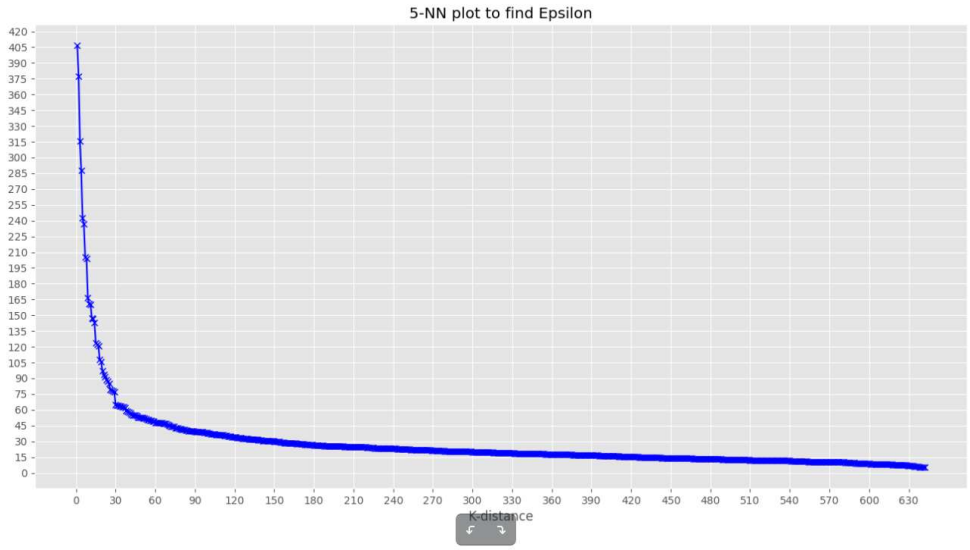
For minpts=3, the 3-NN plot is as follows,



For minpts=4, the 4-NN plot is as follows,



For minpts=5, the 5-NN plot is as follows,



Best Clustering:

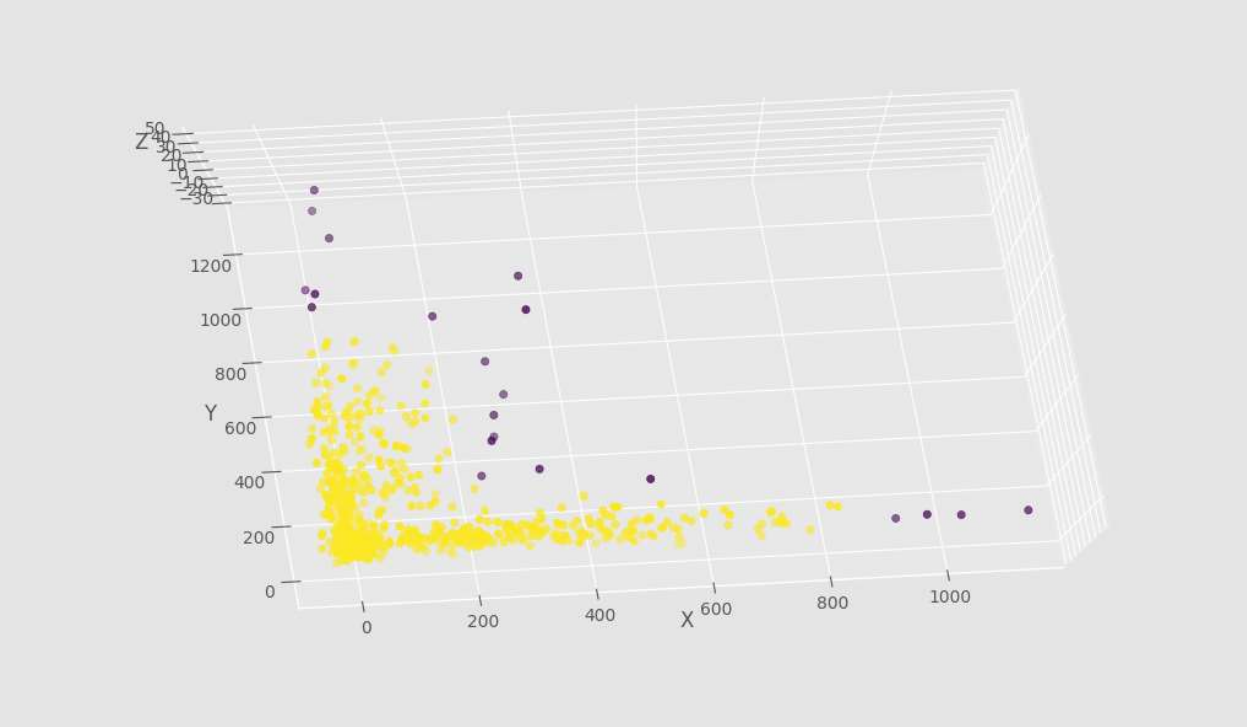
We tried DBSCAN algorithm with the corresponding ε and minpts and arrived at different cluster and noise points values.

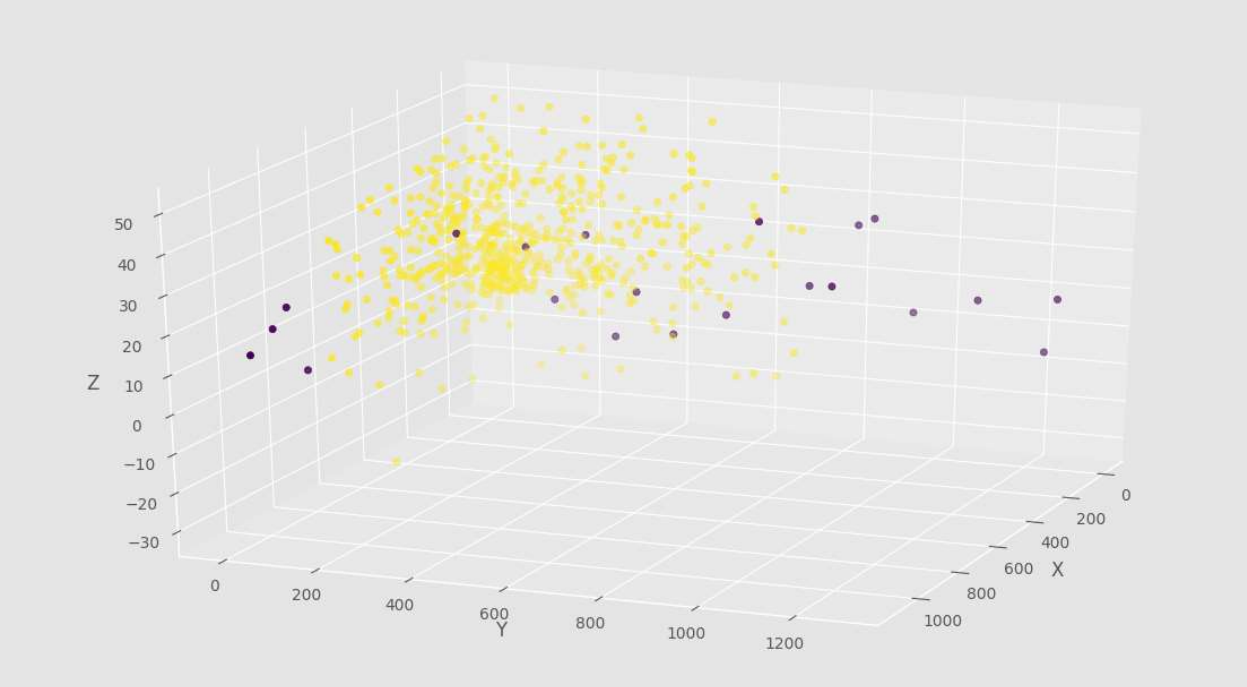
Minpts=3 identifies 11 clusters. But most of the points lie in one major cluster while the other clusters are sparsely populated. Minpts=4 also produces similar results. Both these minpts have a higher ratio of outlier/noise points that do have the same density.

We identify Minpts =5 and ε = 65 to be the parameters that give the best clustering. It identifies one cluster and a minimal number of noise points.

```
ragavi@ubuntu:~/workspace/IOT_Project4/task3$ python -W ignore dbscan.py
('Epsilon:', 65)
('Minpts:', 5)
('No of clusters:', 1)
('No of Noise Points:', 21)
No of points in each Cluster: 0 to n are the clusters, -1 denotes noise
{0: 621, -1: 21}
```

The 3D scatter plot of the cluster and the noise points in various projections is as follows:





Both the above clustering techniques did not handle outliers. The DBSCAN clustering method on the other hand handles noise points / outliers.

Task4: Comparison and Identification of the best clustering:

The given dataset on plotting initially using a 3D scatter plot, majority of the data points appear to be very closely packed while a few points seem to be scattered far away in the three-dimensional plane.

For such a dataset, on using the Hierarchical Clustering, the dendrogram identified three clusters. The clusters on observing using a 3D scatter plot again clustered even points that looked like outliers into one of the clusters due to its linkage property. This type of clustering cannot be used to predict which group a new datapoint falls into if a new point needs to be clustered in the future. Dendrograms need to be drawn to identify its position.

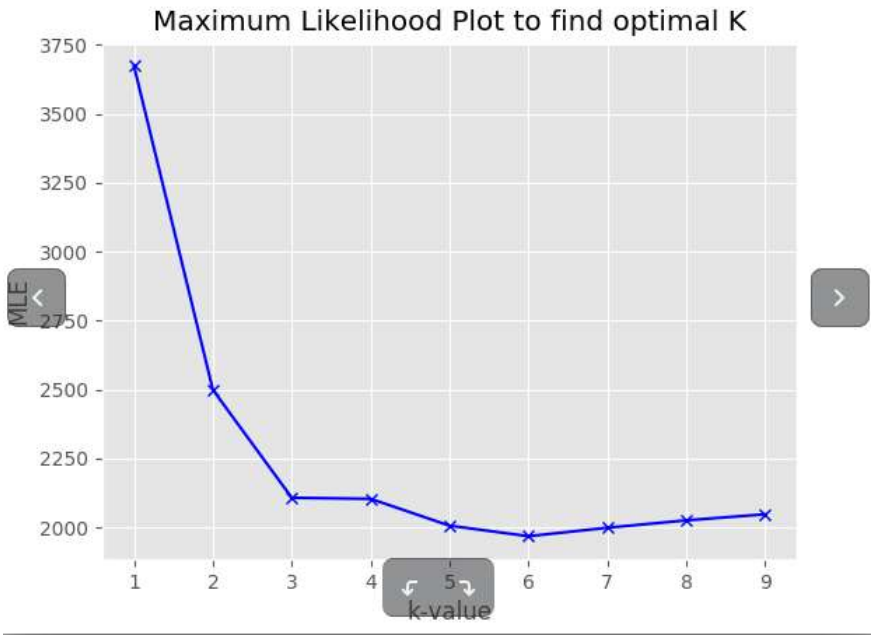
K-Means on the other hand is a centroid based clustering that is iterative and based on the notion of similarity. This similarity is found using the closeness of a data point considered to the centroid of a particular cluster. Our dataset clearly contains points that are far away from the regular lot, but the K-means algorithm when applied fails to identify the data points that deviate highly from the normal set of points. The algorithm is in a compulsion to cluster even the outliers under one of its labels. Further, identifying the best K value is a cumbersome job using the elbow method as we got a smoother curve and AIC had to be used.

The last clustering used was the DBSCAN clustering. We found the best ε for each of the minpts value considered. This clustering method clearly identified the outliers present in the dataset. But majority of the datapoints as they are tightly packed and due to higher density, for minpts >= 5, the algorithm tends to output only one cluster. The noise points seem to be identified with a better accuracy in this model.

Hence, out of all three clustering models, we identify the DBSCAN to be more accurate and cater to the actual nature of the dataset considered in identifying outliers. However, we cannot clearly declare which clustering model is the best as the datasets do not fall under a business context. A business context under which the data set falls, the origin of the data set and the usage of the clustering results when known will be more useful to identify the best clustering method of all.

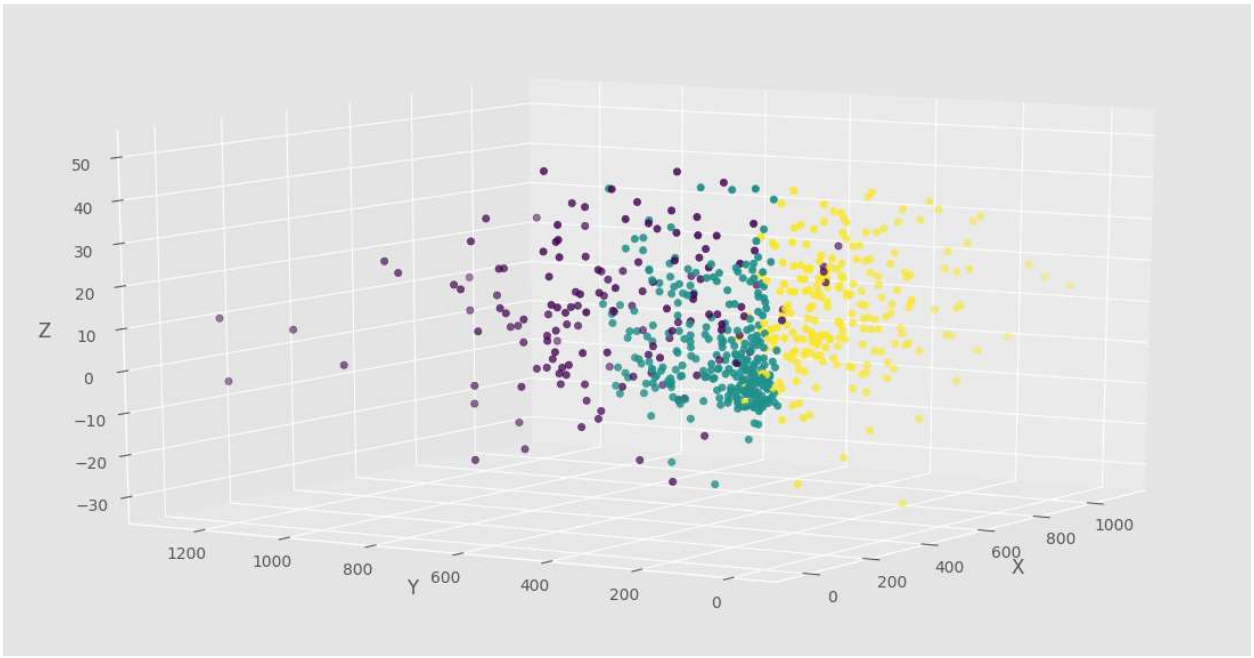
Task: Gaussian Decomposition Method:

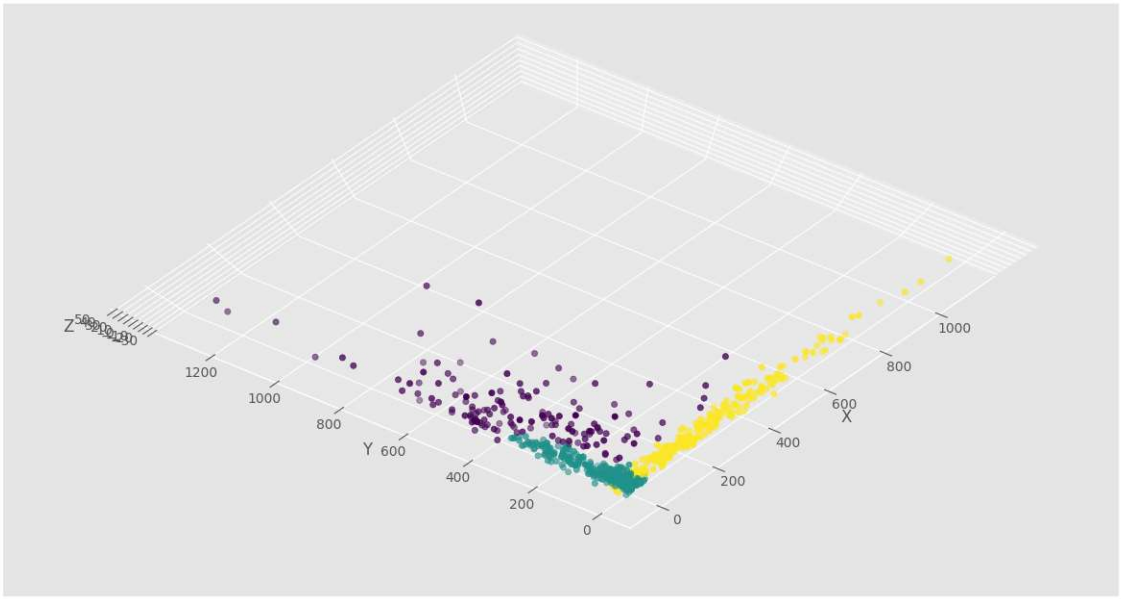
In the Gaussian Decomposition method, we assume that the data points are from a set of normal distributions which is equal to be number of clusters. To use the Gaussian Mixture algorithm, we need the best K or the best probability that these number of normal distributions have given the data points. Hence, we plot the maximum likelihood against the K values. The value that maximizes this likelihood is the best K value or the number of clusters.

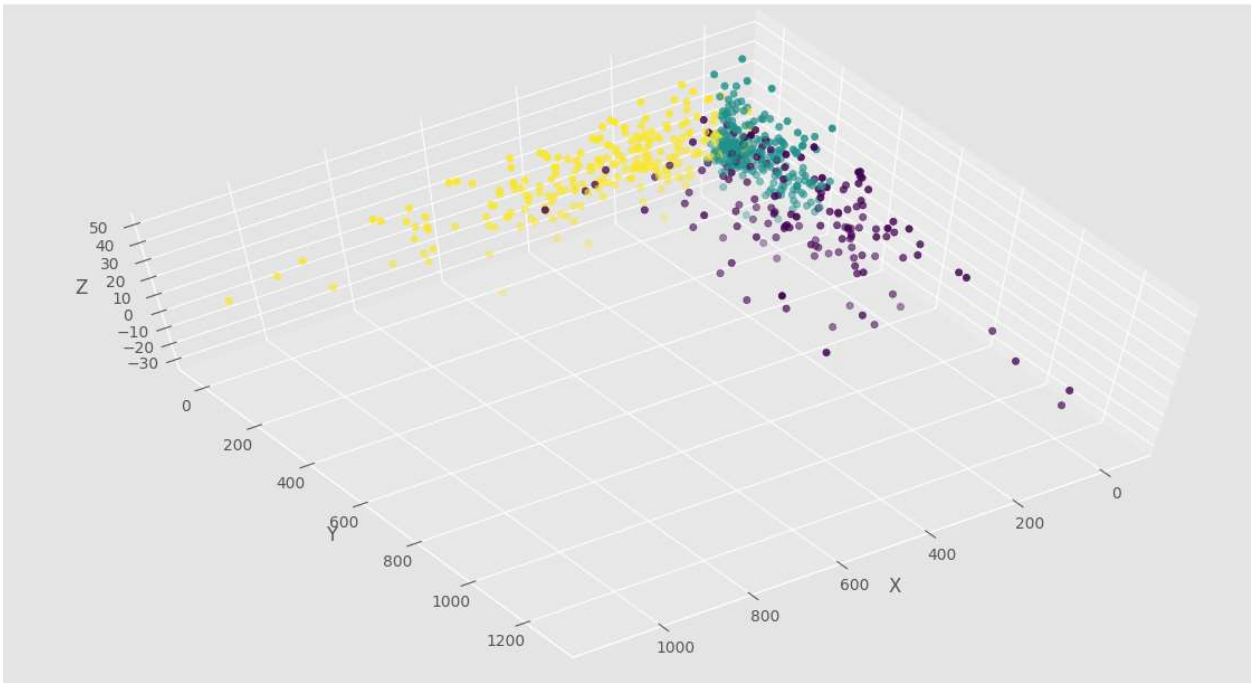


We plot the BIC value for a range of K. The BIC value is the lowest for the Maximum likelihood value ie) the probability value that a data point comes from a particular normal distribution.

On applying K = 3 and plotting the 3D scatter plot, we observe the following,







Comparison with other Clustering methods:

The Gaussian method is majorly used when the clusters are not simple and can be outlined by circles but have more complex shapes that define them. Our dataset also has complex clusters with majority of the data which is tightly packed. The Gaussian method is clearly better efficient than the Hierarchial and K-means method as it considers the co-variance and likelihood scores into consideration while clustering

data points and not using a simple linkage or based on the similarity to a nearest centroid. On comparing to the dbscan algorithm, even though DBSCAN identifies the outliers in the data set, it clustered all data points into one big cluster on increasing the minpts. The Gaussian Mixture model seems to give a better clustering than the dbscan in this case.