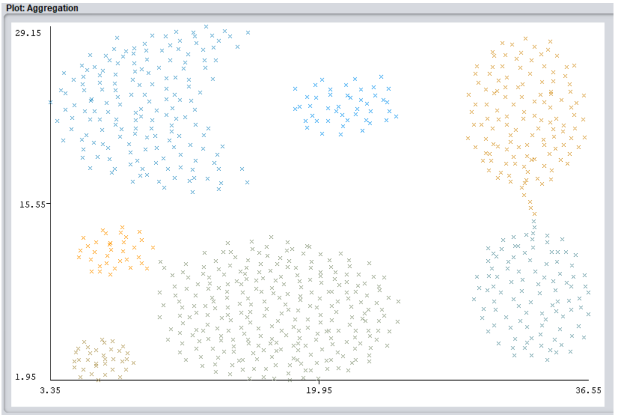
CS18M524 Report

1. All the datasets have been converted to ARFF Format and stored in the **Datasets folder.**

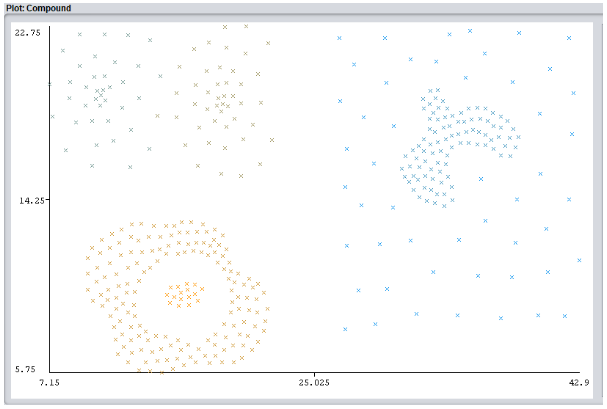
2. The visualizations of all the 8 datasets have been stored in the **Plots folder.**
   In every case the first 2 attributes have been selected , the 3rd column which corresponds to the label has been ignored.
   The actual graphs corresponding to the clusters have been stored in **Pictures Of Plots folder.**

   They are as follows : -

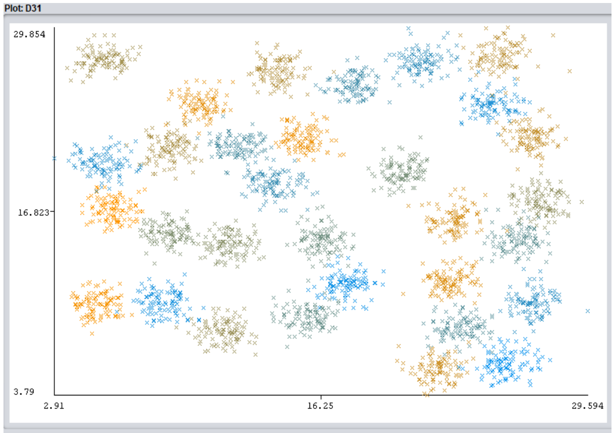
Aggregation



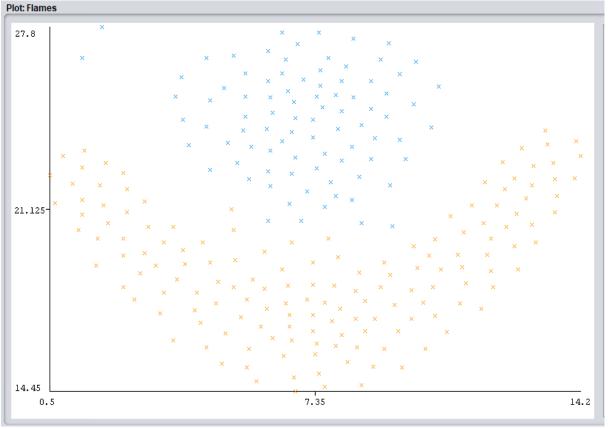
Compound



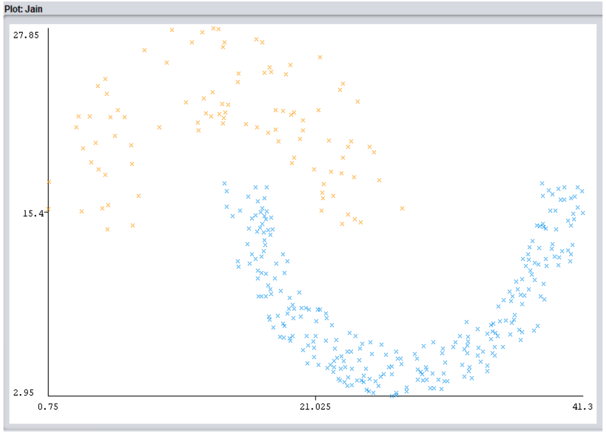
D31



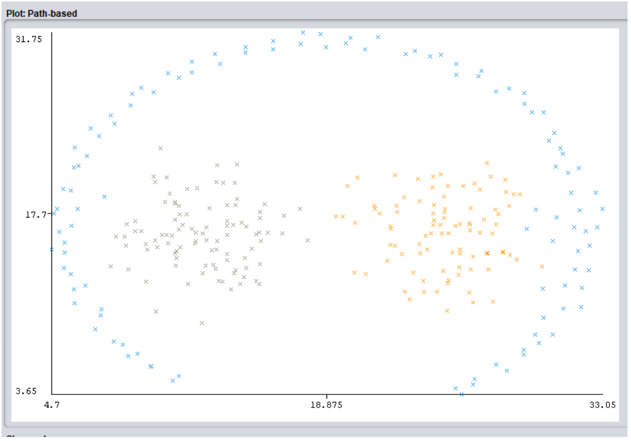
Flames



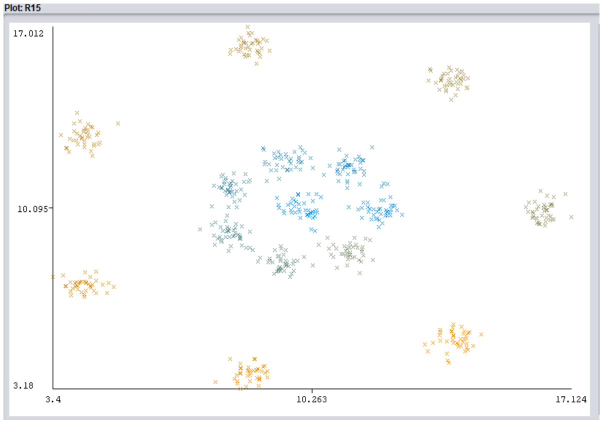
Jain



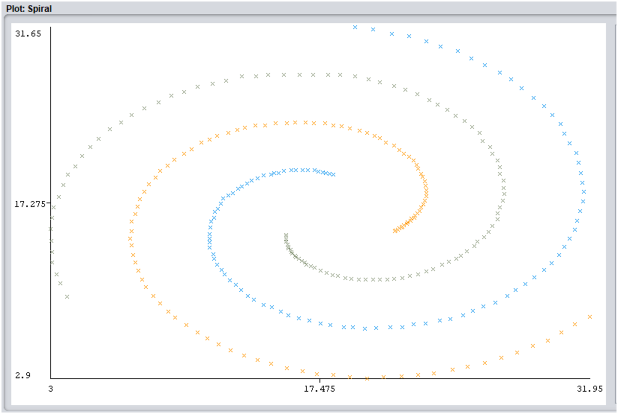
Path-based



R15



Spiral



a. **For Aggregation Dataset:-**

K- means - Should work well as the clusters are approximately spherical. **This works the best**
DBSCAN - The cluster densities are not same , this would make DBSCAN suffer
Hierarchial Clustering (Single Link)- Will not work very well as nearby elements of the two clusters might lead to their merging intone.
Hierarchial Clustering (Complete Link)- Will not work very well as it tends to produce clusters of equal diameters, which is not the case here.

b. **For Compound Dataset:-**

K-means – Structures are not really spherical and outliers are there so K -means won't work well.
DBSCAN – This will work the best as it is equipped to work with outliers and makes no assumption as to shape. But cluster densities are not equal .

Hierarchial Clustering (Single Link)- Will not work well as it will tend to fuse clusters towards the right into a single cluster.
Hierarchial Clustering (Complete Link)- Will work alrighty, but clusters have varying diameters which might make the algorithm suffer.

c. **For D31 Dataset:-**

K-means – This works the best as clusters are spherical and not many outliers.
DBSCAN – This works very well too as the clusters have nearly the same densities.
Hierarchial Clustering (Single Link)- Will not work well as it will tend to fuse clusters into a single cluster.
Hierarchial Clustering (Complete Link)- Will work well as the clusters have similar diameters

d. **For Flames Dataset:-**

K – means - Will not work well as the clusters are not spherical
DBSCAN – Should work well as clusters have approximately equal density.
Hierarchial Clustering (Single Link)- Will not work well as it will tend to fuse clusters into a single cluster.
Hierarchial Clustering (Complete Link)- Will not work well due to varying diameters.

e. **For Jain Dataset:-**

K – means - Will not work well as the clusters are not spherical
DBSCAN – Should work well as clusters have approximately equal density.
Hierarchial Clustering (Single Link)- Will work moderately well as long thin clusters are it's specialty.
Hierarchial Clustering (Complete Link)- Will not work well due to varying diameters.

f. **For Path Based Dataset:-**

K-Means - **Will not work well** as the clusters are not spherical.
DBSCAN – Will not moderately well but clusters formed might be different than what is expected.
Hierarchial Clustering (Single Link)-Will not work well. It might fuse the inner clusters with the outer ring .
Hierarchial Clustering (Complete Link)- Will work alright but might suffer due to varying diameters.

g. **For R15 Dataset:-**

K-Means – Will work well as clusters are spherical, varying inter cluster distances might cause problems.
DBSCAN – Should work well as clusters have approximately equal density.
Hierarchial Clustering (Single Link)- Will not work well as it will tend to fuse clusters that are close together.
Hierarchial Clustering (Complete Link)- Should work well as clusters have similar diameters

h. **For Spiral Dataset**
K-Means - **Will not work well** as the clusters are not spherical.
DBSCAN – Should work well as clusters have approximately equal density.
Hierarchial Clustering (Single Link)- Will work moderately well as long thin clusters are it's specialty.
Hierarchial Clustering (Complete Link)- Will not work well, as the lines are thin and long with no diameter.

3. Result of running K means clustering on R15 with k =8 on Weka

=== Run information ===

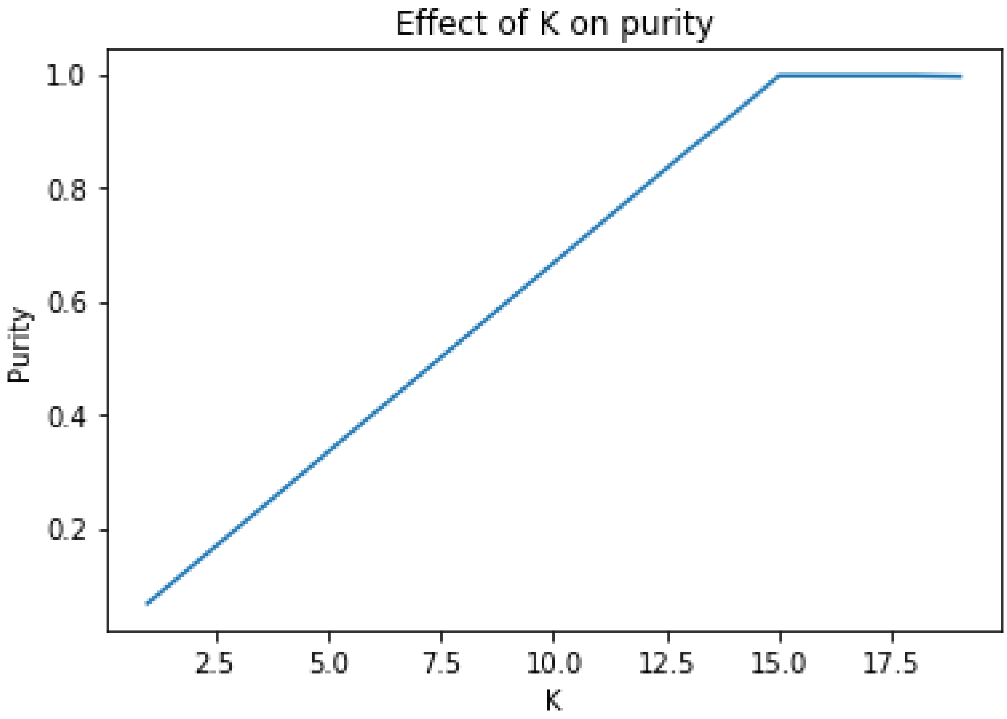
```
Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -
1.0 -N 8 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation:     Aggregation

Instances:    788

Attributes:   3

              Column1

              Column2

Ignored:

              Column3

Test mode:    evaluate on training data

=== Clustering model (full training set) ===

kMeans

======

Number of iterations: 10

Within cluster sum of squared errors: 12.191499856090307

Initial starting points (random):

Cluster 0: 31.75,20.75
Cluster 1: 16.4,8.7
Cluster 2: 19.1,2.65
Cluster 3: 32.55,22.05
Cluster 4: 14.1,10.05
Cluster 5: 16.6,7.95
Cluster 6: 33.25,22.25
Cluster 7: 13.35,28.45
Missing values globally replaced with mean/mode
Final cluster centroids:
                    Cluster#
Attribute    Full Data        0        1        2        3        4        5        6        7
            (788.0)    (52.0)   (115.0)    (55.0)    (47.0)   (113.0)   (113.0)   (125.0)   (168.0)
==========================================================================================
======
Column1      19.5668    32.8462    18.9096    33.3973    21.2872     8.9221    18.0717    32.6952     9.2946
Column2      14.1718    11.6394     9.5157     6.5064    22.9989     7.7044     4.5774    22.2808    22.9527


Time taken to build model (full training data) : 0.04 seconds
=== Model and evaluation on training set ===
Clustered Instances
0       52 (  7%)
1      115 ( 15%)
2       55 (  7%)
3       47 (  6%)
4      113 ( 14%)
5      113 ( 14%)
6      125 ( 16%)
7      168 ( 21%)
```

The python code to run K means on R15 dataset with k =8 is in the file CS18M524_PA3.ipynb

```
The purity with K=8 is ::::  0.5333333333333333
```

Result of varying K from 1 to 20 , on purity :

```
K        Purity
1 0.06666666666666667
2 0.13333333333333333
3 0.2
4 0.26666666666666666
5 0.3333333333333333
6 0.4
7 0.4666666666666667
8 0.5333333333333333
9 0.6
10 0.6666666666666666
11 0.7333333333333333
12 0.8
13 0.8666666666666667
14 0.93
15 0.9966666666666667
16 0.9966666666666667
17 0.9966666666666667
18 0.9966666666666667
19 0.995
```



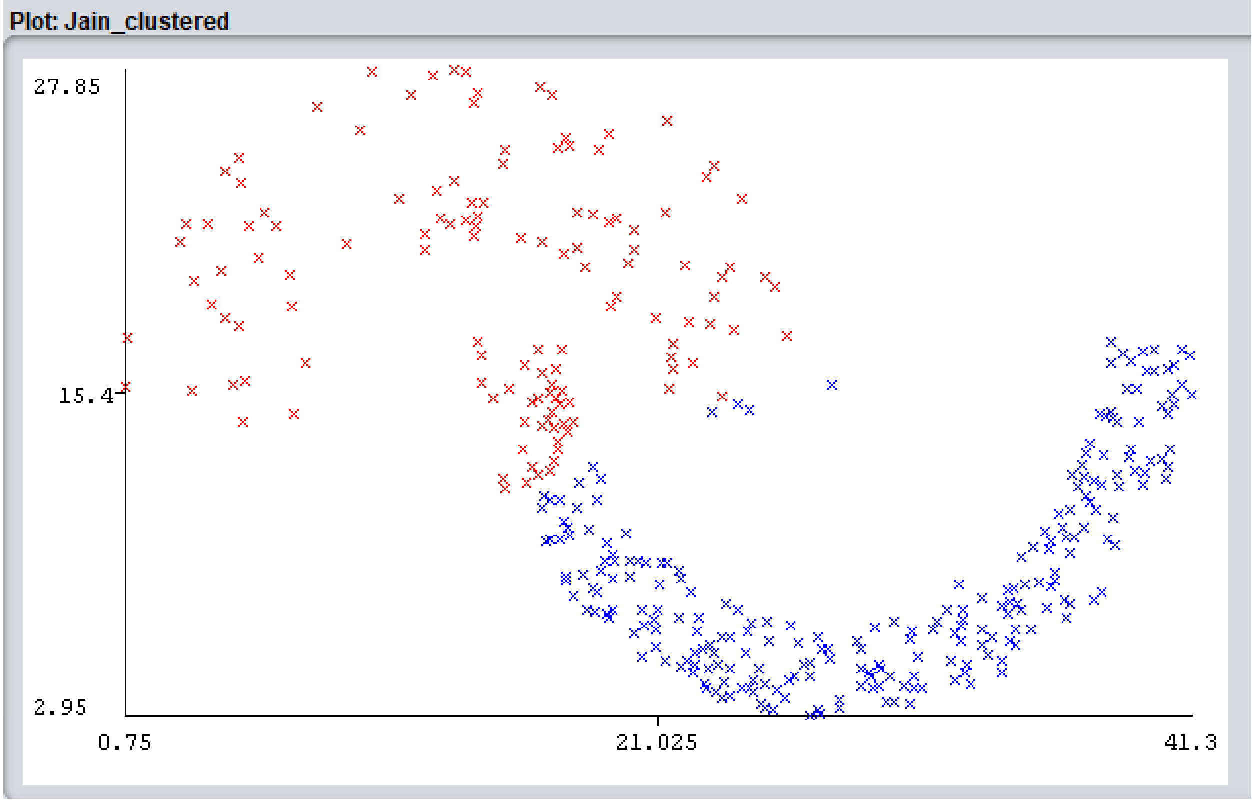
4. Result of running DBSCAN on Jain dataset

Weka Results

```
=== Run information ===

Scheme:                weka.clusterers.MakeDensityBasedClusterer -M 1.0E-6 -W
weka.clusterers.MakeDensityBasedClusterer -- -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -init 0 -
max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A
"weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Relation:     Jain

Instances:    373

Attributes:   3

              Column1

              Column2

Ignored:

              Column3

Test mode:    evaluate on training data

=== Clustering model (full training set) ===

MakeDensityBasedClusterer:

Wrapped clusterer: MakeDensityBasedClusterer:

Wrapped clusterer:

kMeans

======

Number of iterations: 7

Within cluster sum of squared errors: 20.57393157278048

Initial starting points (random):

Cluster 0: 18.4,10.05

Cluster 1: 17.45,20.75

Missing values globally replaced with mean/mode

Final cluster centroids:

                         Cluster#

Attribute    Full Data         0         1

               (373.0)    (234.0)    (139.0)
```

```
============================================

Column1       24.3307    29.7491     15.209

Column2        12.146     8.1271    18.9115

Fitted estimators (with ML estimates of variance):

Cluster: 0 Prior probability: 0.6267

Attribute: Column1

Normal Distribution. Mean = 29.7491 StdDev = 7.3698

Attribute: Column2

Normal Distribution. Mean = 8.1271 StdDev = 3.7476

Cluster: 1 Prior probability: 0.3733

Attribute: Column1

Normal Distribution. Mean = 15.209 StdDev = 5.9807

Attribute: Column2

Normal Distribution. Mean = 18.9115 StdDev = 4.4903

Fitted estimators (with ML estimates of variance):

Cluster: 0 Prior probability: 0.6453

Attribute: Column1

Normal Distribution. Mean = 29.451 StdDev = 7.4825

Attribute: Column2

Normal Distribution. Mean = 8.2517 StdDev = 3.7716

Cluster: 1 Prior probability: 0.3547

Attribute: Column1

Normal Distribution. Mean = 14.9822 StdDev = 6.011

Attribute: Column2

Normal Distribution. Mean = 19.2561 StdDev = 4.3297

Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      244 ( 65%)
```

```
1      129 ( 35%)


Log likelihood: -6.74358
```



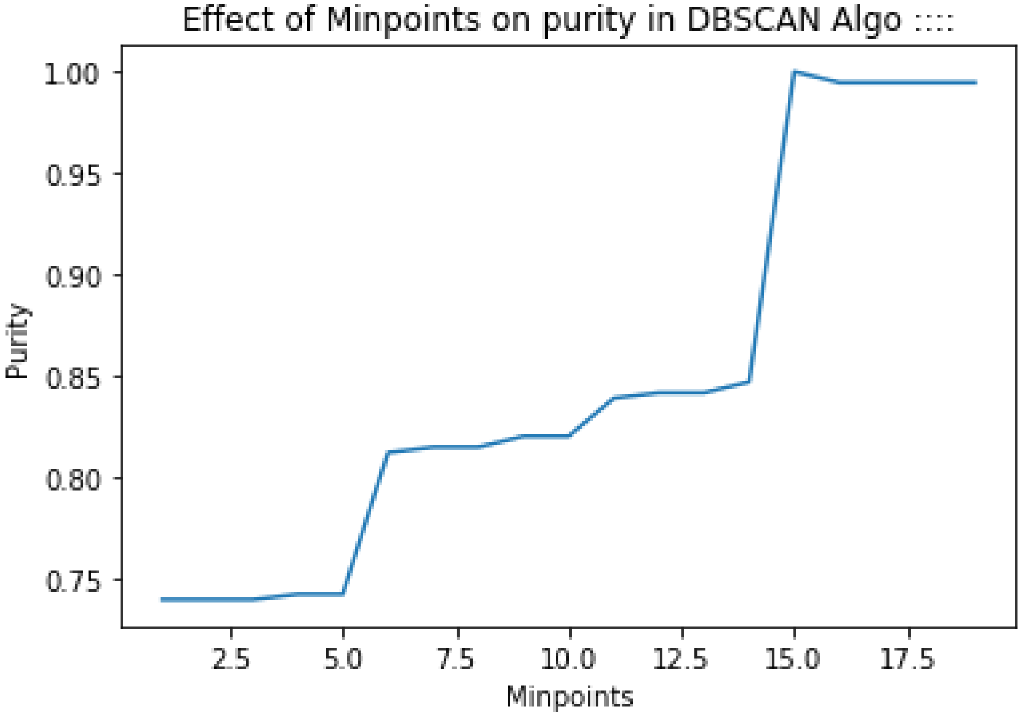
The python code to run DBSCAN on Jain dataset is in the file CS18M524_PA3.ipynb

```
The purity with DBSCAN is ::::  0.739946380697051
```

Effect of varying minpoints on purity:-

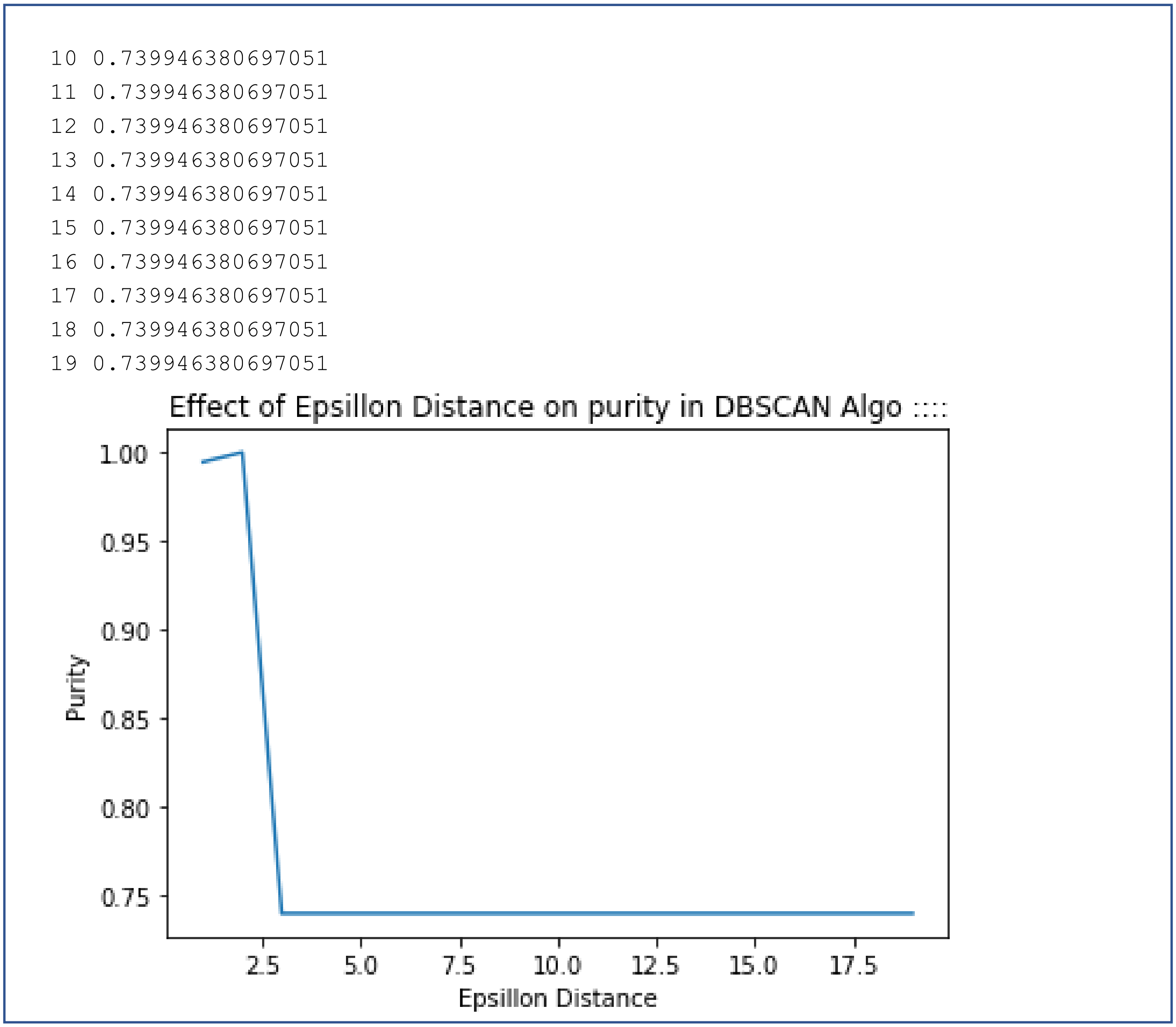
```
*********************Varying min points from 1 to 20*********************
Minpoints        Purity
1 0.739946380697051
2 0.739946380697051
3 0.739946380697051
4 0.7426273458445041
5 0.7426273458445041
6 0.8123324396782842
7 0.8150134048257373
8 0.8150134048257373
9 0.8203753351206434
10 0.8203753351206434
```

```
11 0.839142091152815
12 0.8418230563002681
13 0.8418230563002681
14 0.8471849865951743
15 1.0
16 0.9946380697050938
17 0.9946380697050938
18 0.9946380697050938
19 0.9946380697050938
```



Effect of varying epsilon distance on purity

```
*********************Varying epsillon distance from  from 1 to 20************
*********
Epsillon        Purity
1 0.9946380697050938
2 1.0
3 0.739946380697051
4 0.739946380697051
5 0.739946380697051
6 0.739946380697051
7 0.739946380697051
8 0.739946380697051
9 0.739946380697051
```



```
10 0.739946380697051
11 0.739946380697051
12 0.739946380697051
13 0.739946380697051
14 0.739946380697051
15 0.739946380697051
16 0.739946380697051
17 0.739946380697051
18 0.739946380697051
19 0.739946380697051
```

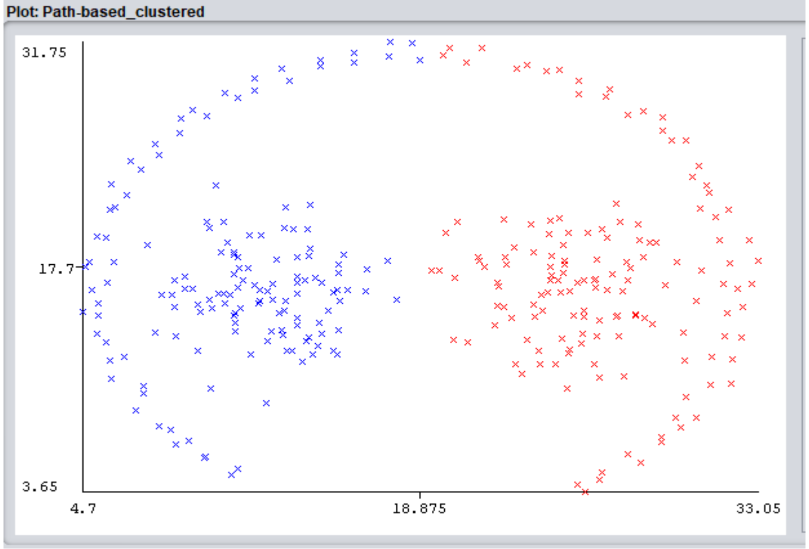
5.

a)Results of running DBSCAN on Path-based dataset

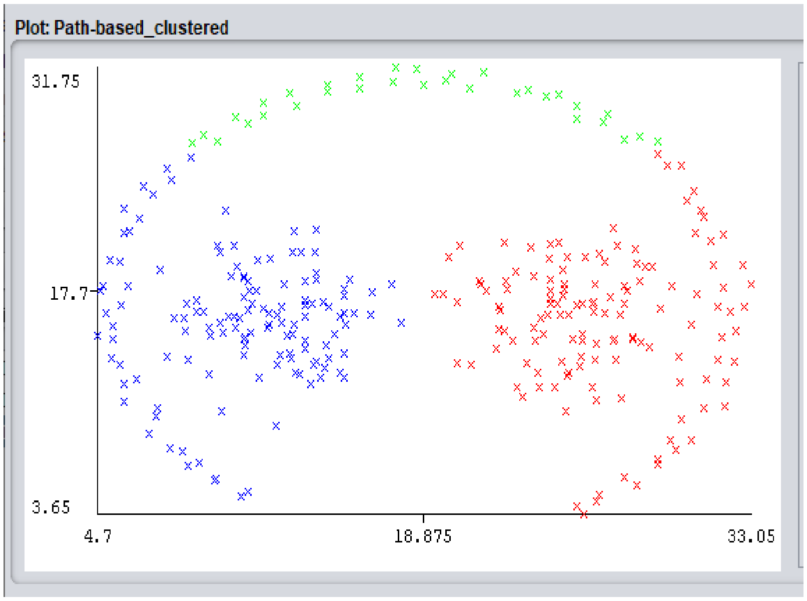
The Weka result has been stored in the **Weka Results** folder by the filename **DBSCAN_ON_Path_Based_Weka**.

The visual representation of the same is as follows :-

This is for N=2 which is selected by default.



For N= 3 the visualization is as follows. Results have been amended to the same file.

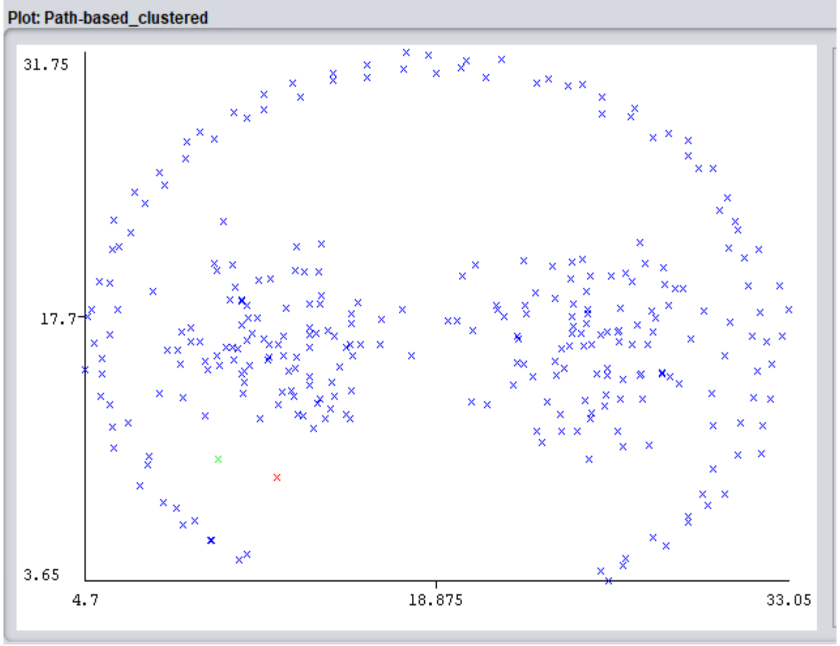


b) Results of running Hierarchical Clustering on Path-based dataset

i) Single Linkage (num Clusters chosen as 3)

The Weka result has been stored in the **Weka Results** folder by the filename **Single_Linkage_Hierarchial_on_Path_Based.**

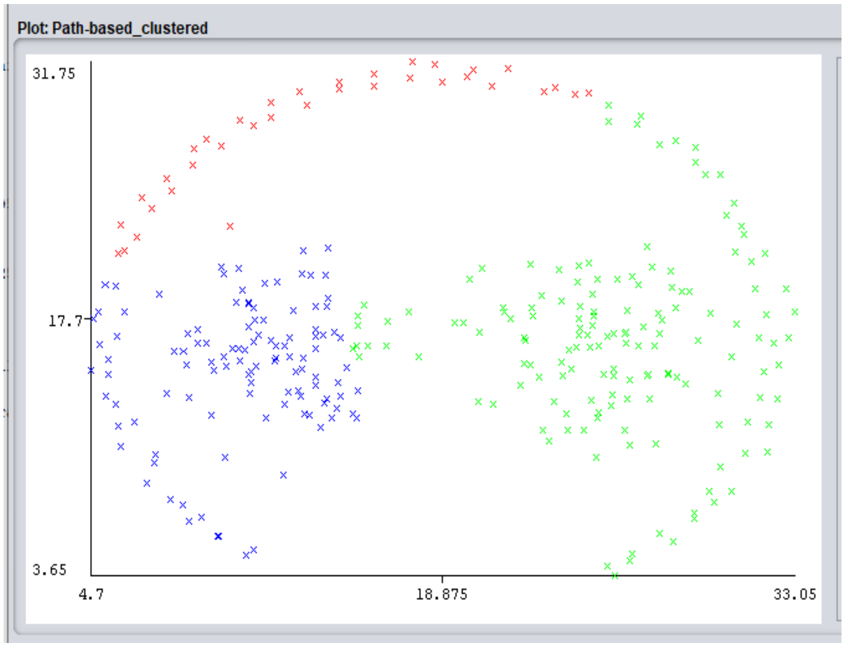
The visual representation of the same is as follows :-



ii) Complete Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Complete_Linkage_Hierarchial_on_Path_Based.**

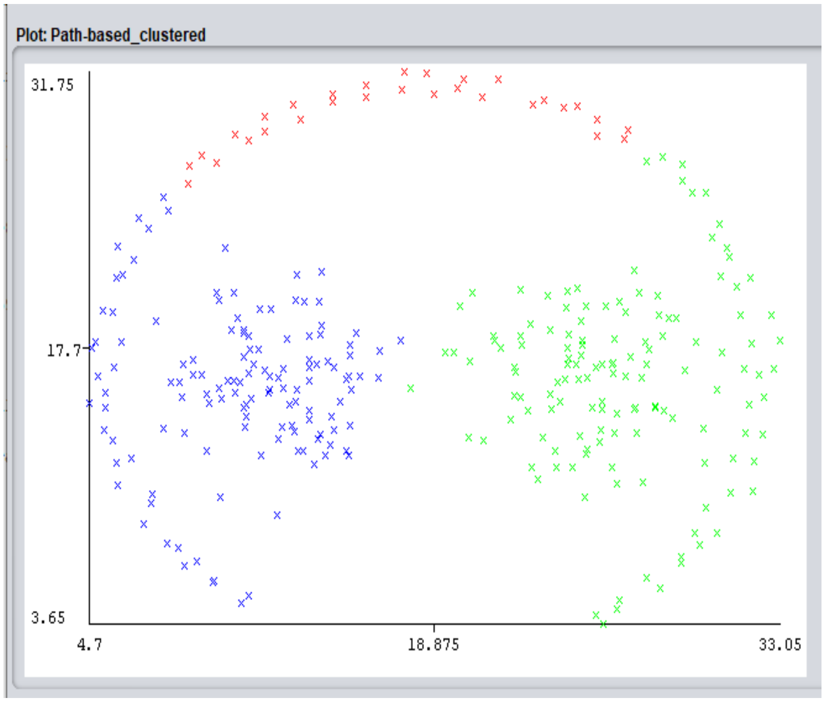
The visual representation of the same is as follows :-



iii) Average Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Average_Linkage_Hierarchial_on_Path_Based.**

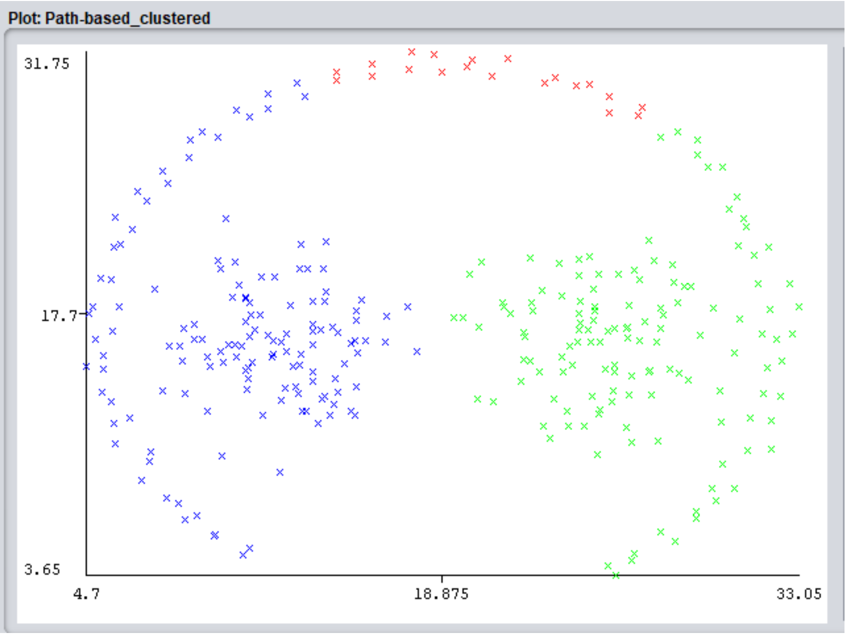
The visual representation of the same is as follows :-



iv) Mean Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Mean_Linkage_Hierarchial_on_Path_Based.**

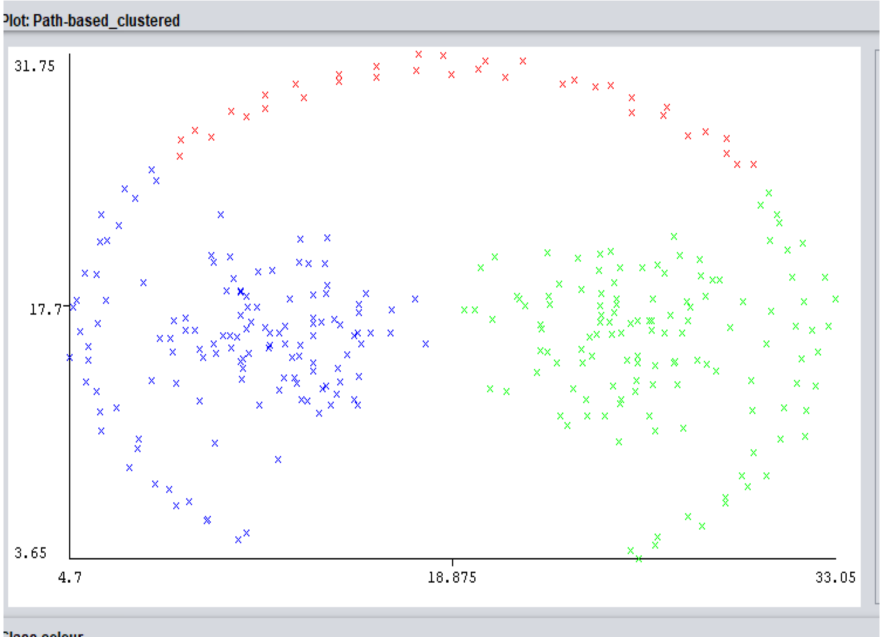
The visual representation of the same is as follows :-



v) Centroid Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Centroid_Linkage_Hierarchial_on_Path_Based.**

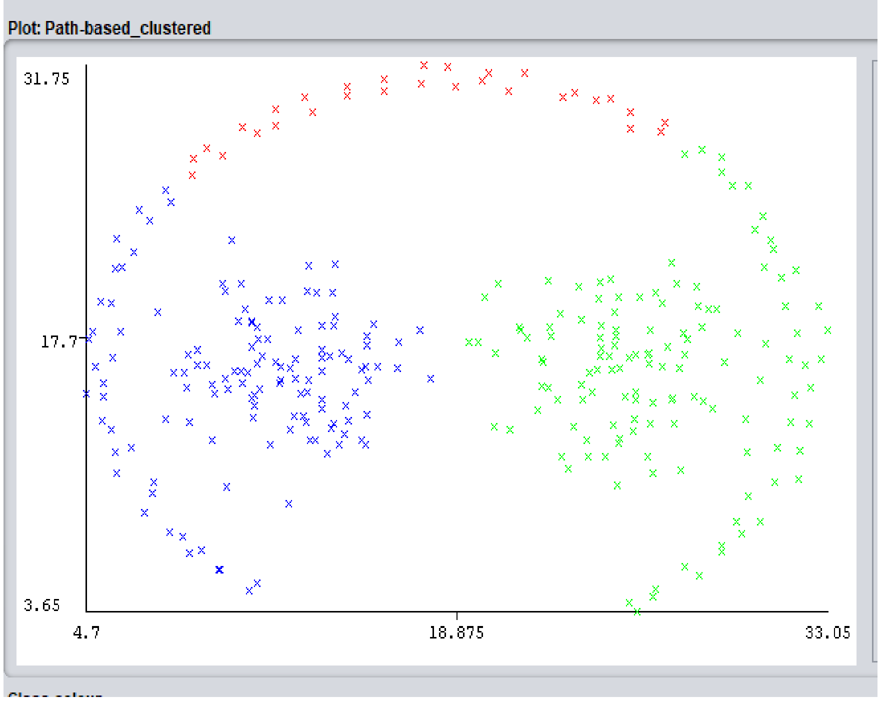
The visual representation of the same is as follows :-



vi) WARD Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **WARD_Linkage_Hierarchial_on_Path_Based.**

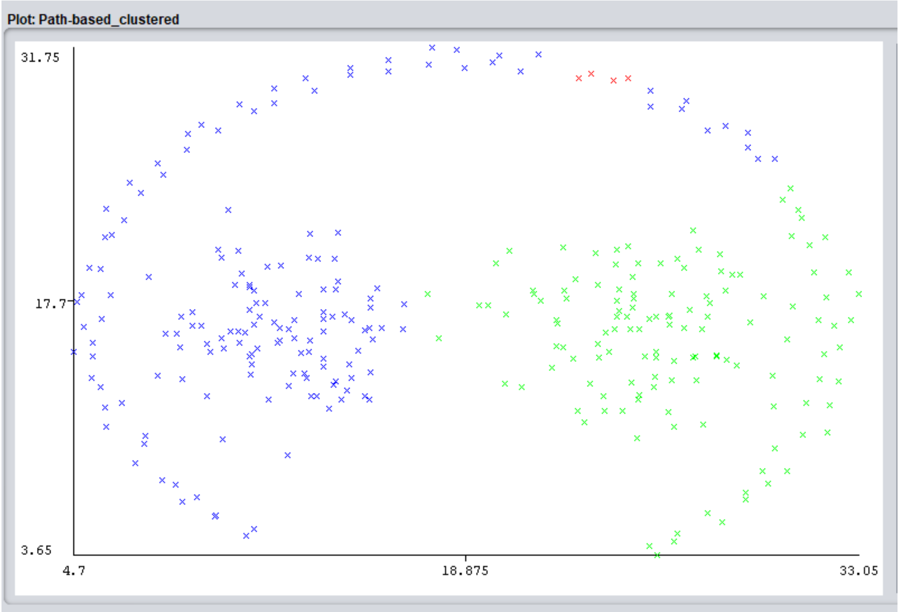
The visual representation of the same is as follows :-



vii) ADJCOMPLETE LINKAGE

The Weka result has been stored in the **Weka Results** folder by the filename **ADJCOMPLETE_Linkage_Hierarchial_on_Path_Based.**

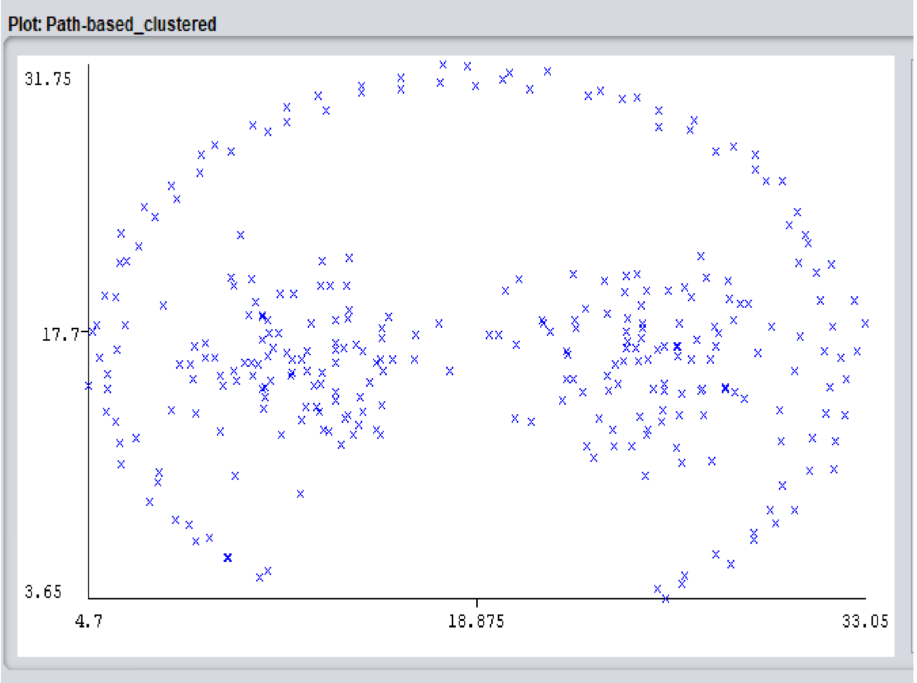
The visual representation of the same is as follows :-



viii) Neighbor Joining Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Near_Neighbors_Hierarchial_on_Path_Based.**

The visual representation of the same is as follows :-



In Summary, none of the algorithms have been able to identify the 3 clusters correctly but Centroid and WARD Linkage is at least close. DBSCAN with N =2 does not work well in the Path Based dataset as it is forms only 2 clusters instead of 3. With N=3 it performs as good as WARD Linkage.

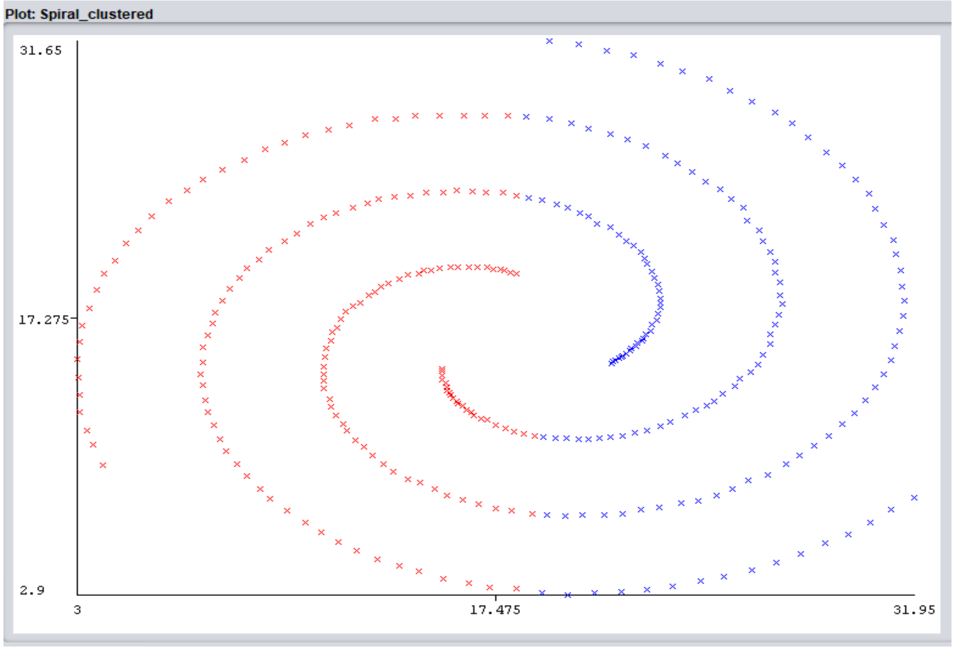
Spiral Dataset

a) **Result of running DBSCAN on Spiral dataset**

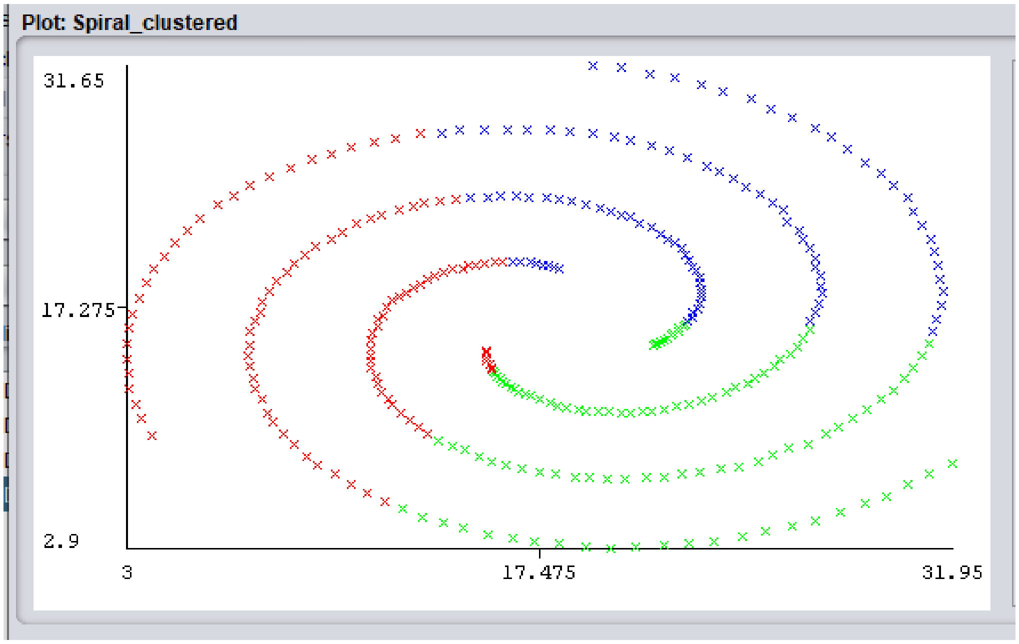
The Weka result has been stored in the **Weka Results** folder by the filename **DBSCAN_ON_Spiral.**

The visual representation of the same is as follows :-

This is for N=2 which is selected by default



For N = 3 the results are as follows. The Weka results have been amended to the same file.

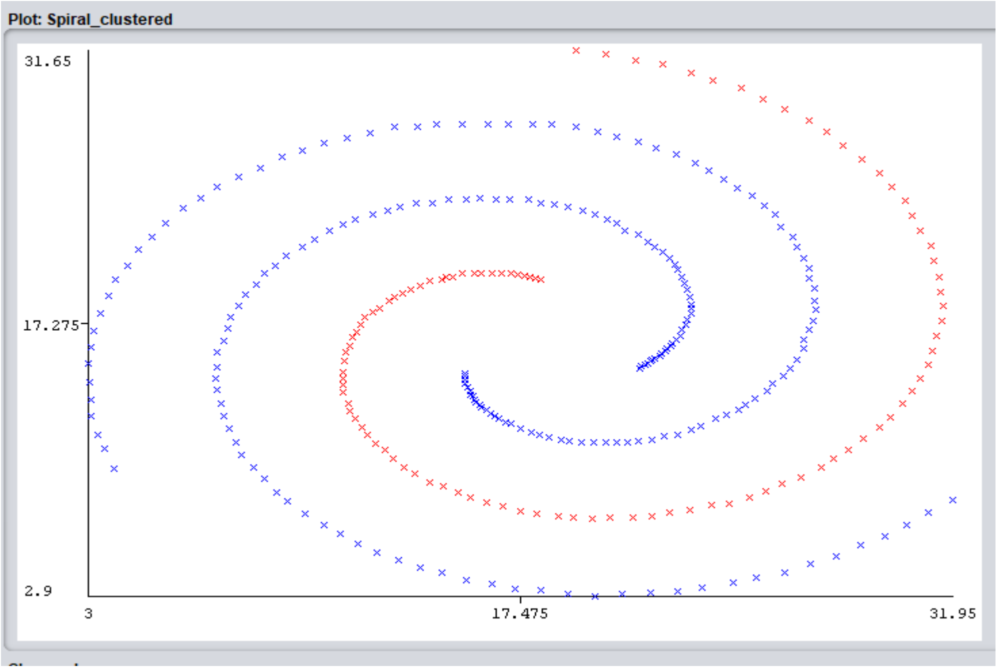


b) Results of running hierarchial clustering on Spiral dataset with different linkages

i) Single Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Single_Linkage_Hierarchial_on_Spiral.**

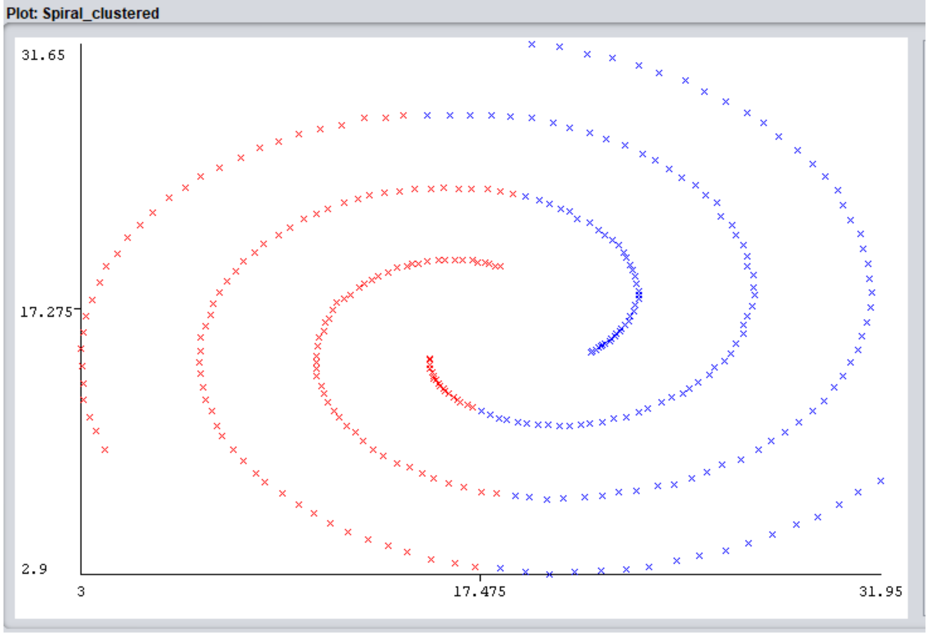
The visual representation of the same is as follows :-



ii) Complete Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Complete_Linkage_Hierarchial_on_Spiral.**

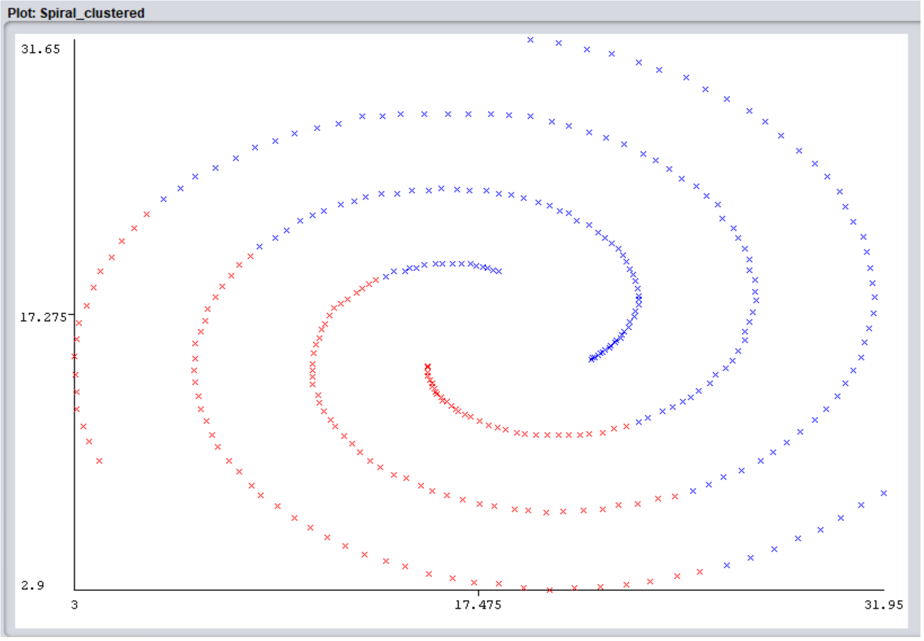
The visual representation of the same is as follows :-



iii) Average Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Average_Linkage_Hierarchial_on_Spiral.**

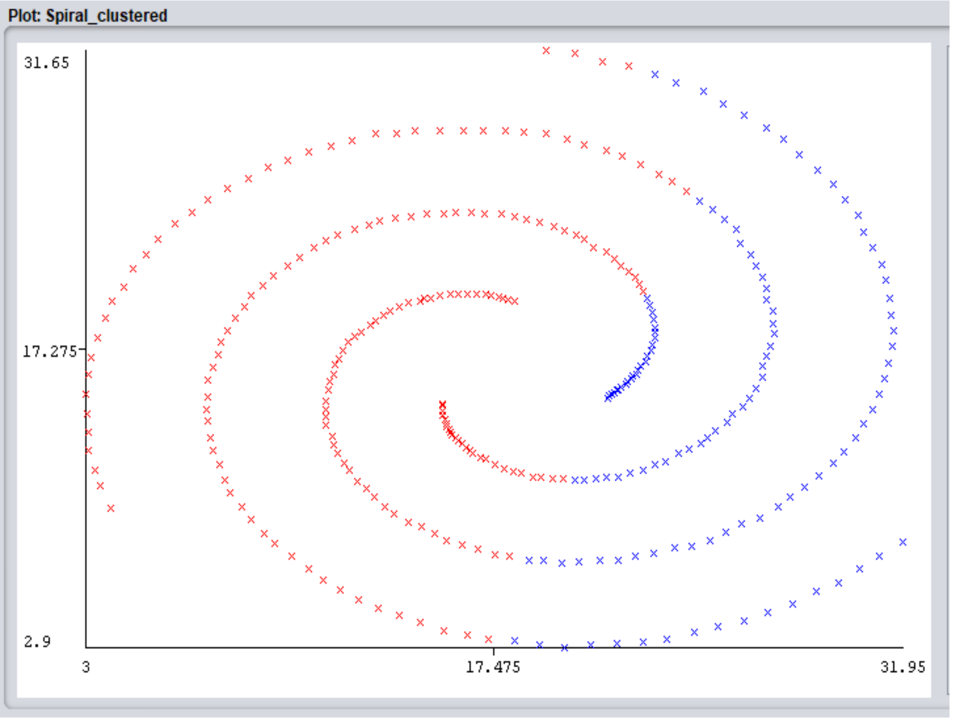
The visual representation of the same is as follows :-



iv) Mean Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Average_Linkage_Hierarchial_on_Spiral.**

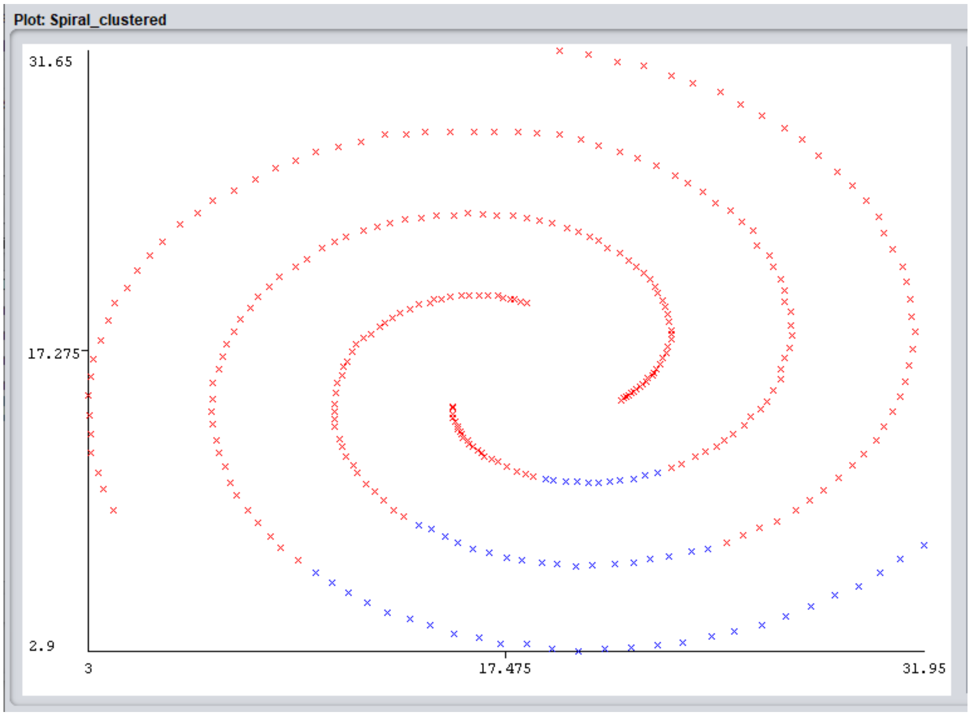
The visual representation of the same is as follows :-



v) Centroid Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Average_Linkage_Hierarchial_on_Spiral.**

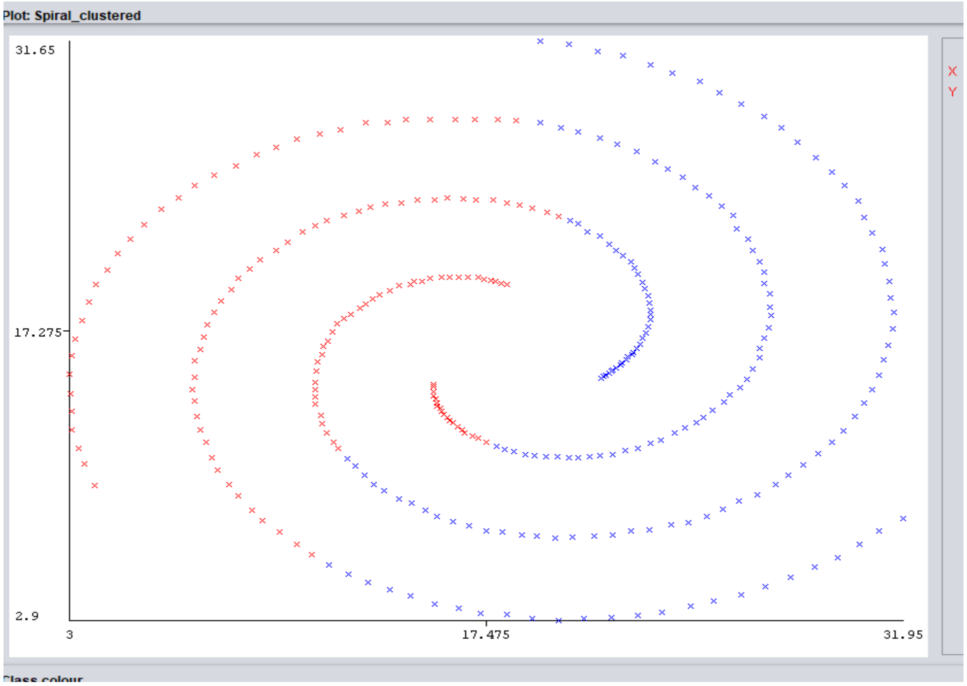
The visual representation of the same is as follows :-



vi) WARD Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **WARD_Linkage_Hierarchial_on_Spiral.**

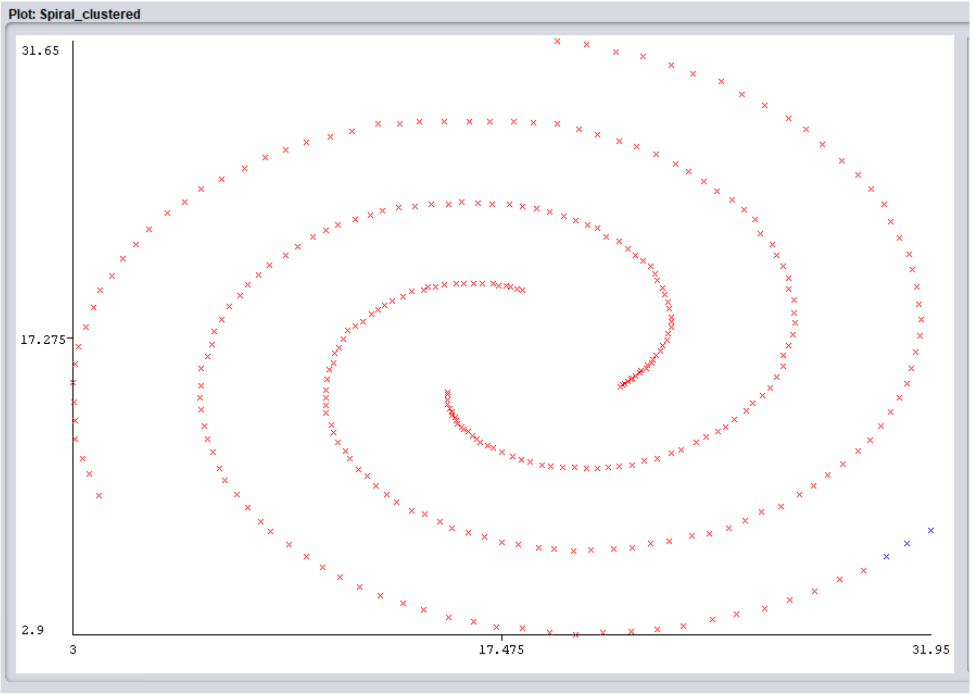
The visual representation of the same is as follows :-



vii) ADJCOMPLETE Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **ADJCOMPLETE_Linkage_Hierarchial_on_Spiral.**

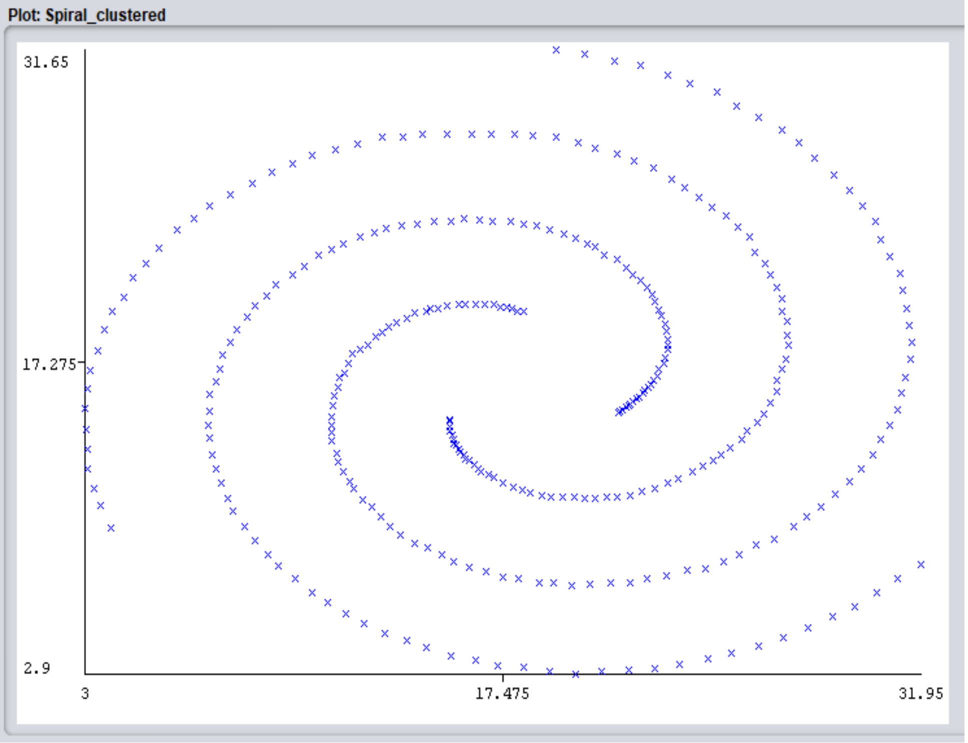
The visual representation of the same is as follows :-



viii) Neighbor Joining Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Near_Neighbors_Hierarchial_on_Spiral.**

The visual representation of the same is as follows :-



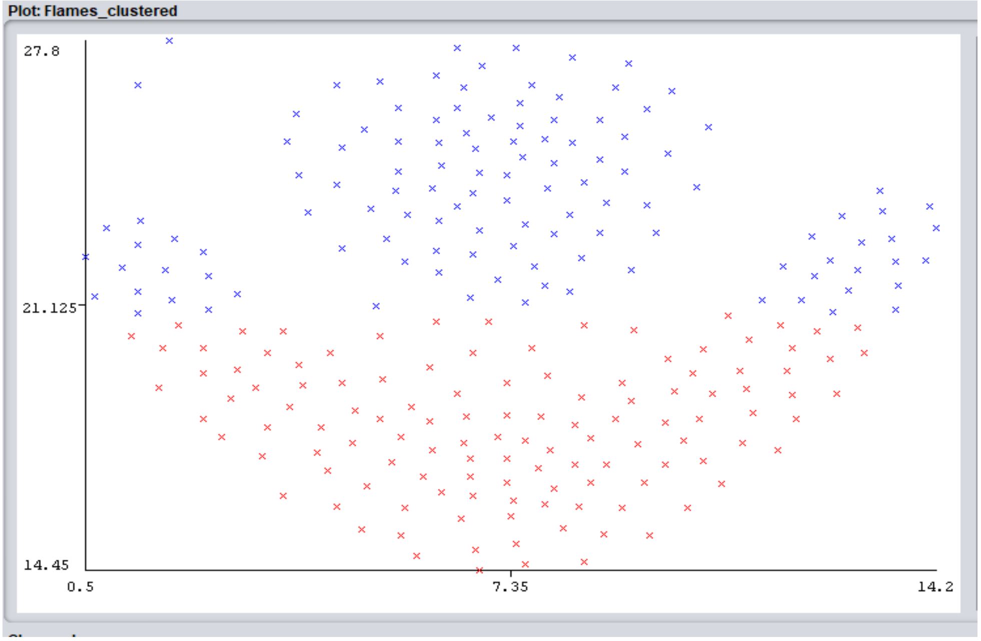
In summary none of the algorithms are able to tell the 3 clusters apart but Single Linkage does a decent job in that it tells at least one cluster correctly. DBSCAN with N=2 detects only 2 clusters and performs poorly but with N= 3 it is the only algorithm able to detect the 3 clusters although they are not correctly detected

Flames Dataset

a) Results of running DBSCAN on Flames Dataset

The Weka result has been stored in the **Weka Results** folder by the filename **DBSCAN_on_Flames.**

The visual representation of the same is as follows :-

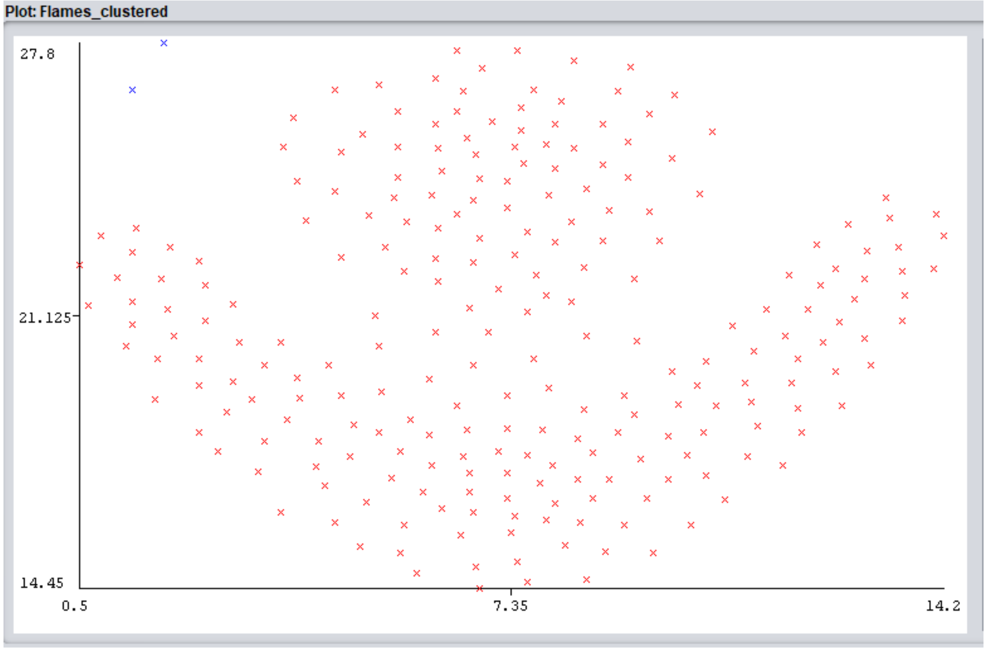


b) Result of running Hierarchial Clustering on Flames Dataset

i) Single Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Single_Linkage_Hierarchial_on_Flames.**

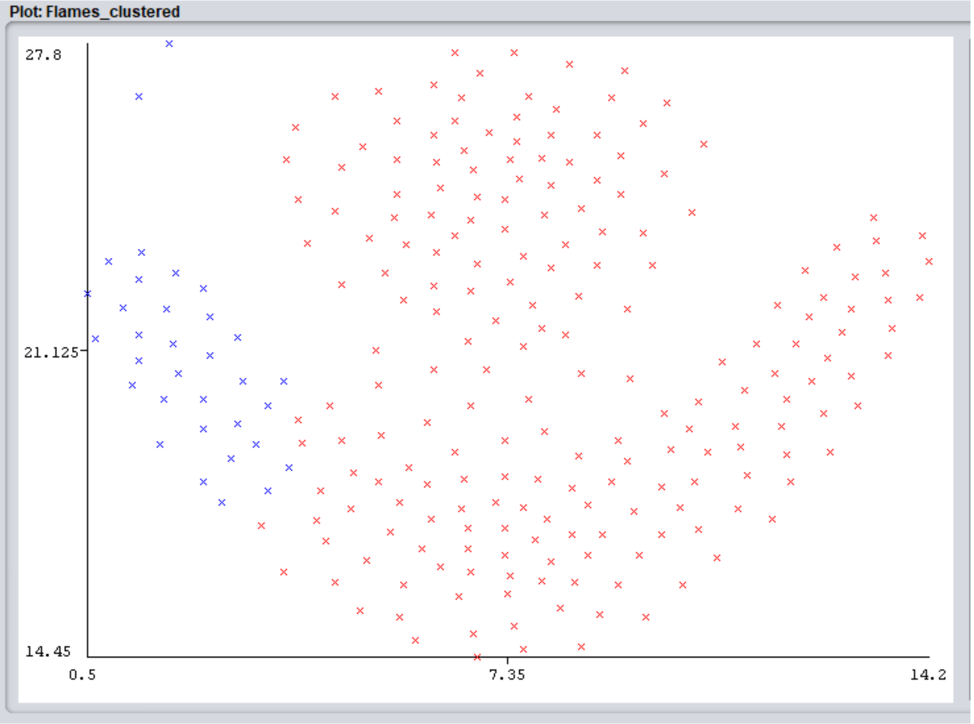
The visual representation of the same is as follows :-



ii) Complete Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Complete_Linkage_Hierarchial_on_Flames.**

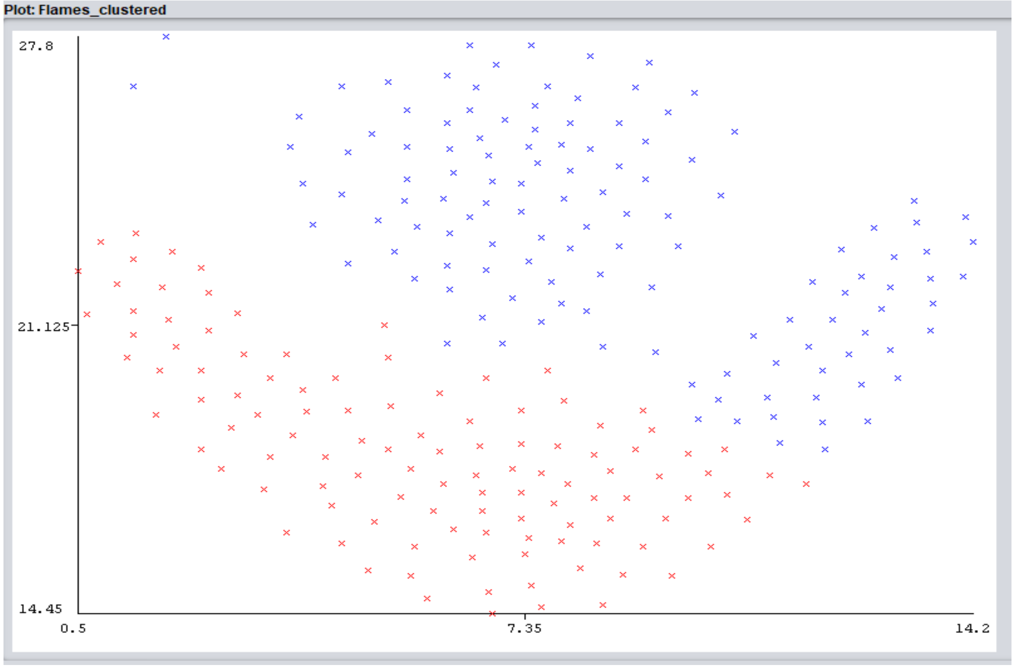
The visual representation of the same is as follows :-



iii) Average Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Average_Linkage_Hierarchial_on_Flames.**

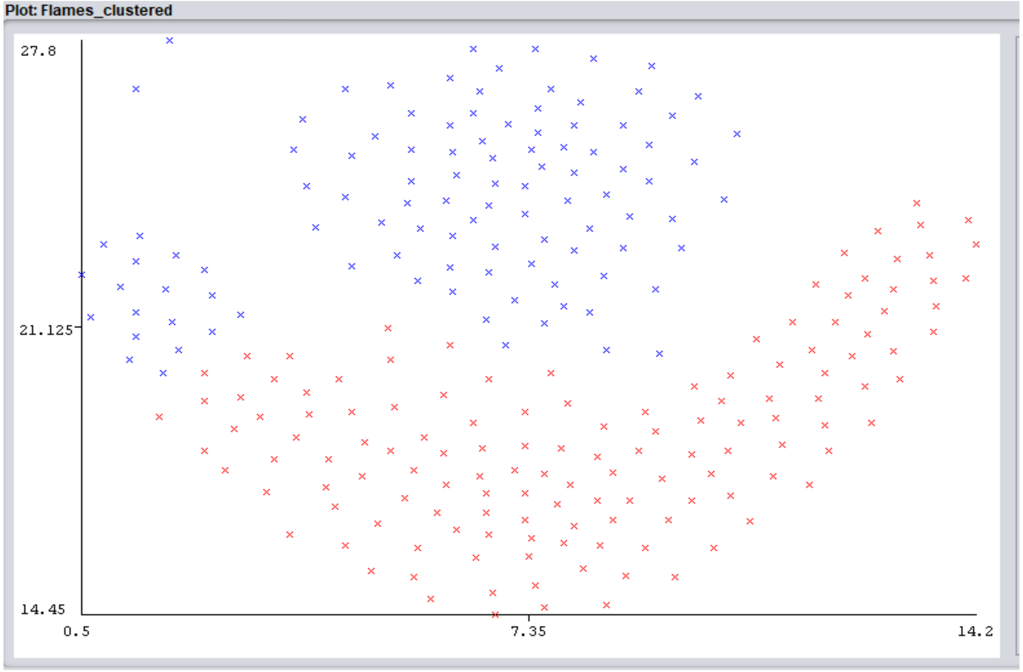
The visual representation of the same is as follows :-



iv) Mean Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Mean_Linkage_Hierarchial_on_Flames.**

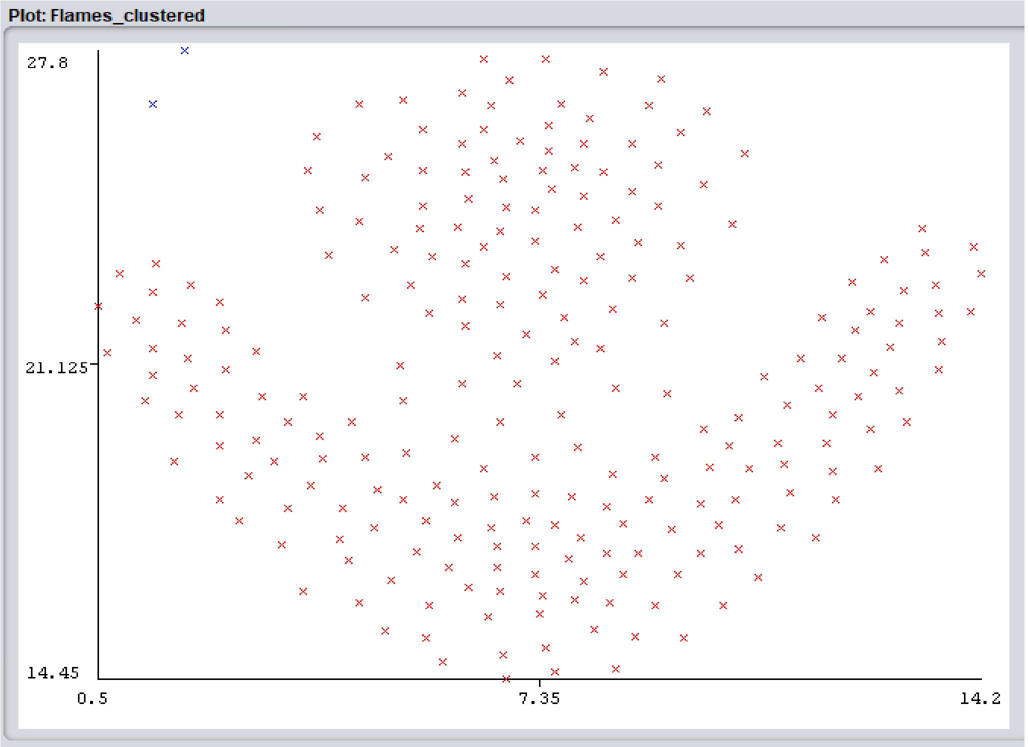
The visual representation of the same is as follows :-



v) **Centroid Linkage**

The Weka result has been stored in the **Weka Results** folder by the filename **Centroid_Linkage_Hierarchial_on_Flames.**

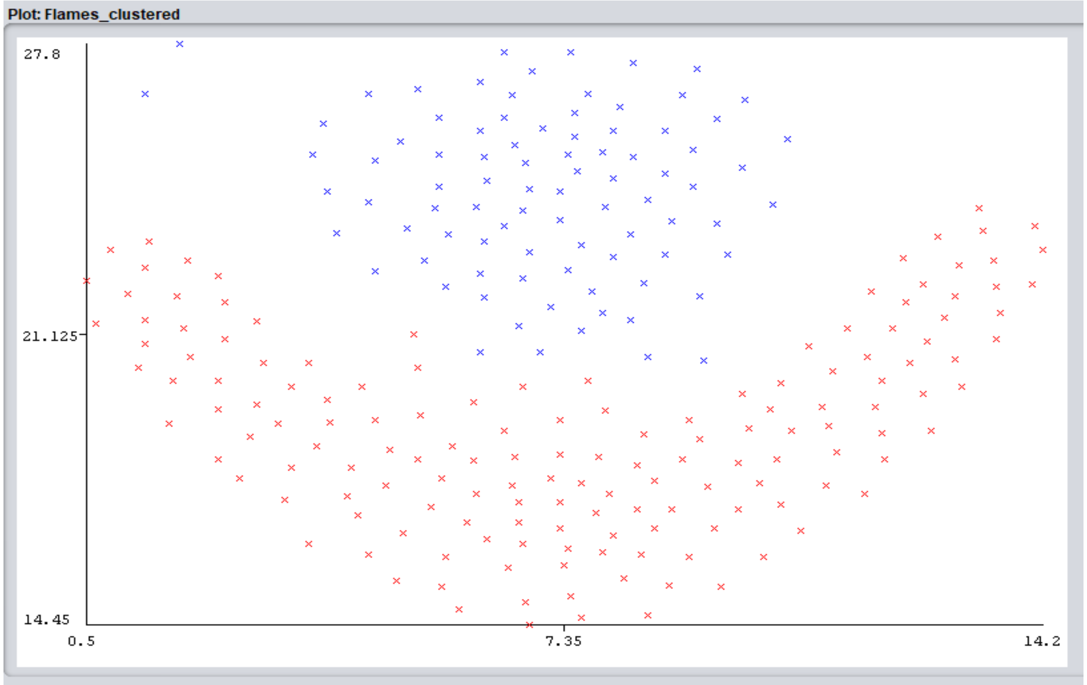
The visual representation of the same is as follows :-



v) **WARD Linkage**

The Weka result has been stored in the **Weka Results** folder by the filename **WARD_Linkage_Hierarchial_on_Flames.**

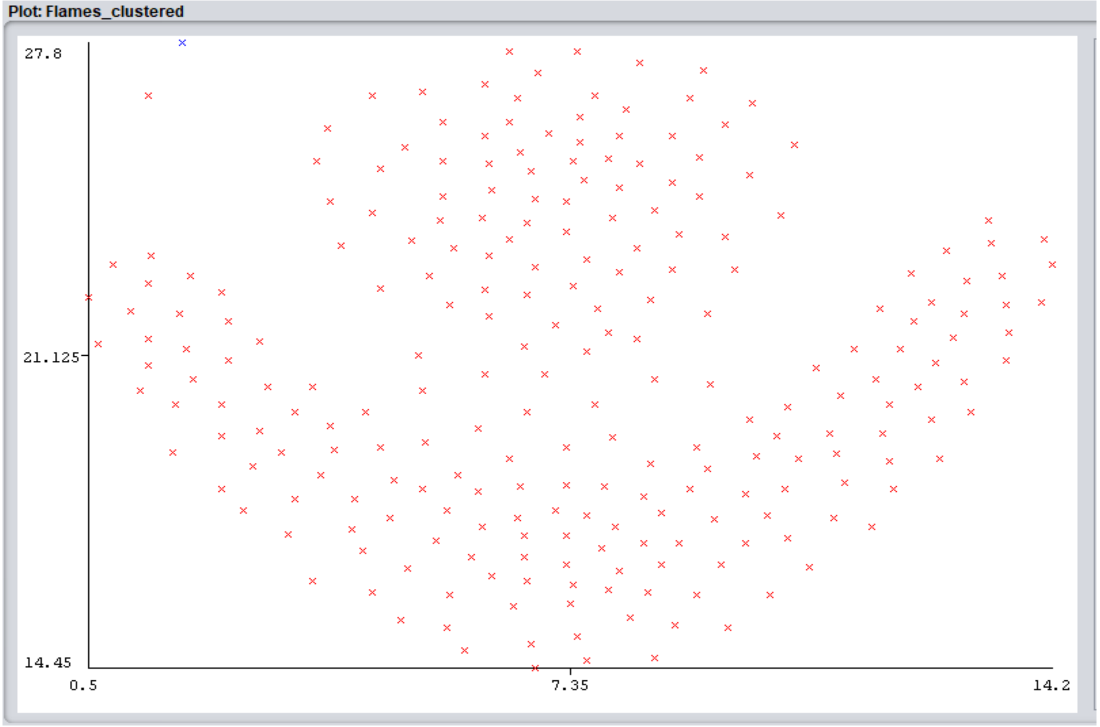
The visual representation of the same is as follows :-



vi) ADJCOMPLETE Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **ADJCOMPLETE_Linkage_Hierarchial_on_Flames.**

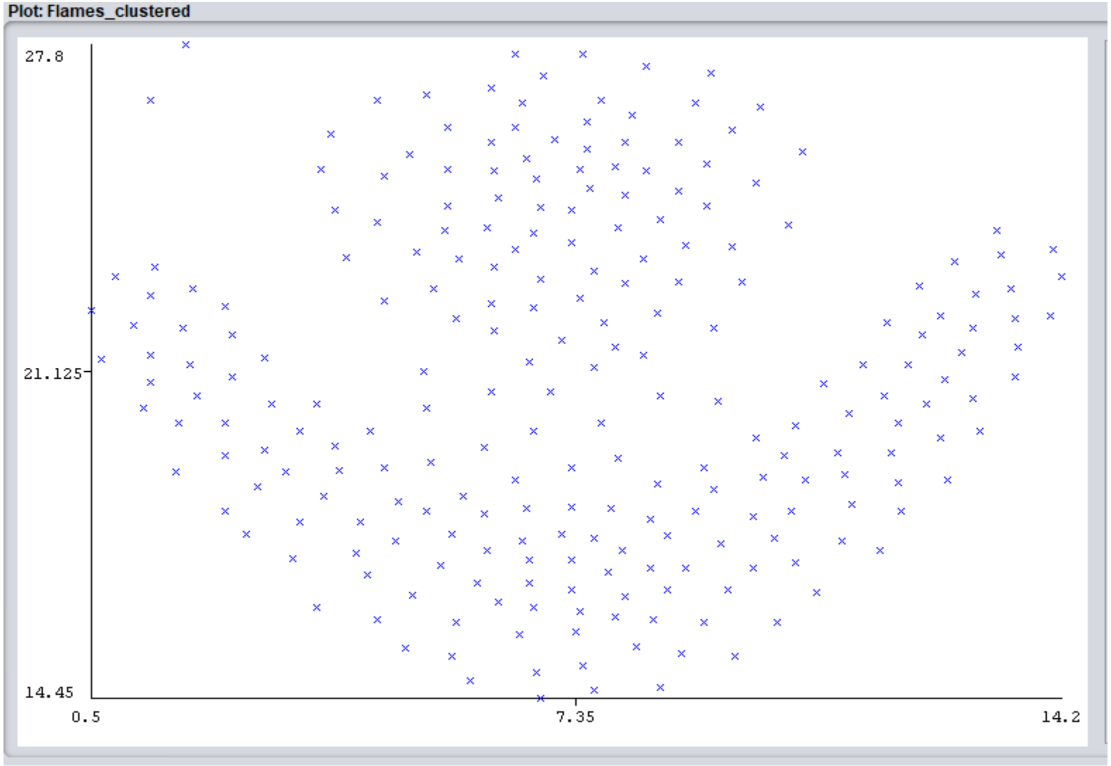
The visual representation of the same is as follows :-



vii) Near Neighbor Joining Linkage

The Weka result has been stored in the **Weka Results** folder by the filename **Near_Neighbors_Linkage_Hierarchial_on_Flames.**

The visual representation of the same is as follows :-

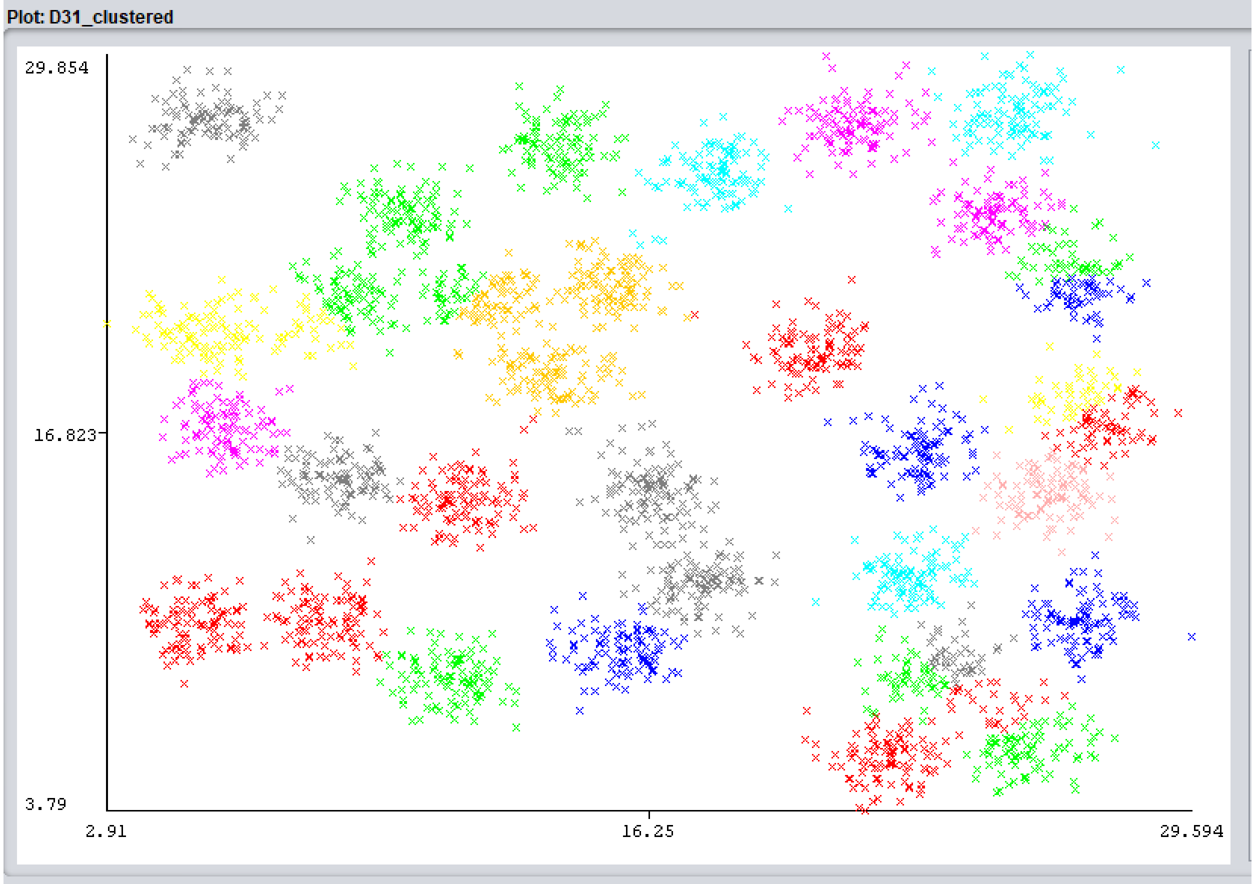


In summary, out of all the above algorithms the Hierarchial Clustering Algorithm with WARD Linkage performs the best with Flmaes dataset as it separates out the 2 clusters correctly. DBSCAN also performs well as in it detects the 2 clusters but some of the points are misclassified. So WARD Linkage is better

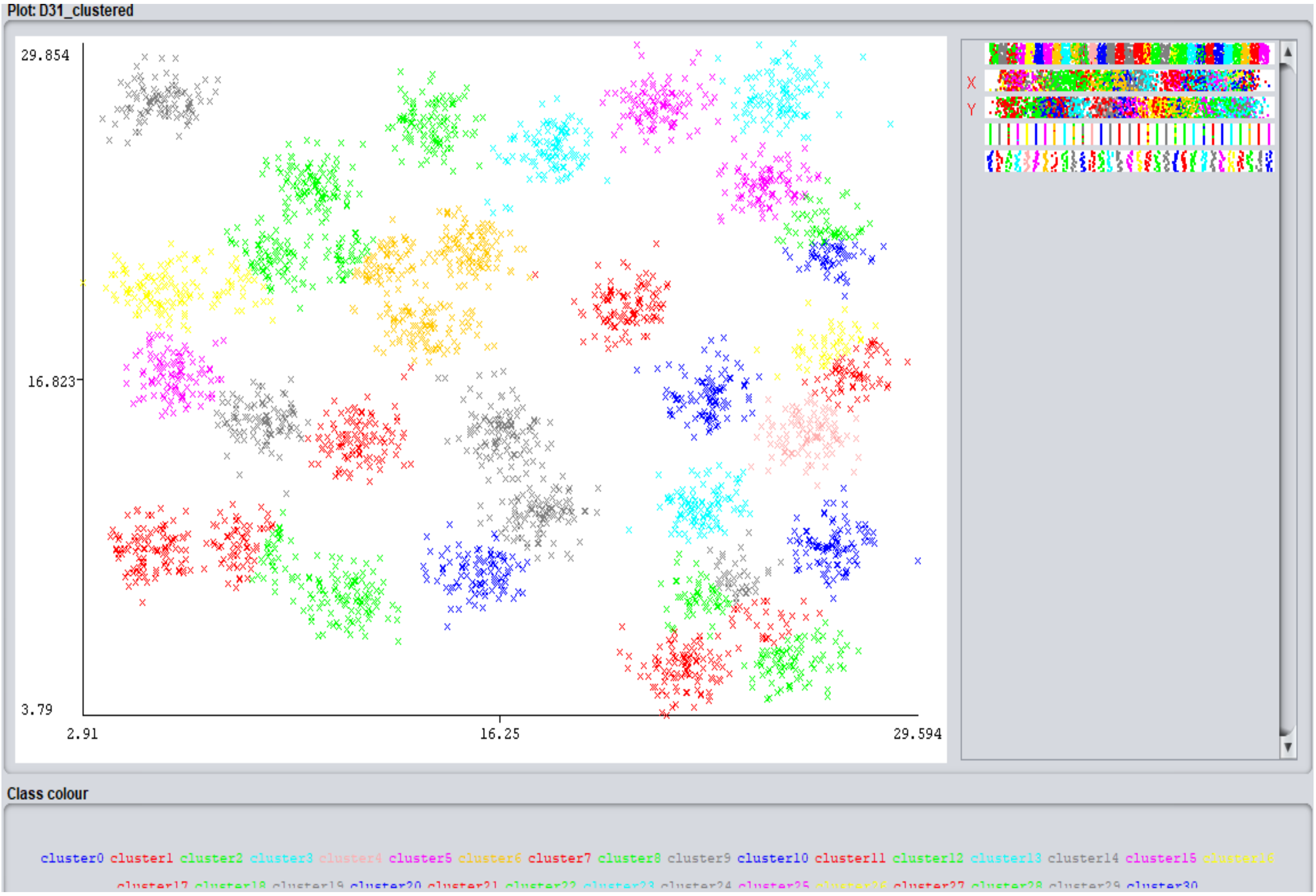
6) Result of running K-Means on D31 dataset with K = 32

The Weka result has been stored in the **Weka Results** folder by the filename **K_Means_On_D31.**

The visual representation of the same is as follows :-



As we see here = 32 not only recovers all clusters but splits one of the clusters into 2. So , now trying with K= 31.


Plot: D31_clustered
29.854
16.823
3.79
2.91
16.25
29.594
X
Y
Class colour
cluster0 cluster1 cluster2 cluster3 cluster4 cluster5 cluster6 cluster7 cluster8 cluster9 cluster10 cluster11 cluster12 cluster13 cluster14 cluster15 cluster16
cluster17 cluster18 cluster19 cluster20 cluster21 cluster22 cluster23 cluster24 cluster25 cluster26 cluster27 cluster28 cluster29 cluster30

With K = 31 itself, we are able to recover all the 31 clusters .

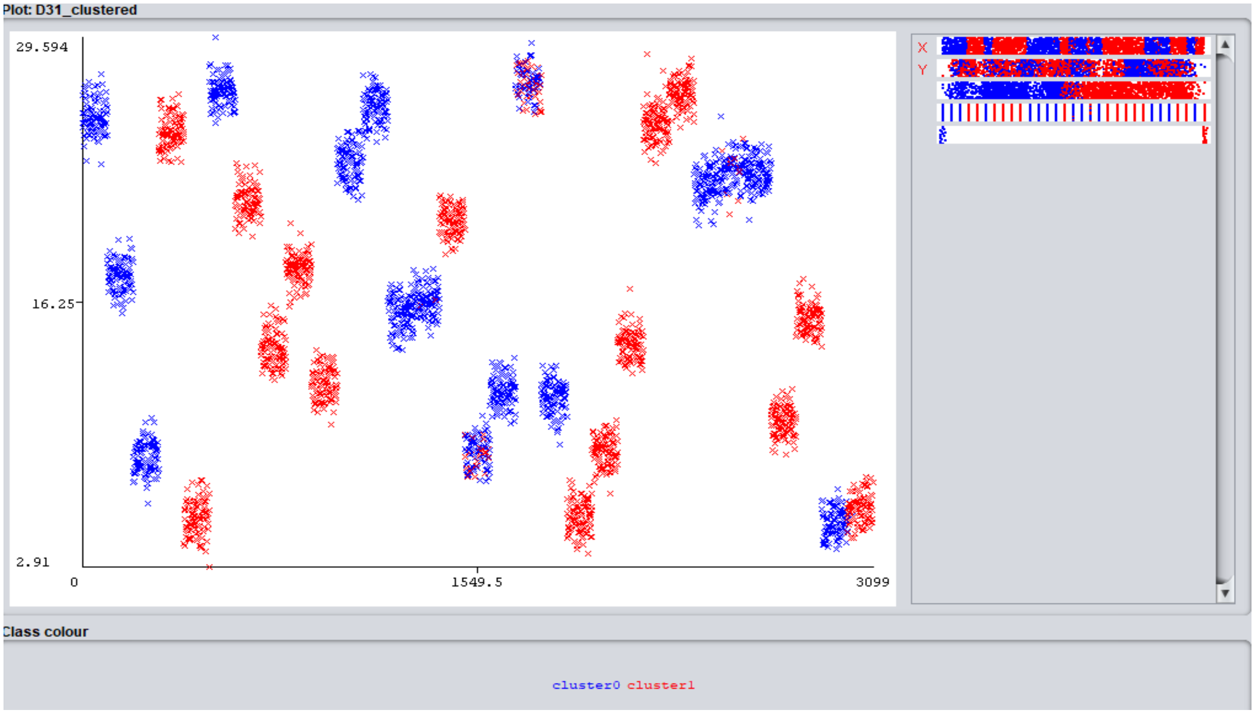
Now applying DBSCAN on D31 dataset.

=== Run information ===

Scheme: weka.clusterers.MakeDensityBasedClusterer -M 1.0E-6 -W weka.clusterers.SimpleKMeans -- -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

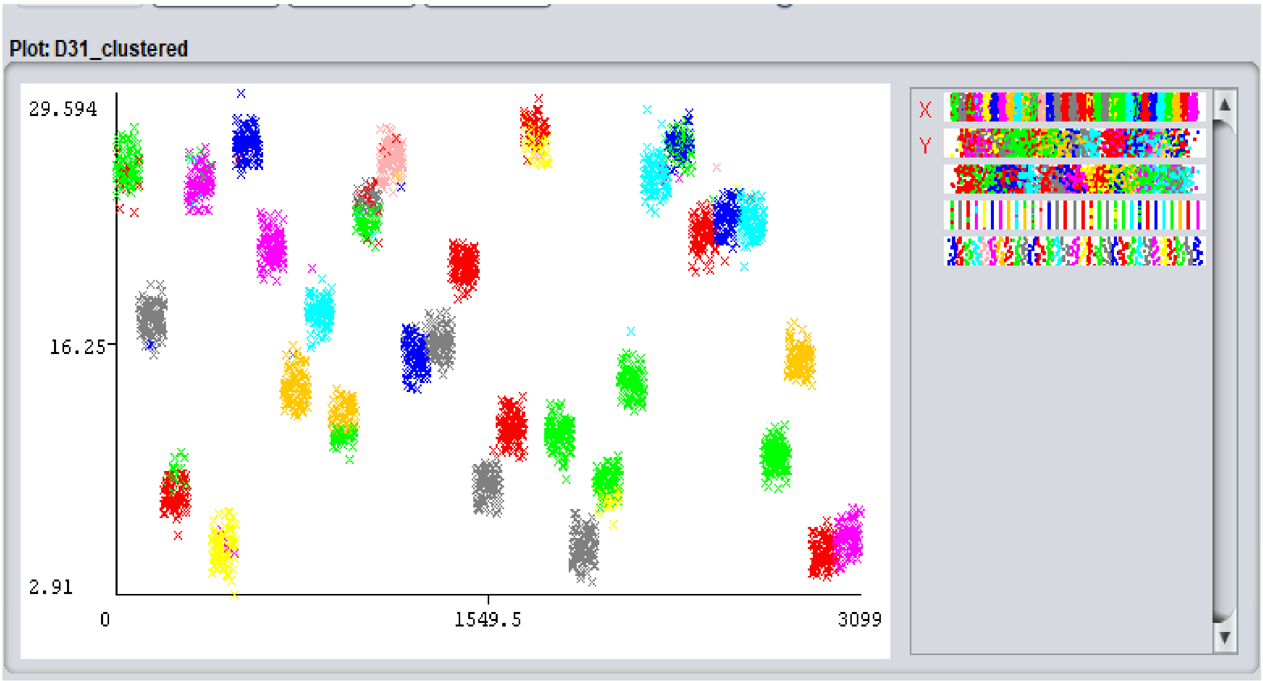
Rest of the result is in the file DBSCAN_On_D31_with_N=2 in the Weka Results folder.

DBSCAN internally calls the K-Means Clustering algorithm and by default K =2. Following is the visualization.



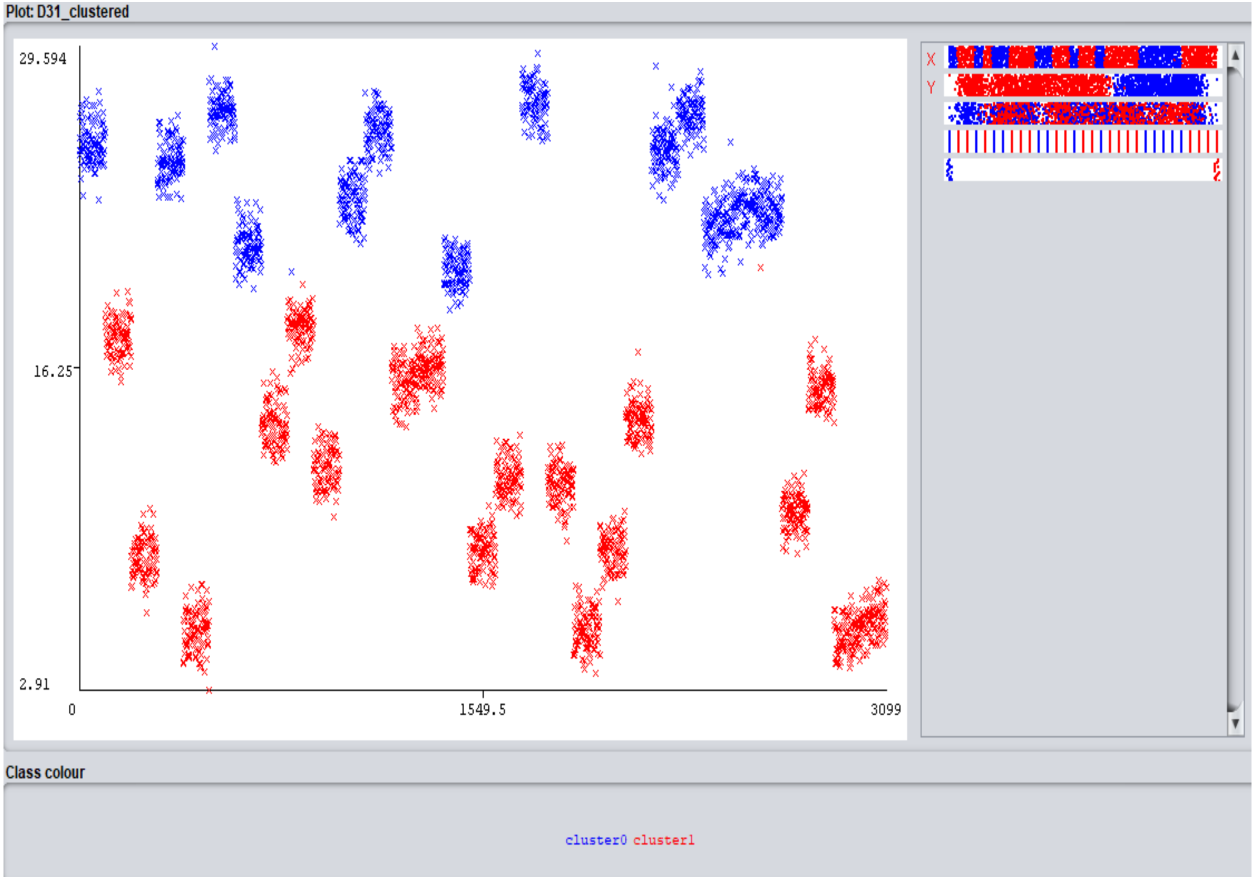
We can safely say that DBSCAN with K=2 (selected internally) does not perform well at all with this dataset. It somehow classifies the 31 different clusters into just 2 clusters.

But if we make K=31 , then it performs quite well. It is able to detect 30 out of 31 clusters. Results in file DBSCAN_on_D31_with_N=31



Now applying hierarchical clustering with WARD linkage on this dataset.

The Weka results have been stored in the Weka Results folder with the filename WARD_Linkage_Hierarchial_on_D31.



Hierarchial clustering with WARD linkage clusters all datapoints into only 2 clusters but we can say that it performs slightly better than DBSCAN with N=2 (internally) since the clusters do not look as random. It also however fails to recover the 31 clusters.