**Q1)**

a)

K means algorithm with default parameters. Gives

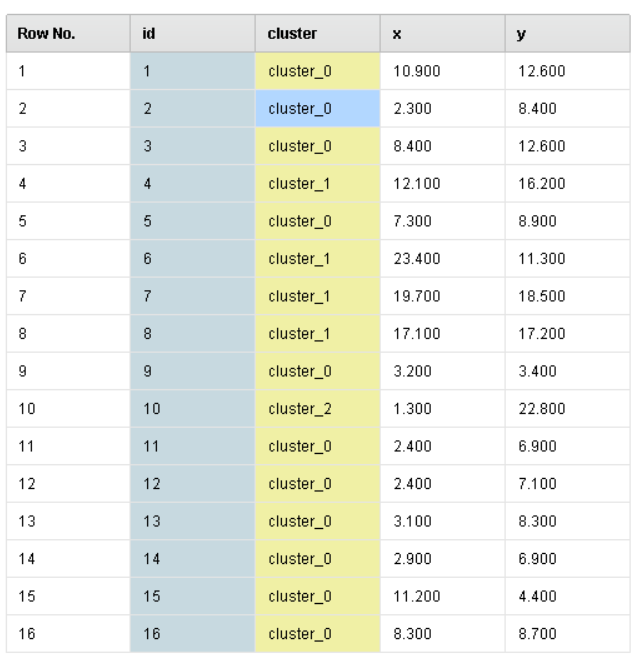
**Cluster Model**

Cluster 0: 11 items

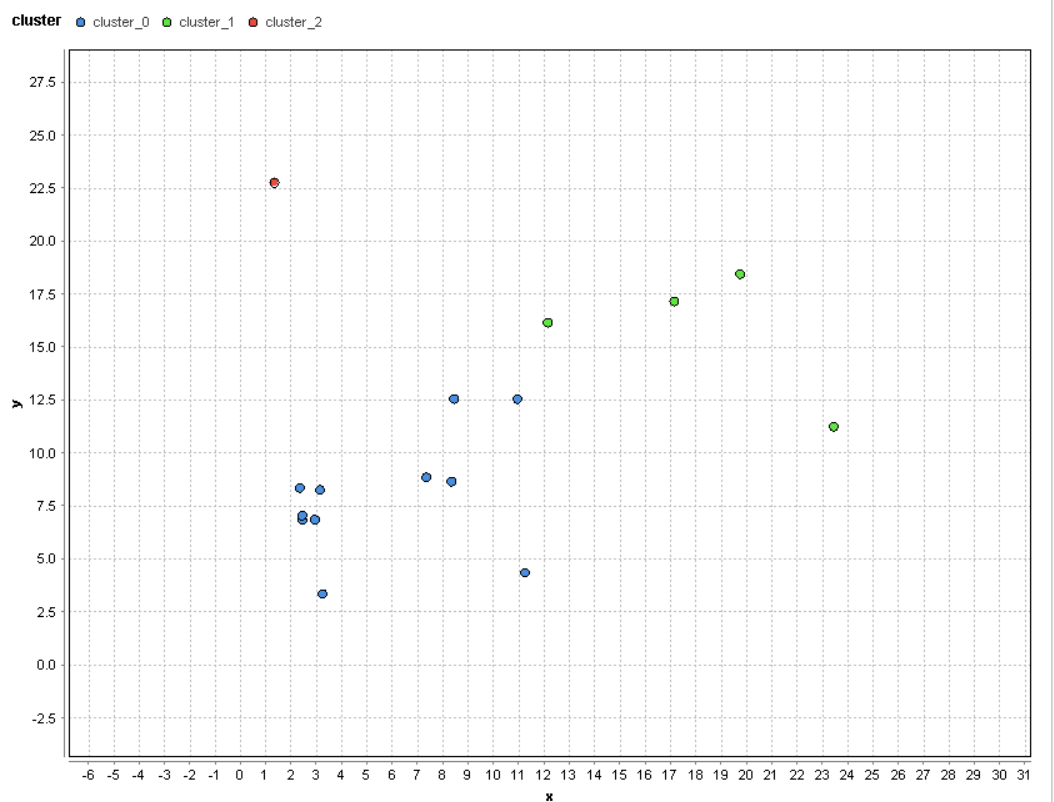
Cluster 1: 4 items

Cluster 2: 1 items

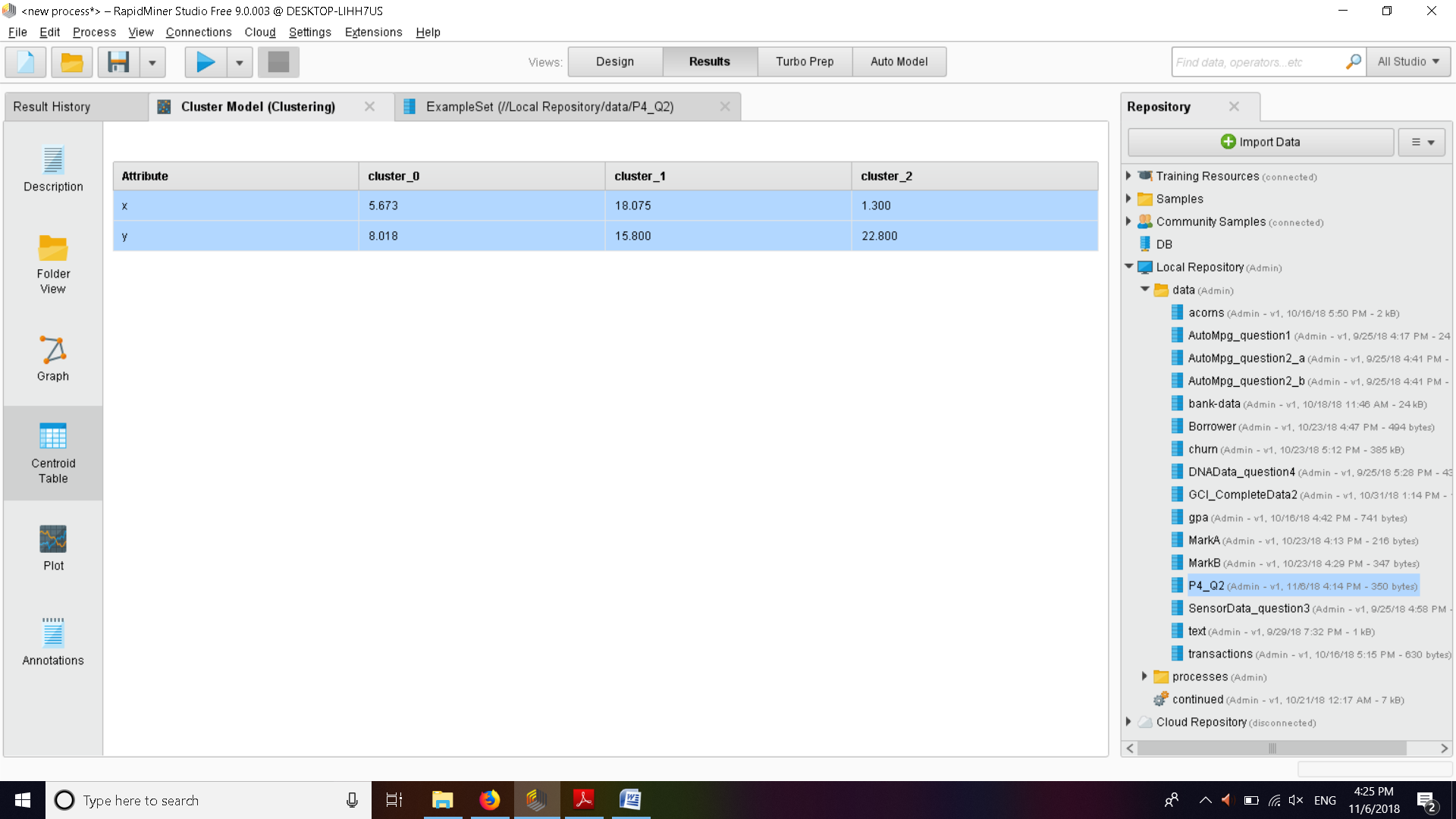
Total number of items: 16

Assignment of clusters is:

Cluster plots:



Cluster centroids:



One could argue the points in the middle (11,12.5),(12,16) could belong to either cluster\_0 or cluster\_1. There is slight vagueness of the results. This is because the dataset itself has some ambiguity and the clusters are not clear.

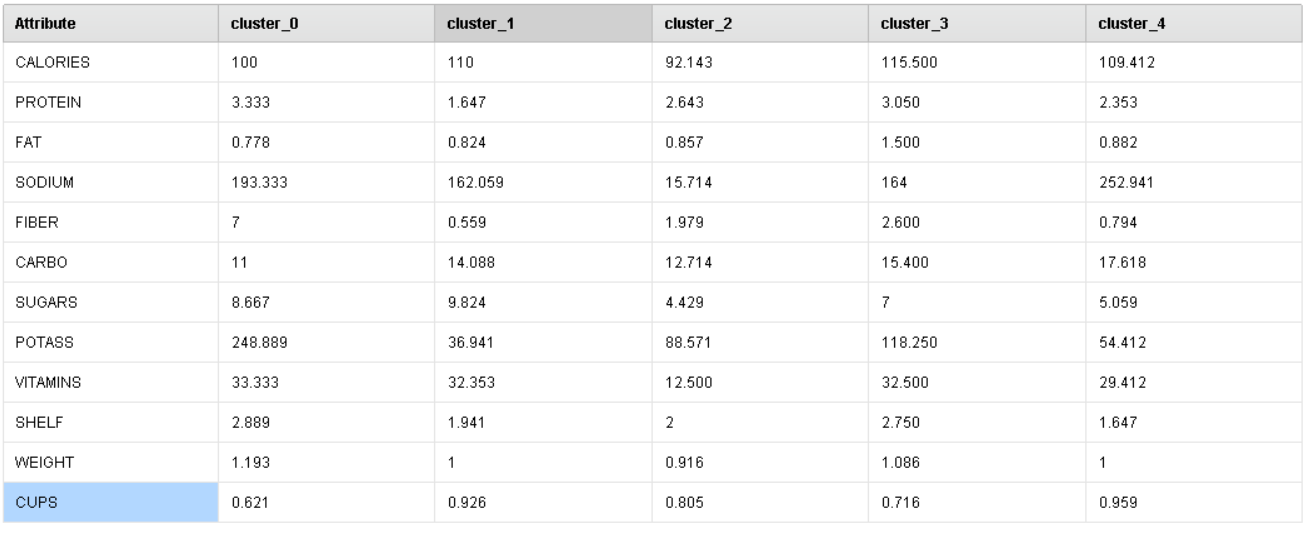
K means algorithm can also produce different results in different runs depending on the initial centers. However K means algorithm is very fast and can work on high dimensional data easily. This leads to clustering in a fast manner.

However you need to select k for good clustering. In the above example one could say that there are 4 clusters and set k as 4. This selection is crucial for the accuracy of kmeans.

**Q2**

With optimization steps 100 we get clusters:

As



With optimization steps 10,000 we get the same clusters as above. This implies that the clusters can be found correctly in 100 iterations. This implies the actual data clusters are not that ambiguous

Checking the performance of clustering for different k values.  
With k=3

**PerformanceVector**

PerformanceVector:

Avg. within centroid distance: -4541.289

Avg. within centroid distance\_cluster\_0: -6909.495

Avg. within centroid distance\_cluster\_1: -4159.237

Avg. within centroid distance\_cluster\_2: -3591.385

Davies Bouldin: -0.806

With k=5

**PerformanceVector**

PerformanceVector:

Avg. within centroid distance: -2878.473

Avg. within centroid distance\_cluster\_0: -5630.443

Avg. within centroid distance\_cluster\_1: -2528.119

Avg. within centroid distance\_cluster\_2: -2646.517

Avg. within centroid distance\_cluster\_3: -2715.571

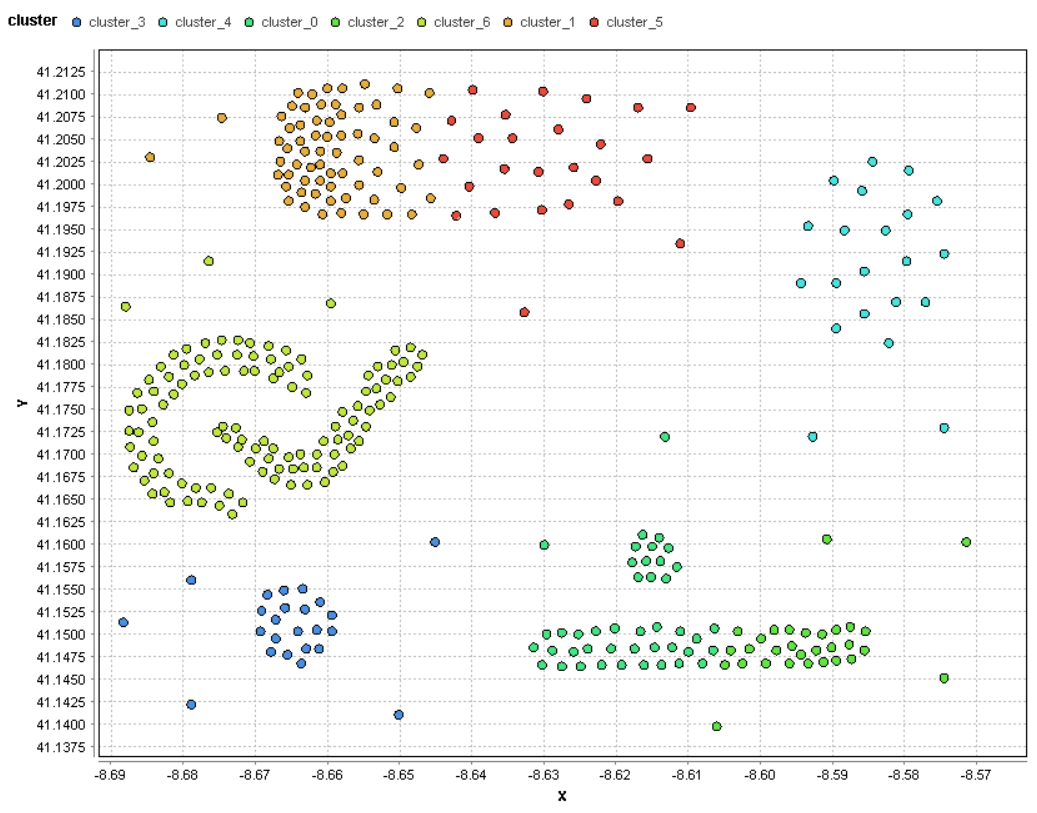
Avg. within centroid distance\_cluster\_4: -2154.575

Davies Bouldin: -0.948

The results of k=5 result in a lower davies bouldin index and the average pairwise distance of points of the clusters is also lower hence k=5 is a better option than k=3.

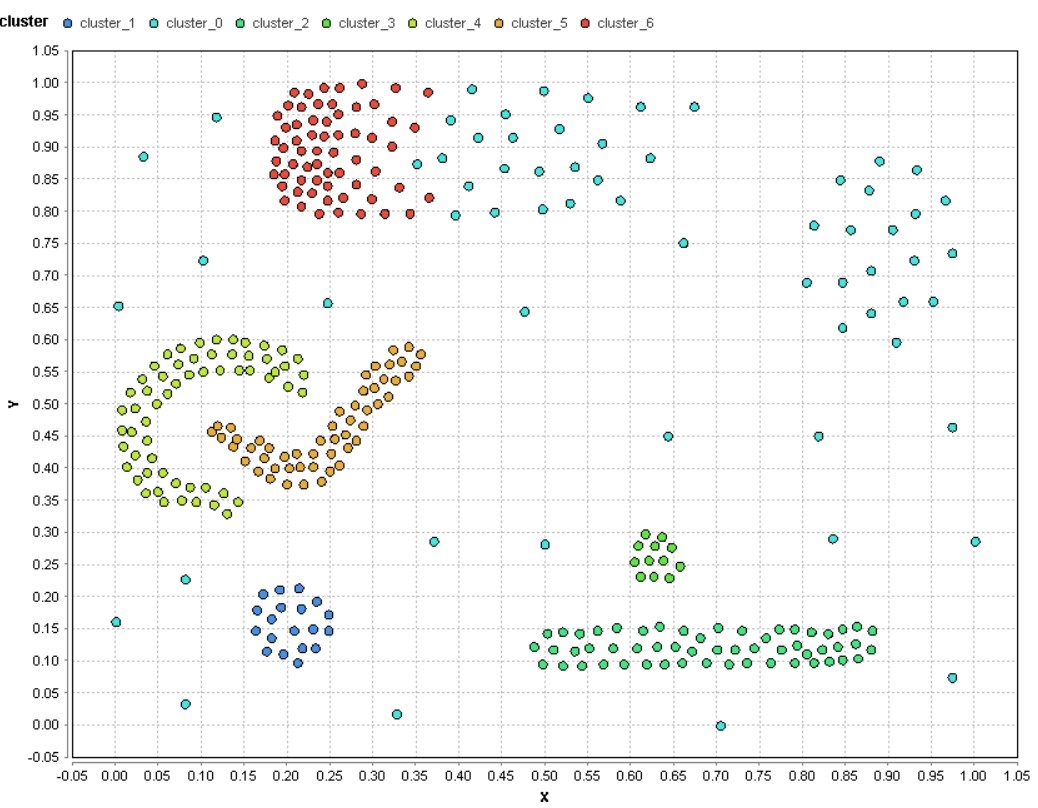
**Q3)**

With k=7 and given parameters we get the plot as:



Most of the clusters are not generated properly by kmeans. As we can see the light green cluster\_6 should have been clustered into two clusters because of their lower pairwise distance of inner points. Also clusters at the top and the bottom are split into two when they should have been the same cluster.

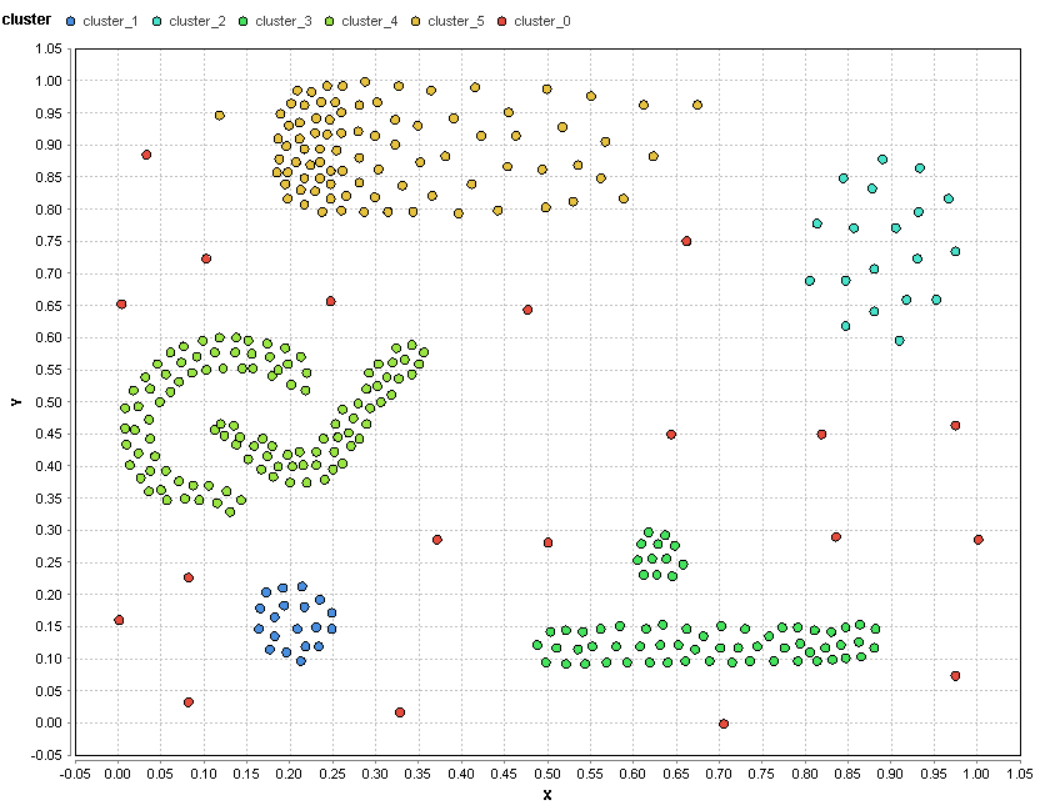
With DBScan algorithm and given parameters we get:



The clusters generated in this method are clearly better than kmeans and it beats most of the drawbacks of kmeans. Almost all of the clusters are assigned properly. The top cluster however is split into two in the middle somewhat abruptly. This could have been fixed by adjusting the epsilon of the algorithm. This would cause the red cluster to properly get assigned. And the light blue cluster would be separate from the red cluster.

4

With epsilon as 0.08 we get clusters as :



This fixes the results of the previous question. However the light green clusters that are tightly enclosed are not split because we increased epsilon hence increasing the size of the neighborhood. This leads to the green clusters being assigned in the same neighborhood.

This implies setting the value of epsilon properly is crucial for proper clustering of the data. Depending on the dataset the epsilon value should be properly assigned for good clustering of the points.