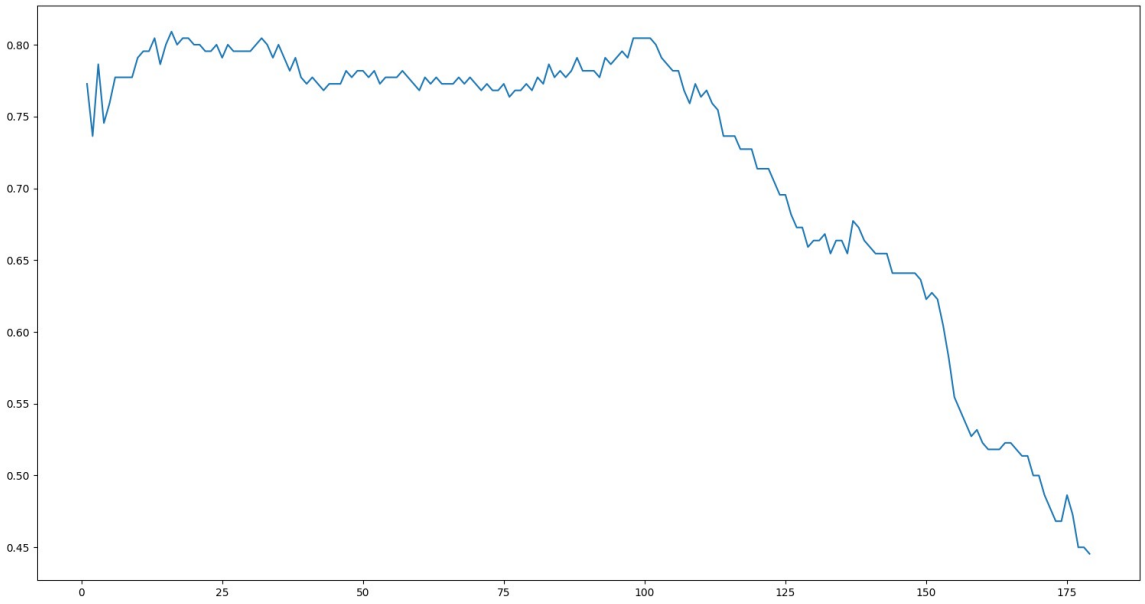
CENG499
HOMEWORK-2

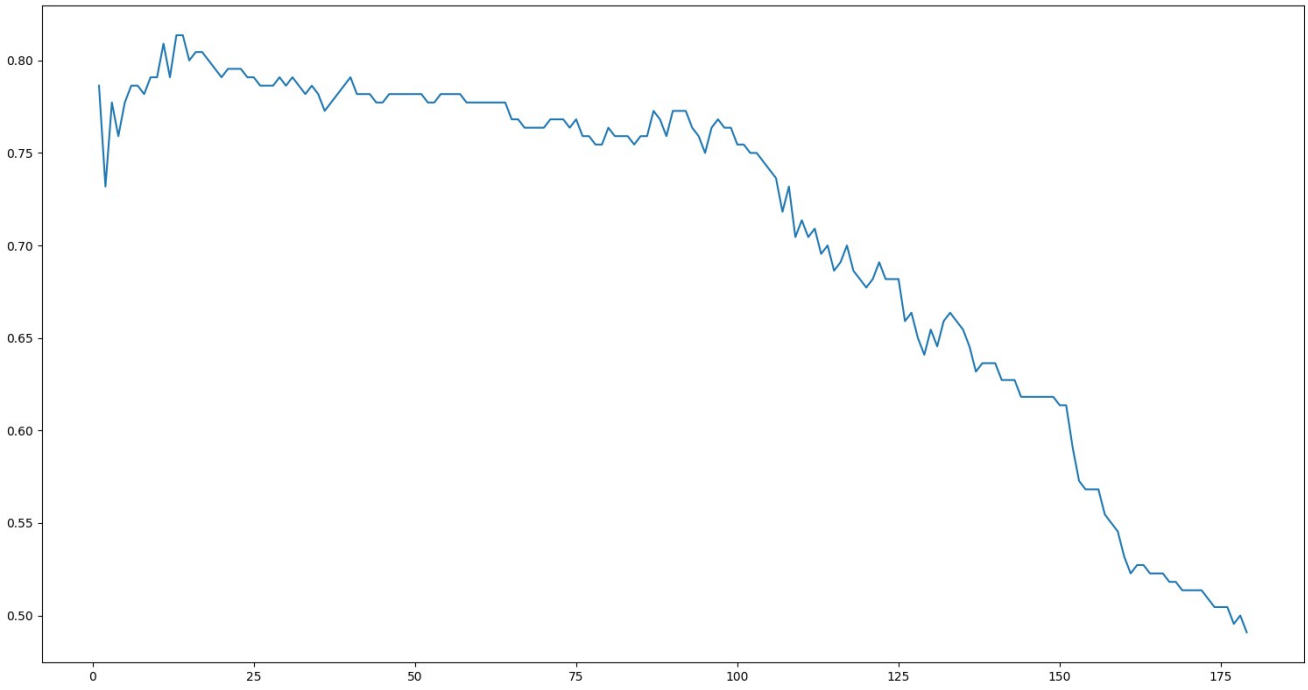
Efekan Çakmak, 2309797

K-Nearest-Neighbor

The plot below indicates that corresponding average accuracies from 10-fold cross-validation with Euclidean Distance.



The plot below shows that corresponding average accuracies from 10-fold cross-validation with Manhattan Distance.



In both cases, first I want K to have almost highest validation accuracy and secondly, I want to keep K as smaller as possible by considering computational cost and class-balancing issues.

In both plot, smallest Ks(between 1&10) are unreliable for edge instances since voting are handled small pieces of train data. Thus, we see volatility in the plot caused by over-fitting.

In both plot, Ks between 15&90 are seems safe to use since they yields similar accuracies.
I select highest accuracy for integrity and smallest K for computation ease.
For the one using Euclidean Distance, I select 16.
For the one using Manhattan Distance, I select 13.

In both plot, it can be seen that after K equals 100(almost half of the set), average validation accuracy decreases as K increases since K exceeds almost %75 of the amount of the samples. This means new instances are labeled directly depending on imbalance of the set, also this causes under-fitting.

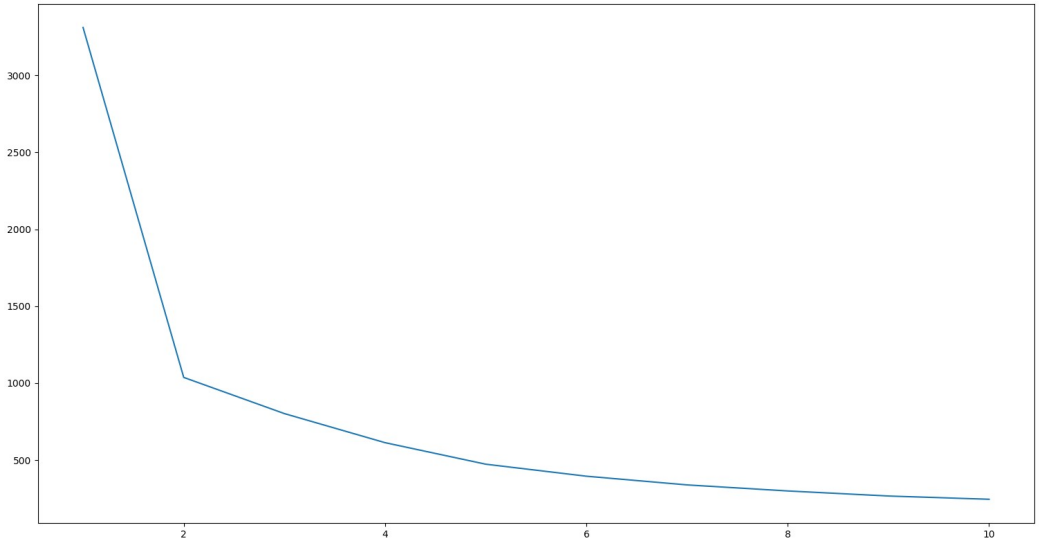
In conclusion, I got approximately 82.7% accuracy with Euclidean Distance and approximately 81.1% with Manhattan Distance on the test set.

```
efekan@efekan:~/Desktop/ML-HW2/hw2$ python3 knn_plot.py
Best K for L1: 16
Test Accuracy for L1:  0.8277777777777777
Best K for L2: 13
Test Accuracy for L2:  0.8111111111111111
```

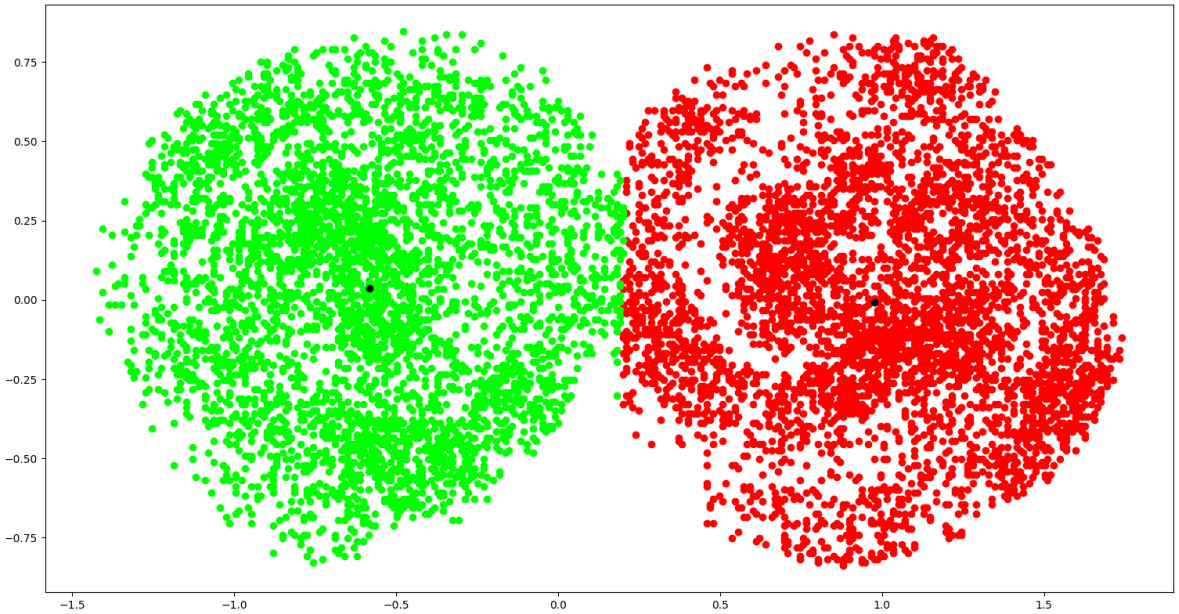
K-Means

In order to create initial clusters, I used randomly generated vectors. Then, train each data-set with different 10 randomly generated initial clusters. Lastly, I selected the one having the highest accuracy for each data set.

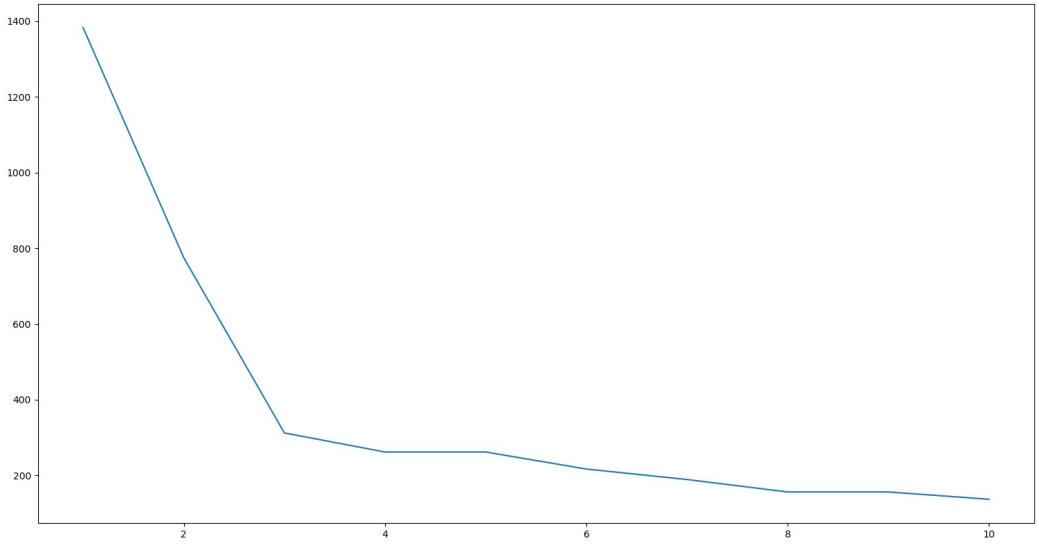
This plot below indicates values where x-axis denotes k values and y-axis denotes the objective function value on that k value in Dataset 1.



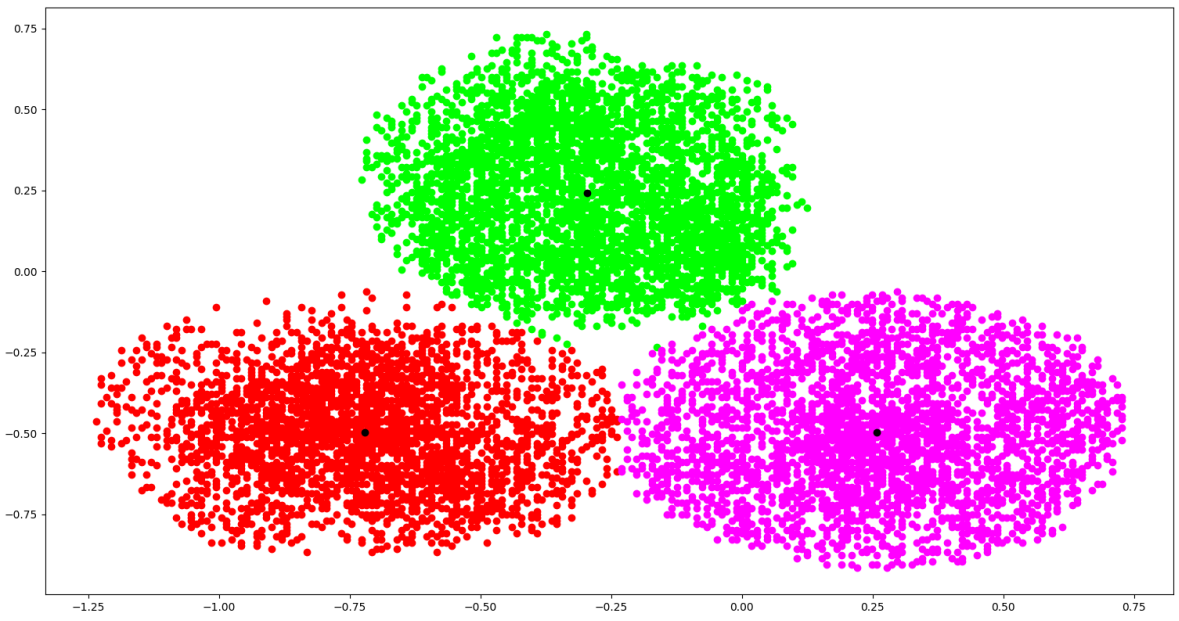
After K=2, there is no important decrease on objective function. By elbow method, I select K as 2. Then resulting clusters can be seen below. Red dots are instances owned by cluster0, Green dots are instances owned by cluster1, and black dots are indicating cluster centroids.



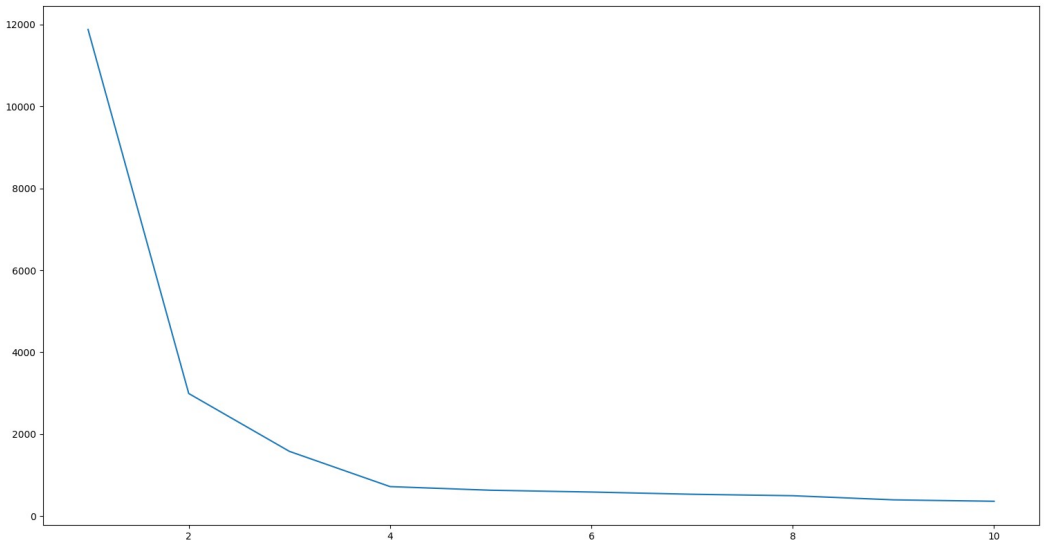
This plot below shows values where x-axis denotes k values and y-axis denotes the objective function value on that k value in Dataset 2.



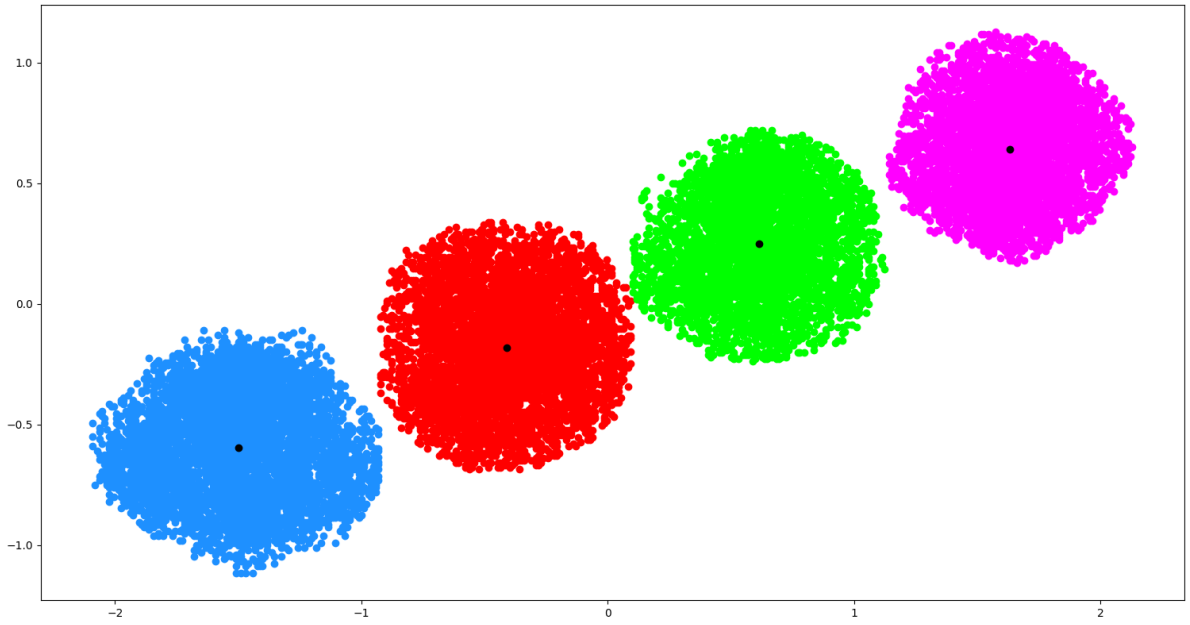
Until K=3, there is significant decrease on objective function.
After K=3, there is no significant decrease on objective function. By elbow method, I select K as 3. Then resulting clusters can be seen below. Red dots are instances owned by cluster0, Green dots are instances owned by cluster1, Pink dots are instances owned by cluster2, and black dots are indicating cluster centroids.



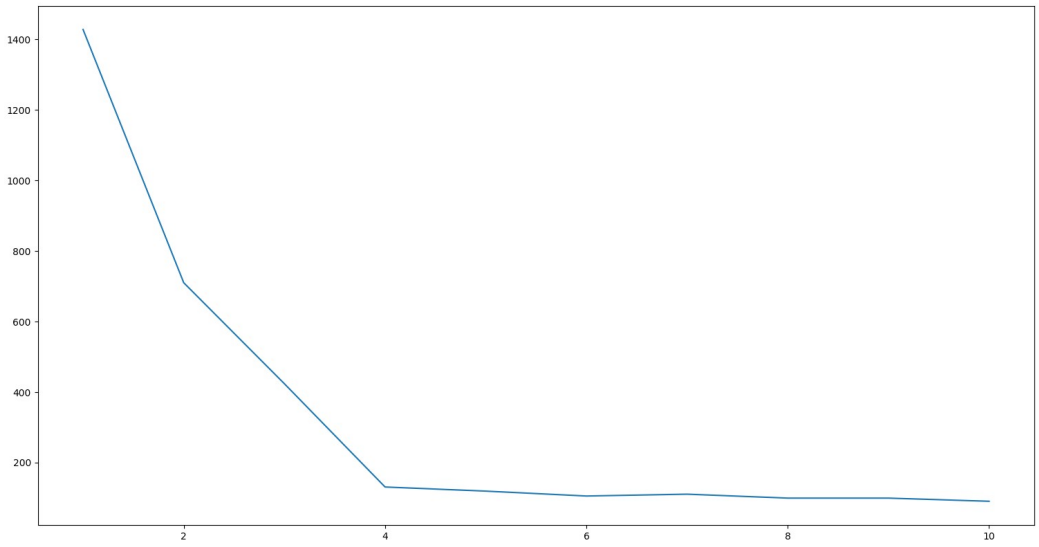
This plot below demonstrates values where x-axis denotes k values and y-axis denotes the objective function value on that k value in Dataset 3.



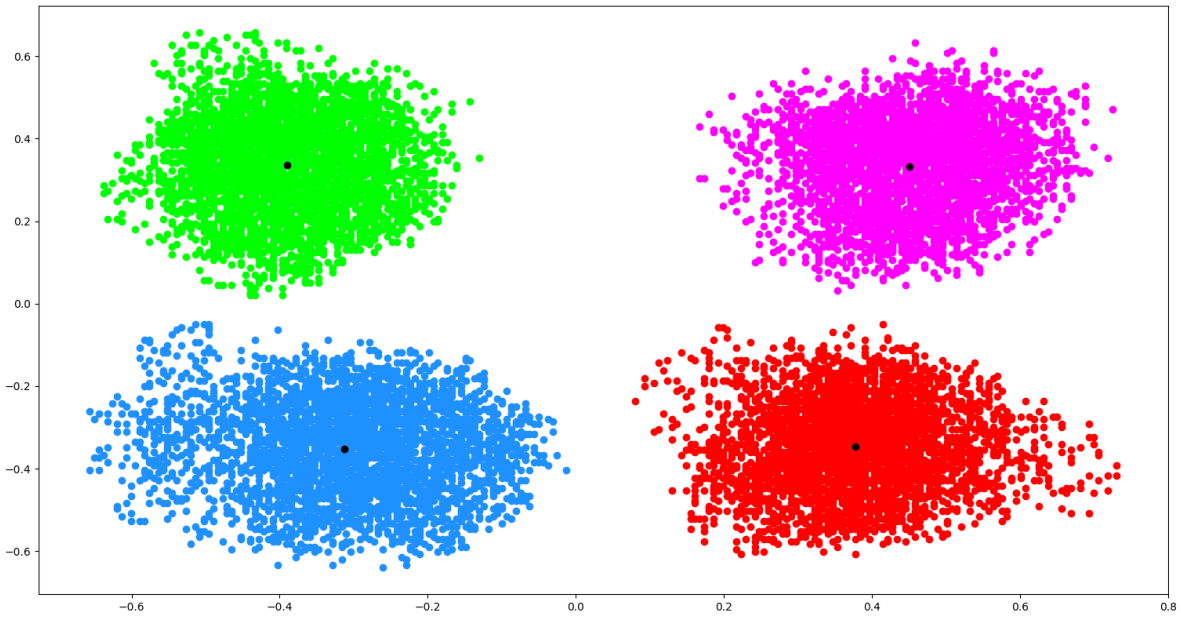
After K=4, there is no significant decrease on objective function. By elbow method, I select K as 4. Drop from 2 to 4(approximately 66%) has almost the same importance with drop from 1 to 2(approximately 75%) by considering logarithmic differences.
Then resulting clusters is shown below. Red dots are instances owned by cluster0, Green dots are instances owned by cluster1, Pink dots are instances owned by cluster2, blue dots are instances owned by cluster3, and black dots are indicating cluster centroids.



This plot below indicates values where x-axis denotes k values and y-axis denotes the objective function value on that k value in Dataset 4.



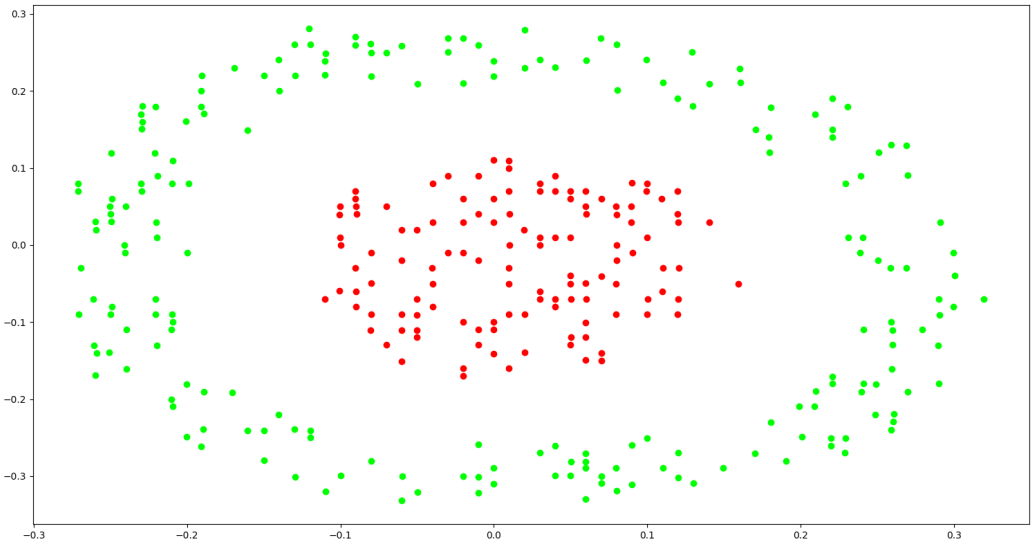
After K=4, there is no crucial decrease on objective function. By elbow method, I select K as 4. Down from 2 to 4(approximately %80) is more important than down from 1 to 2(approximately %50) by considering logarithmic differences.



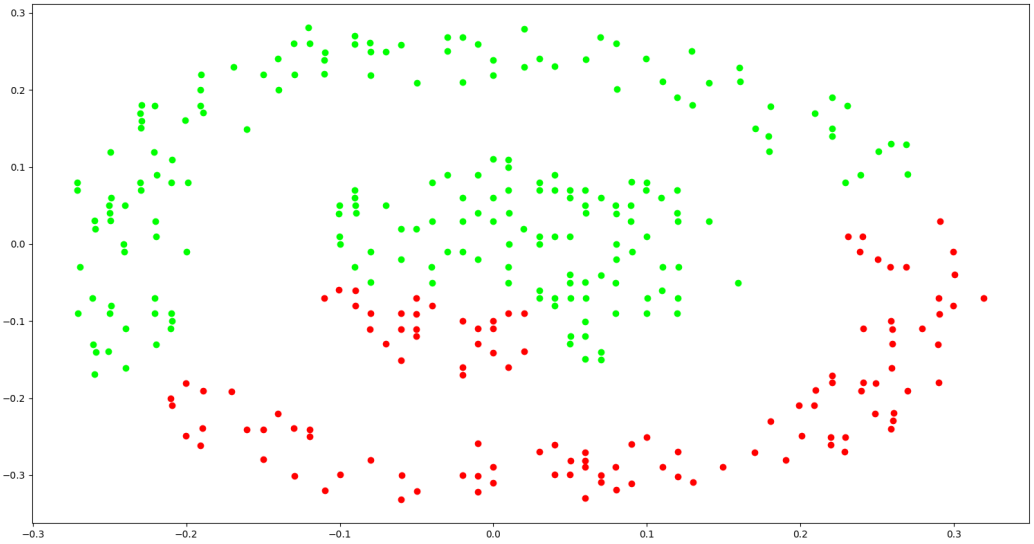
Then resulting clusters can be seen above. Red dots are instances owned by cluster0, Green dots are instances owned by cluster1, Pink dots are instances owned by cluster2, blue dots are instances owned by cluster3, and black dots are indicating cluster centroids.
To sum up, decreasing objective function is not our purpose because when amount of clusters and set size are equal, objective function becomes zero. Therefore, we need to catch rapid drops to select proper K.

Hierarchical Agglomerative Clustering

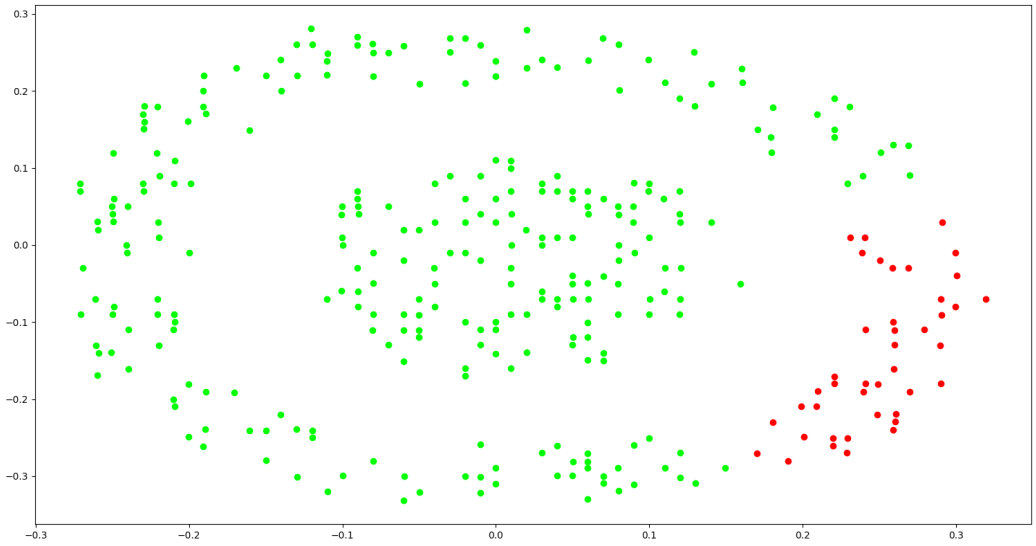
It can be seen below that the Dataset 1 is clustered by using Single-Linkage criteria.
In my opinion, Single-Linkage criteria is suitable for this dataset because we can say intuitively there are two clusters, a ring outside and a disk inside. Then, minimum distance between ring and disk is obviously higher than every nearest-neighbor pairs in their assigned clusters. This means that single-linkage will work like 1-NN and get succeed.



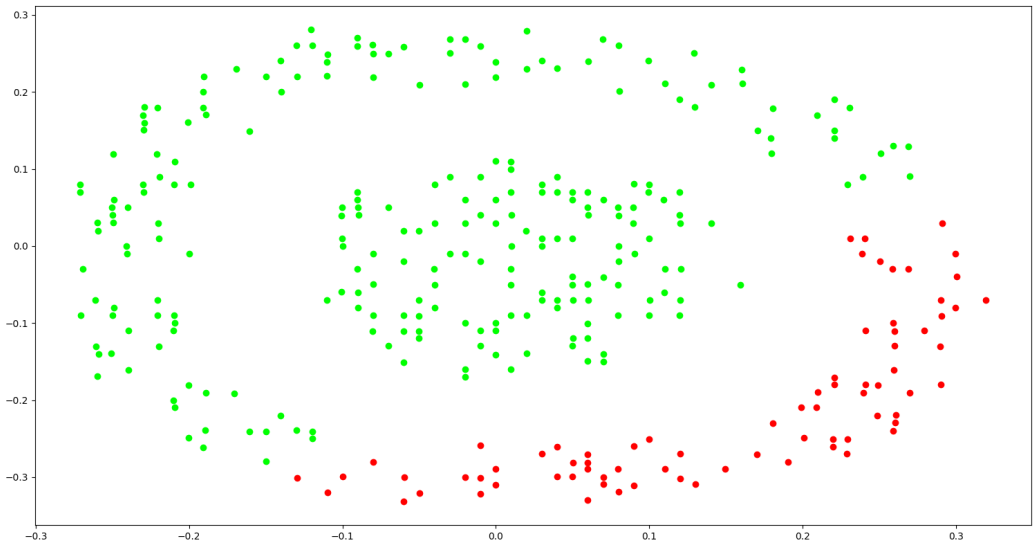
The plot below indicates that the Dataset 1 is clustered by using Complete-Linkage criteria.
I think that this criteria is not suitable for this dataset because the ring cluster has some instance pairs, whose distance between them is higher than distance to other cluster instances. Thus, selecting furthest instances between clusters has no meaning.



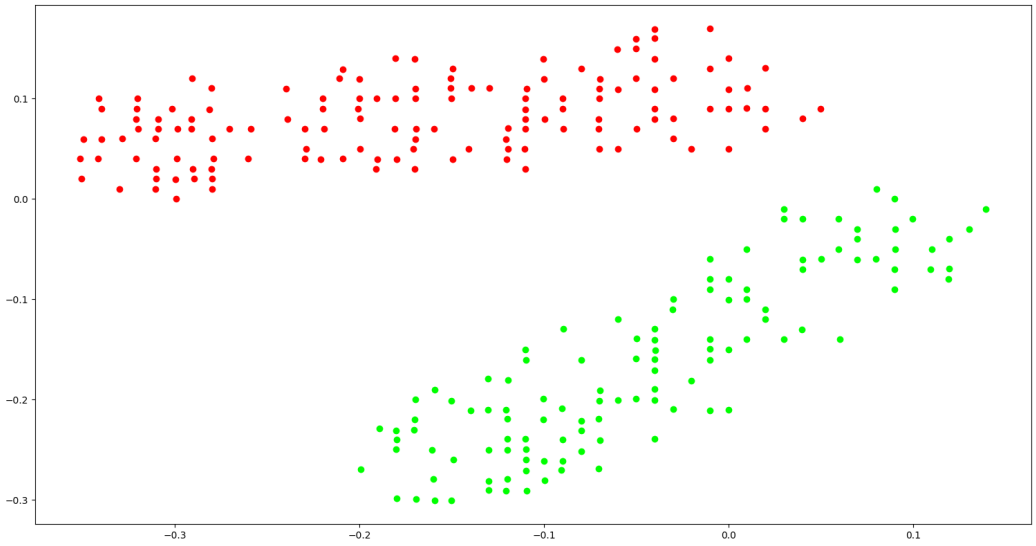
The plot below shows that the Dataset 1 is clustered by using Average-Linkage criteria.
This criteria is not suitable for this dataset since a ring cluster has some instances, distance between them higher than distance to other cluster. Then, calculation of average-linkage distance between ring cluster and an instance from center cluster will yield lower than ring cluster and an instance from ring cluster's rightmost instance. Consequently, it will fail.



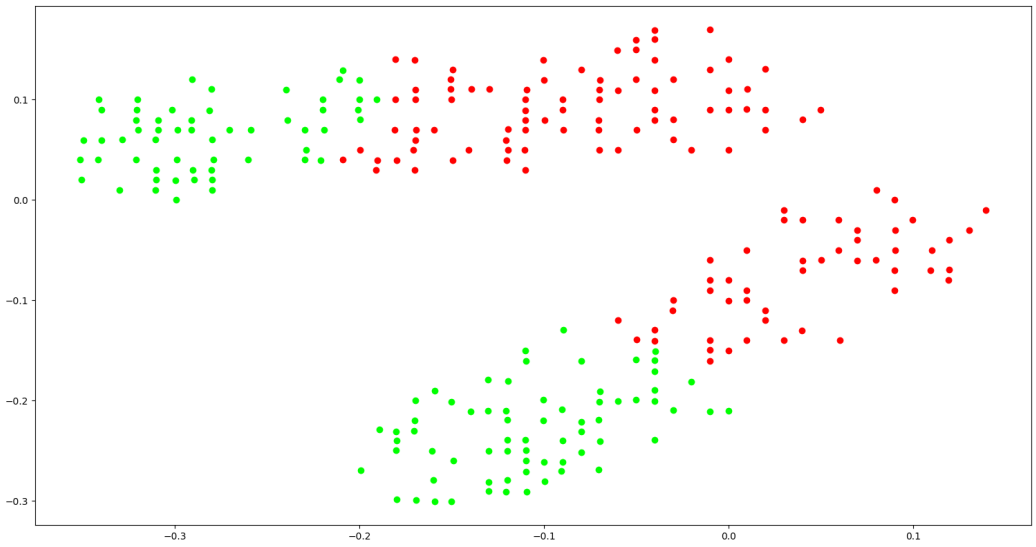
As can be shown below, the Dataset 1 is clustered by using Centroid-Linkage criteria.
I think that this criteria is not suitable for this dataset because people can say intuitively there is two cluster in this dataset, a ring outside and a disk inside. The ring's and center cluster's centroids are too close. Thus, we cannot choose two centroid that can divide this dataset successfully even we visualize the dataset.



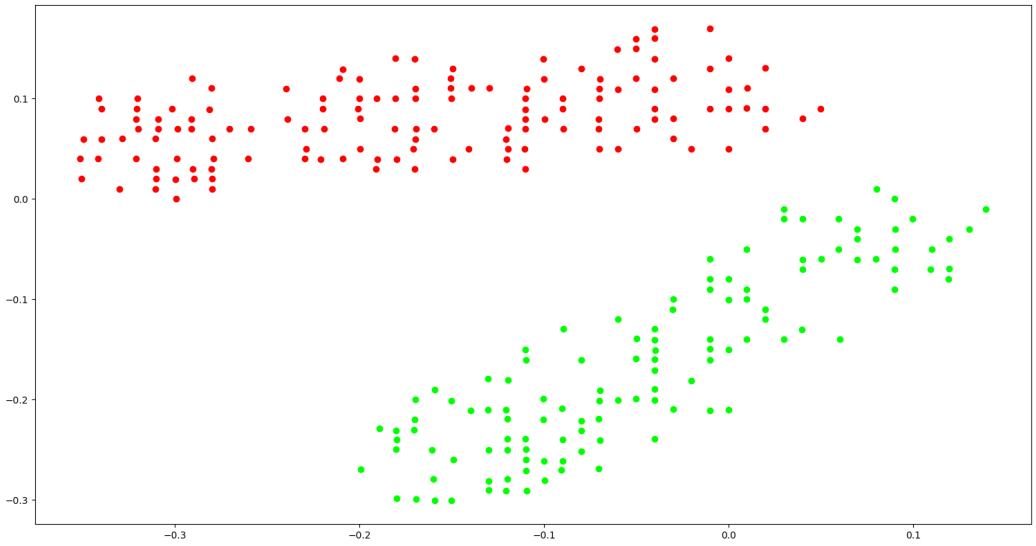
It can be seen below that the Dataset 2 is clustered by using Single-Linkage criteria.
I think this criteria is suitable for Dataset 2 from the same reason in the Dataset 1. We can say intuitively there are two clusters in Dataset2. Then, minimum distance between a cluster and other is obviously higher than every nearest-neighbor pairs in their assigned clusters. This means that single-linkage will work like 1-NN and get succeed.



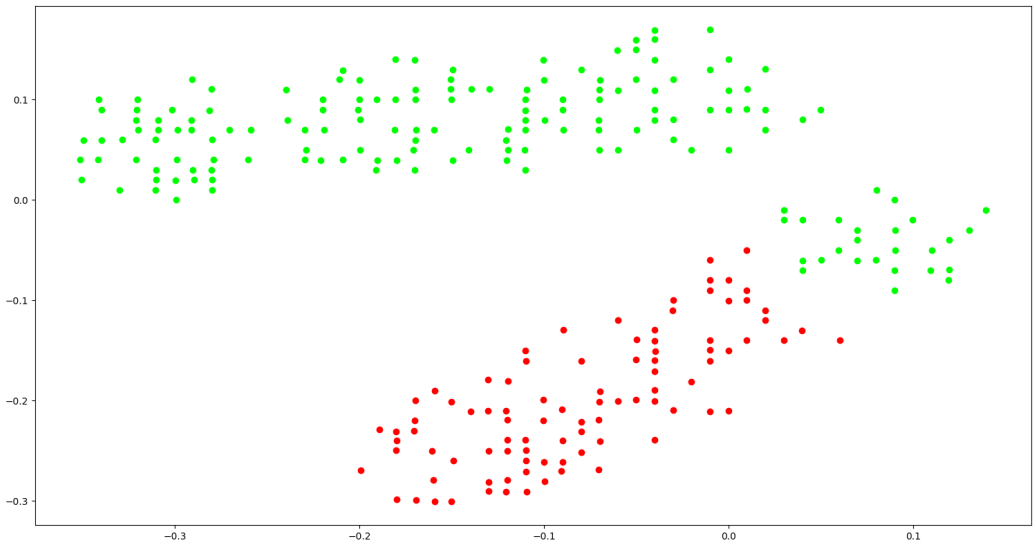
The plot below indicates that the Dataset 2 is clustered by using Complete-Linkage criteria.
It is not suitable for this dataset because true clusters are widely long and their longest instance pair is so higher than average distance between these true clusters. So that, it will see away its neighbors and see close outliers. Consequently, it fails.



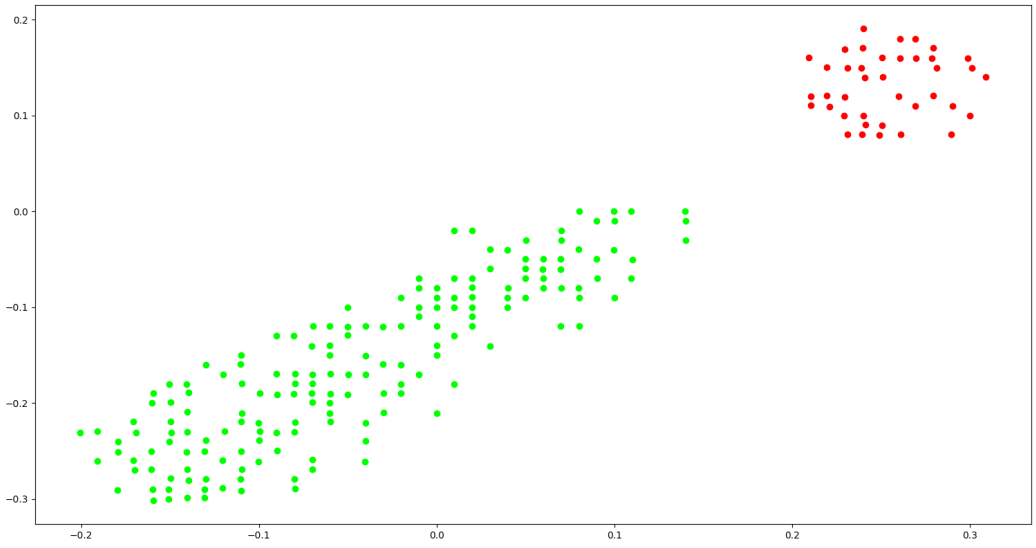
The plot below shows that the Dataset 2 is clustered by using Average-Linkage criteria
As can be seen below, this criteria suits the Dataset2. In this data set, distances between instances from different clusters vary. At right, distances are low, while high at left. So that using average of them would work properly.



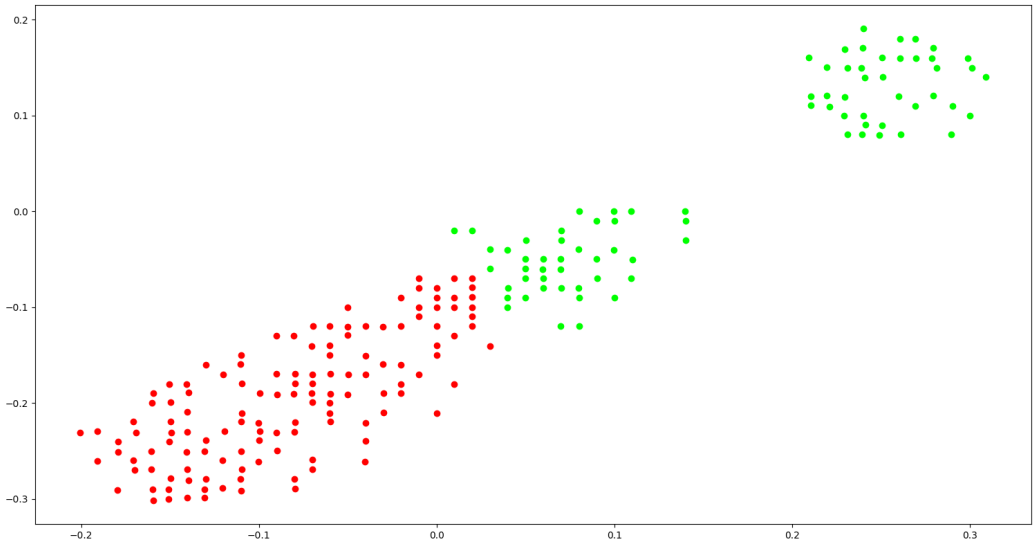
As can be shown below, the Dataset 2 is clustered by using Centroid-Linkage criteria.
I think we can not say directly this criteria is totally suitable or not for Dataset 2 because clustering seems okay except for rightmost outliers. Normally, if we plot two center by hand, they are clustered perfectly, but behavior of HAC algorithm(actually caused by first merging clusters) generates first clusters on not right areas so that it may fails.



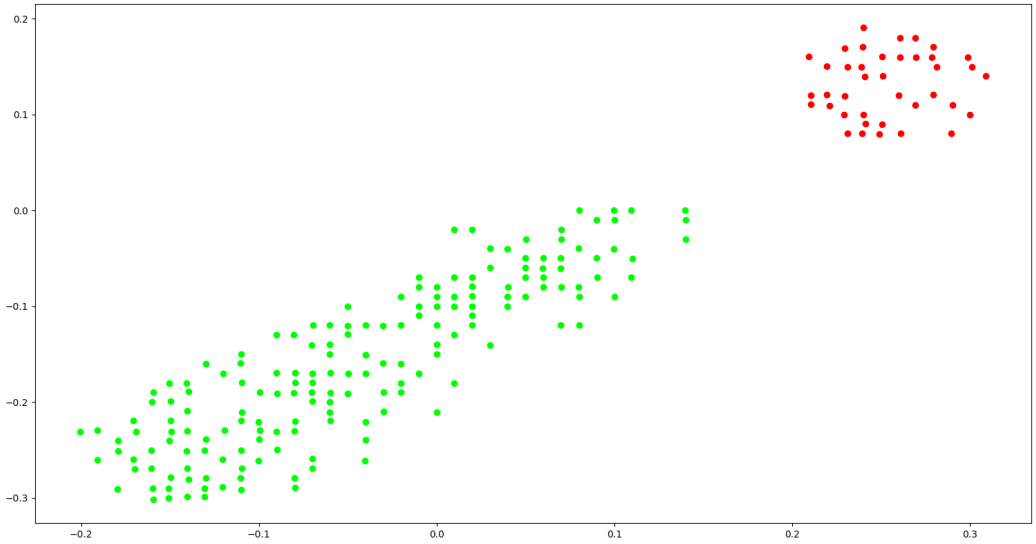
It can be seen below that the Dataset 3 is clustered by using Single-Linkage criteria.
Like other successful result of single-linkage examples, in this example, distances between instances from different clusters are explicitly higher than any nearest-neighbor pair in each clusters. Therefore, merging with closest instances yield proper result and this criteria is suitable for this dataset.



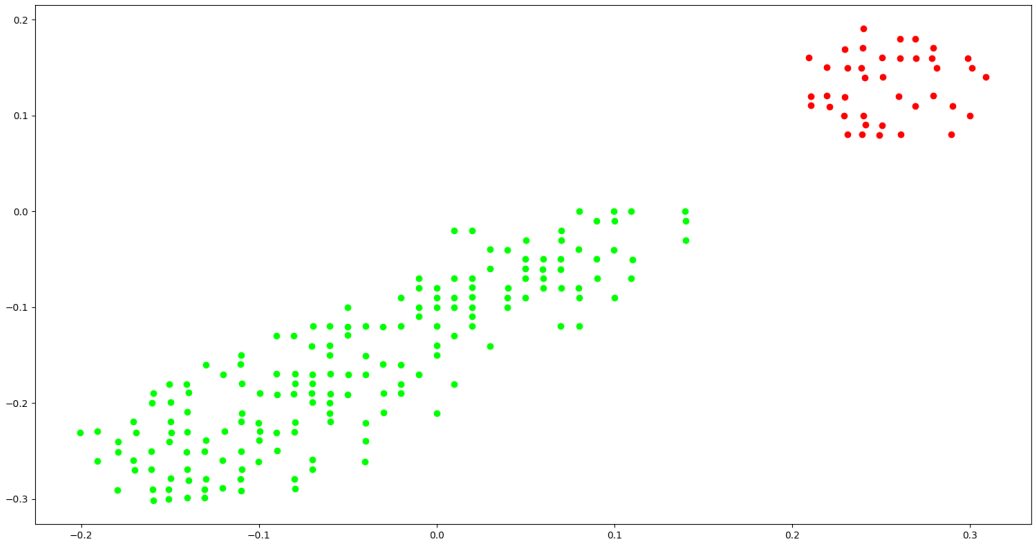
The plot below indicates that the Dataset 3 is clustered by using Complete-Linkage criteria.
It is obvious that Complete-Linkage criteria does not suit for Dataset 3. Since this algorithm care longest distances, it seems as if there are green points between two parts of the green cluster in algorithm perspective. And other problem is that the longest distance in left cluster is too high so that it yields outliers.



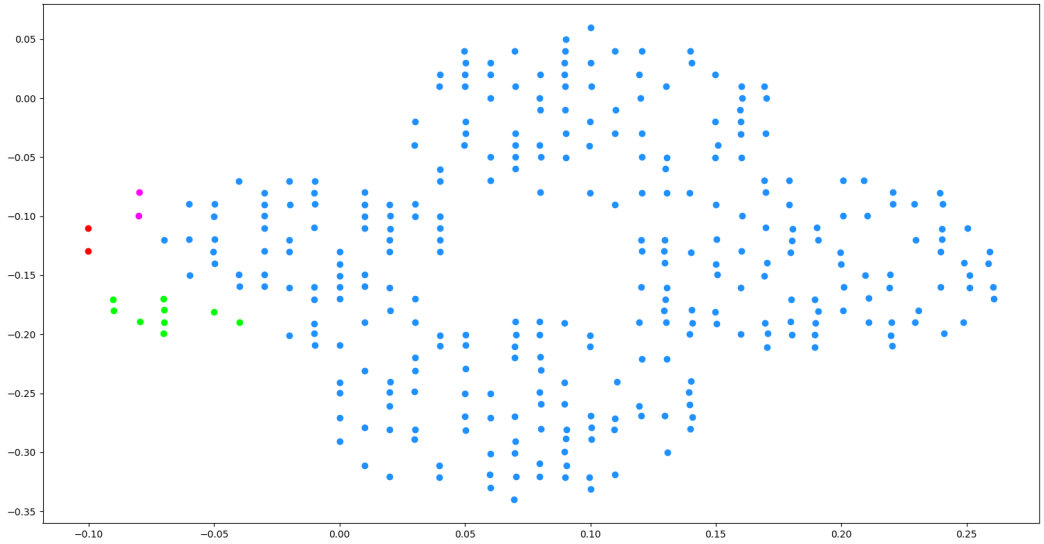
The plot below shows that the Dataset 3 is clustered by using Average-Linkage criteria.
The reason why Average-Linkage is suitable for this dataset is that there is enough distance between closest pairs of two main clusters. This distance will increase average-linkage result significantly and make the algorithm yield properly.



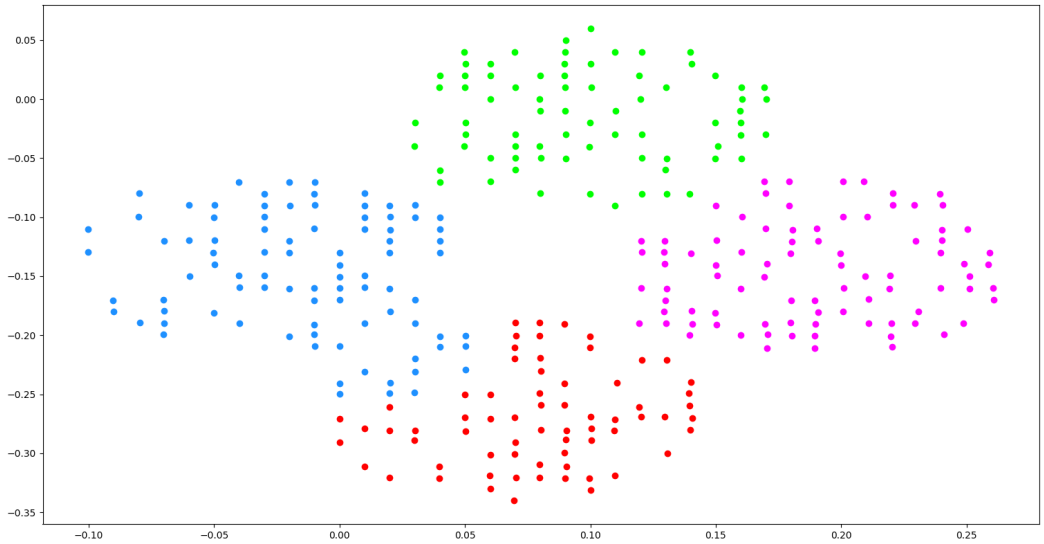
As can be shown below, the Dataset 3 is clustered by using Centroid-Linkage criteria.
Centroid-Linkage is suitable for Dataset 3. Because we have two clusters whose centroids are far from others and obviously distance of the closest pair between these clusters is sufficiently high.



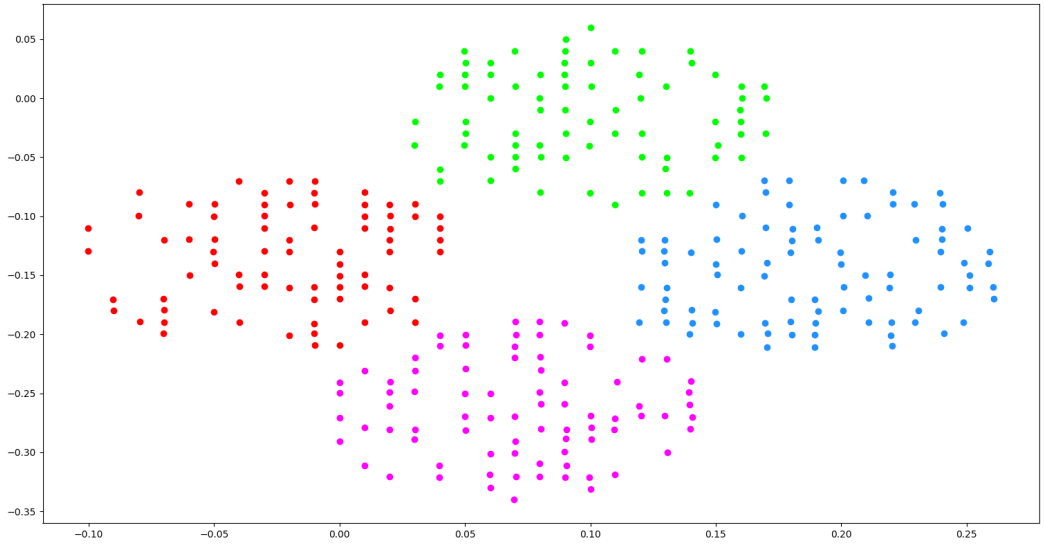
It can be seen below that the Dataset 4 is clustered by using Single-Linkage criteria.
It is not suitable for this dataset because some edge instances from different clusters are closer than a cluster's instances so that instances from different clusters' instances can be merged at the beginning.



The plot below indicates that the Dataset 4 is clustered by using Complete-Linkage criteria.
I think that this criteria is almost suitable for this dataset, except for a few outliers. Unlike Single-Linkage, this criteria avoids merging instances from two different clusters since we use the furthest distance between instances from clusters.



The plot below shows that the Dataset 4 is clustered by using Average-Linkage criteria.
It can be seen from the plot that this criteria is suitable for Dataset 4. Since differences between cluster centers are similar and clusters resemble circle, calculating average distance between an instance and a cluster will yield similar result to distance to centroids. Lastly, since centroids are far from others, this method is safe to use and yield similar results with Centroid-Linkage for this dataset.



As can be shown below, the Dataset 4 is clustered by using Centroid-Linkage criteria.
There are four clusters and their centroids are obviously far from others and average distance pairs between clusters are close. This conditions help us prevent from merging with contrary clusters. Consequently, this criteria is suitable for Dataset 4.