TUGAS 2: K - MEANS

Name = Puruso Muhammad Hanunggul

Class = IF-39-INT

NIM = 1301153680

Problem Description (Case Study)

In this case, we were given 688 data as trainset which each data consist 2 attributes. Because this is unsupervised learning, we have to observe what the data is, and we must create a program that can classify the data itself without knowledge about the data class before. It means, the machine should create a cluster to specify which class for each data by using K – Means Method.

Method and Design

This Problem is solved by using K-Means, one of Unsupervised Learning Method in Machine Learning. There are few steps of algorithm to solve this problem. First, we have to determine the value of K. K, is the value that determine how many cluster that will be consist. In this case, I initialize the value of K is 4. Then, select random data from trainset as much as value of K (Four), and assume those random value are the centroid for each cluster.

After we get the initial centroid for each cluster, do the iteration that check the Euclidean distance for each data in trainset to each centroid. Check which centroid has the smallest distance with the data, then, put the data into a member/ class of the centroid which has the smallest distance.

When all data in trainset has been checked, update the centroid for each cluster based on its member. Then, do this iteration until the whole centroid doesn't change anymore. Then, it will give us the final result of centroid for each clusters. Use those centroid to specify class for test set data.

I define 4 clusters into 4 codes, which are; X021, G750, F10K, and R309. In this case, even though I was initialize 4 clusters. By running the program repeatedly, mostly the cluster only consist 2 up to 3, because there is a centroid that is too far for every data in test set. Even sometimes, the program will gives only 1 cluster.

Final step, after check all data for each centroid, write the result in txt file, which has been loaded by the program to save the result.

Since in this case we use unsupervised learning with K-Means, there's no benchmark which K has better performance. Hence, we can see that even I set the value of K is 4, mostly, the test set only fill up to 3 clusters instead of all clusters.

Screenshot

```
hasil random: 9
21.7    21.9

hasil random: 469
17.95   11.9

hasil random: 79
11.05   23.9

hasil random: 661
9   12.7

x: 0
kx 21.7     ky 21.9

x: 1
kx 17.95    ky 11.9

x: 2
kx 11.05    ky 23.9

x: 3
kx 9    ky 12.7
```

Here is the example to get random value. In this case, the program will generate 4 random vectors since I initialize the value of K is 4.

Split the value of data into attribute X and Y for each chosen random data

```
new centC [9.92, 22.461]
total: 688
iterasi: 16
total: 136 99 165 288
old centA [30.656, 22.351]
old centB [32.583, 8.774]
old centC [9.92, 22.461]
new centA [30.656, 22.351]
new centB [32.67, 8.792]
new centC [9.897, 22.401]
total: 688
iterasi: 17
total: 136 99 165 288
old centA [30.656, 22.351]
old centB [32.67, 8.792]
old centC [9.897, 22.401]
new centA [30.656, 22.351]
new centB [32.67, 8.792]
new centC [9.897, 22.401]
total: 688
```

Do the iteration until the centroid doesn't change anymore. Each time program runs will give a different number of iteration because the random value of each centroids are also different.

```
---Final---
 centroid A: [30.656, 22.351] centroid B: [32.67, 8.792] centrod C: [9.897, 22.401] centrod D: [15.362, 7.114]
member A: 6
 member B: 0
 member C 56
 member D 38
```

And here an example of final result. There are 4 clusters, but when it implemented to test set, test set only needs 3 clusters.

Member A consist 6 data, member B consist 0 data, member C consist 56 data, member D consist 38 data.

It is not absolute value, each time program runs sometimes will give a different cluster for each data but the grouping is similar. Furthermore, you can check the source code in attached file