MACHINE LEARNING AND DATA MINING 2

LABWORK 2: Clustering

GROUP MEMBERS:

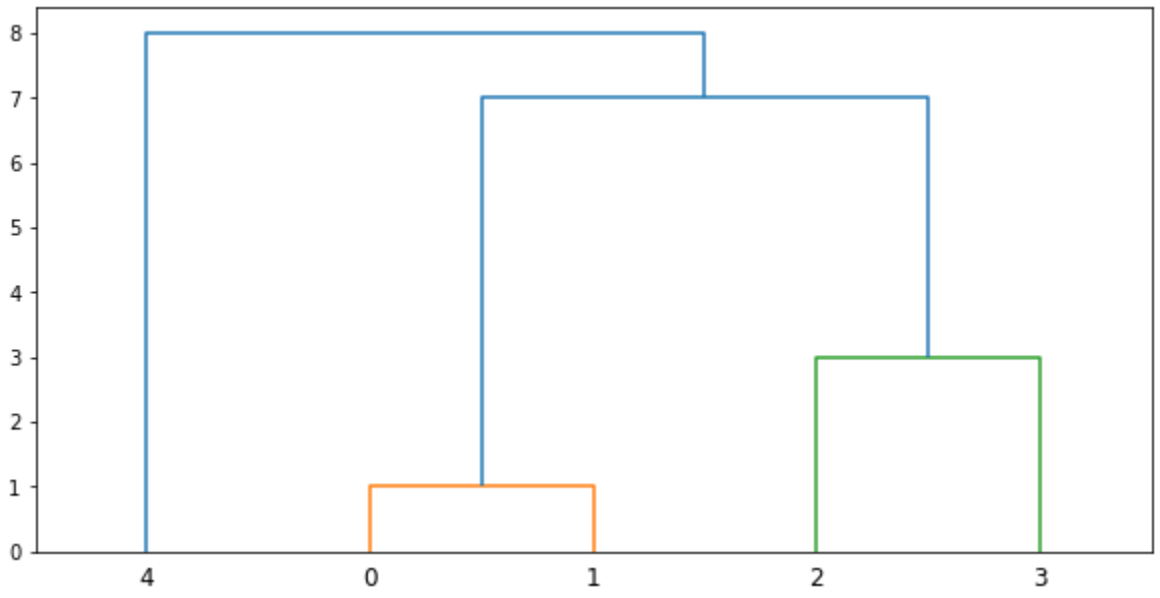
Đào Hải Long BA9-041

Đặng Thái Sơn BA9-053

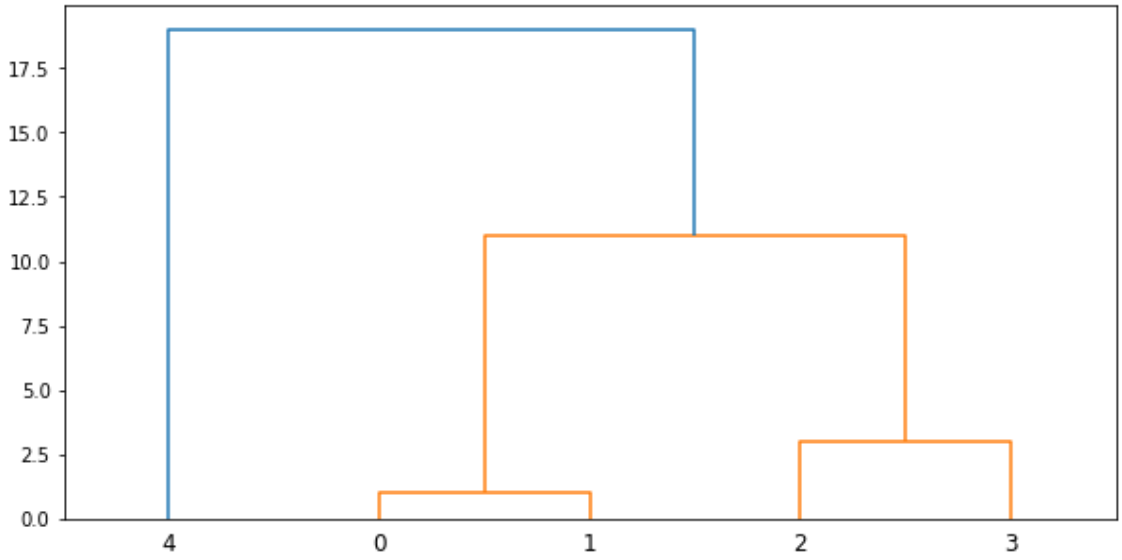
A/ AHC

I/ Apply the AHC clustering for the X dataset and draw the clustering result using function dendrogram() in Python

1. Using Single Linkage



2. Using Complete Linkage



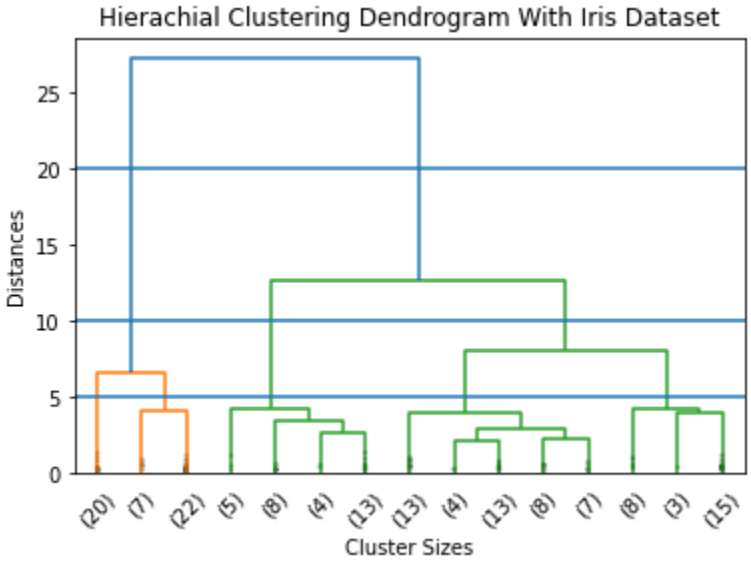
II/ Apply the AHC clustering on two more datasets from UCI

1. Iris dataset

a) Study of data features

Iris dataset has 4 dimensions. For this dataset, all features are continuous and quantitative because they are real numbers showing length and width in centimeters.

b) The dendrogram



We use the bottom-up algorithm and this is the binary cluster tree created by the linkage function when viewed graphically. In the figure, the numbers along the horizontal axis represent the indices of the objects in the original dataset. The links between objects are represented as upside-down U-shaped lines. The height of the U indicates the distance between the objects.

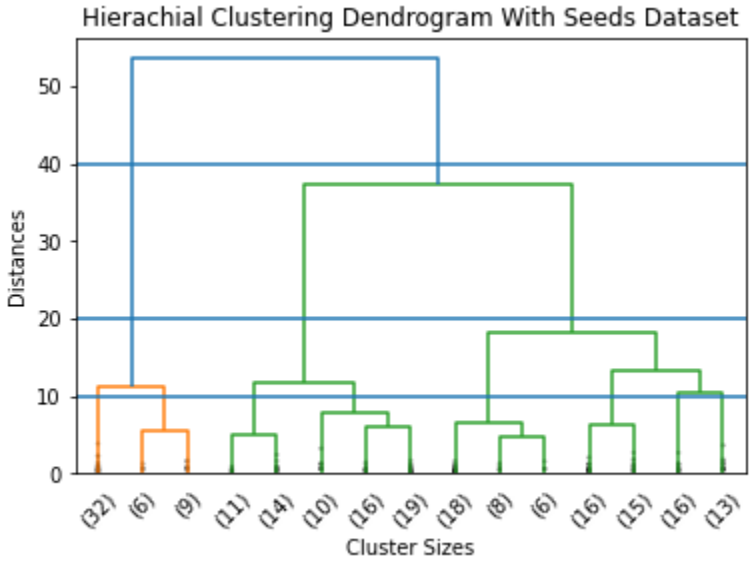
In addition, the result does not scale well, especially the ones at the bottom due to the huge number of data.

2. Seeds dataset

a) Study of data features

Seeds dataset has 7 dimensions. For this dataset, all features are continuous and quantitative because they are real numbers.

b) The dendrogram



As explained above, we use the bottom-up algorithm and this is the binary cluster tree created by the linkage function when viewed graphically. In the figure, the numbers along the horizontal axis represent the indices of the objects in the original dataset. The links between objects are represented as upside-down U-shaped lines. The height of the U indicates the distance between the objects.

Moreover, same as the result earlier, this diagram does not scale well, especially the ones at the bottom due to the huge number of data.

III/ Advantages and drawbacks of AHC

1. Advantages

- It does not require any input parameters in advance.
- Simple visualization and easy comparison in similarity between objects.

2. Drawbacks

- The complexity is high: $O(n^2)$ or $O(n^3)$.
- The result is sensible to noisy.
- It does not scale well. We have to check all the number of data before splitting.

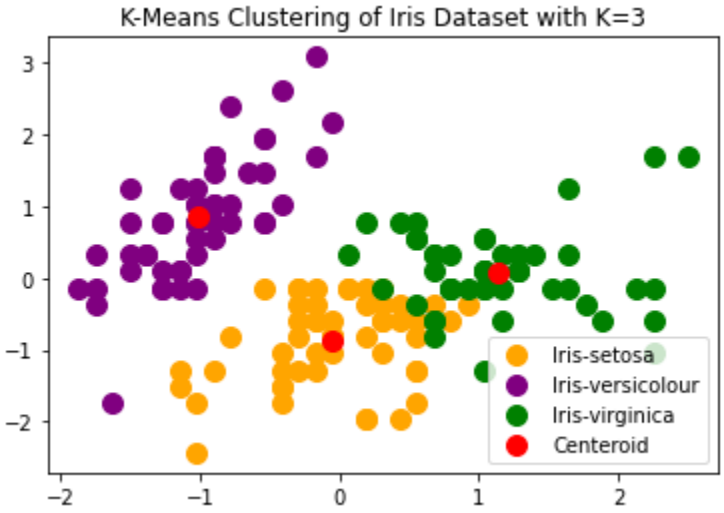
B/ K-means

I/ Iris dataset

1. K-means experimental protocol where K = 3

1. Given that 3 centroids are known, an object x_i (i = 1, 2, 3) is assigned to the nearest centroid to minimize the quantization error.
2. Once all the input objects have been assigned, for each cluster C_k (k = 1, 2, 3), we estimate the new centroid.
3. These steps are repeated until the centroids are convergent.

Here is the result we get when we set K = 3:

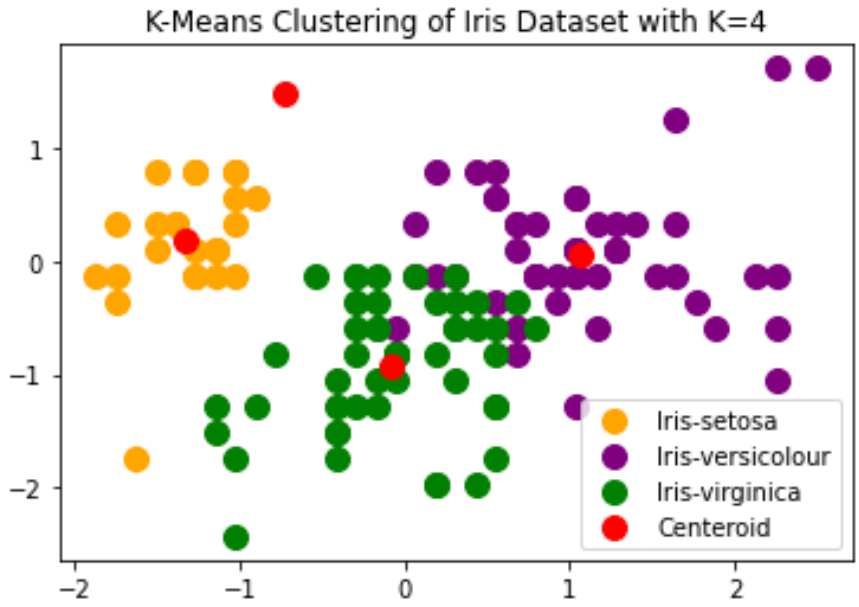


2. Centroid initialization of the K-means

In this Iris dataset, we see that the data is already divided into three classes: Setosa, Versicolour, and Virginica, so we decided to implement the K-means using 3 centroids according to the number of classes. In our opinion, this is the appropriate number for centroid initialization, which can make it easy for clustering.

3. Analyze, compare the results with different K and calculate the clustering quality

Here, we change the value of K to 4 and get the following result:

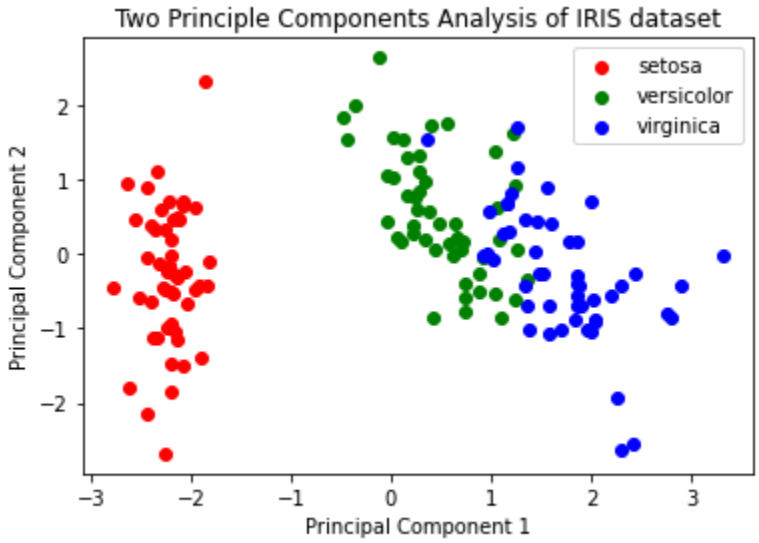


Visually, there is not much improvement in clustering when we only increase the centroid number of 1 unit although the quality rises statistically. To be more specific, to calculate the clustering quality, we used the accuracy evaluation to reflect the proportion of objects that were correctly assigned:

- The accuracy when **K = 3** is: **0.96**
- The accuracy when **K = 4** is: **1.513333**

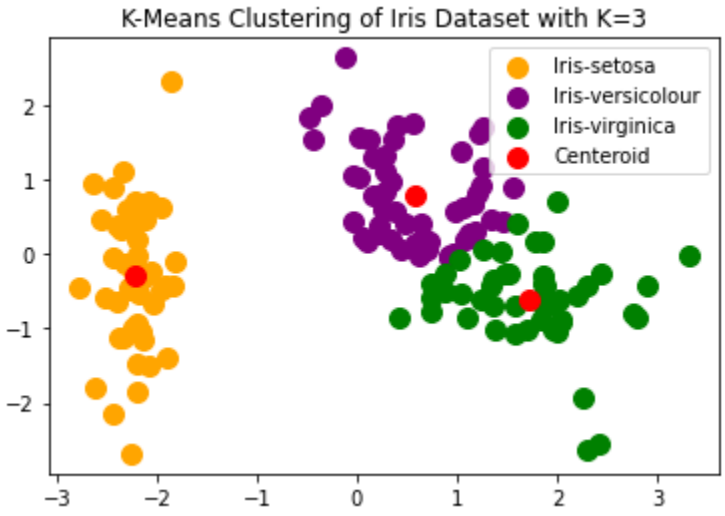
4. Use PCA to visualize the data distribution in 2D

We use 2 components for PCA and here is the result:



5. Apply K-means after PCA

After using PCA, we again apply K-means with K = 3 and get the following result:



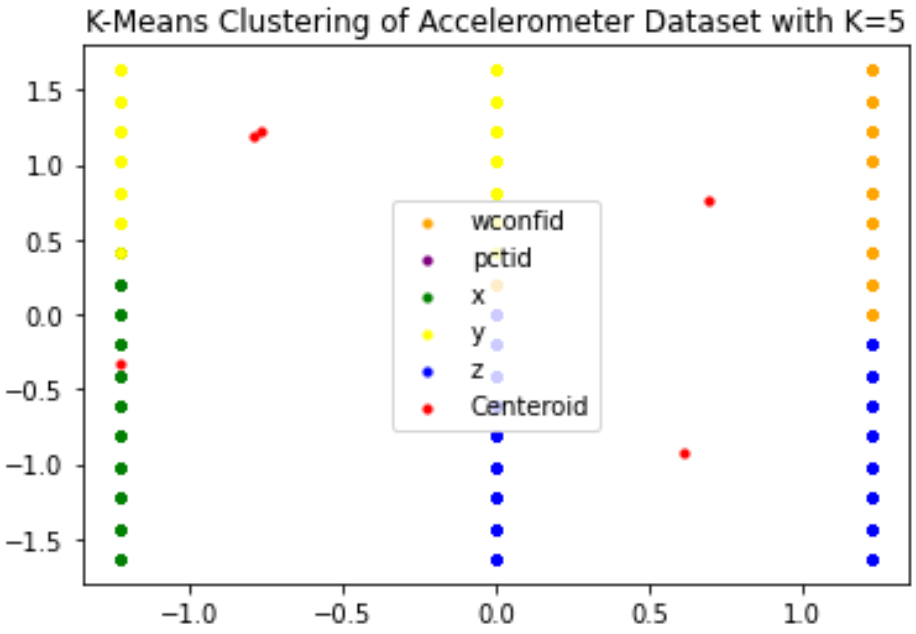
Visually, the result is still clearly clustered. In terms of accuracy, after applying PCA then K-means, we can get the result of **0.98**, which is just a little bit higher than the result before dimensionality reduction.

II/ Accelerometer dataset

1. K-means experimental protocol where K = 5

1. Given that 5 centroids are known, an object x_i (i = 1,…, 5) is assigned to the nearest centroid to minimize the quantization error.
2. Once all the input objects have been assigned, for each cluster C_k (k = 1, …, 5), we estimate the new centroid.
3. These steps are repeated until the centroids are convergent.

Here is the result we get when we set K = 5:

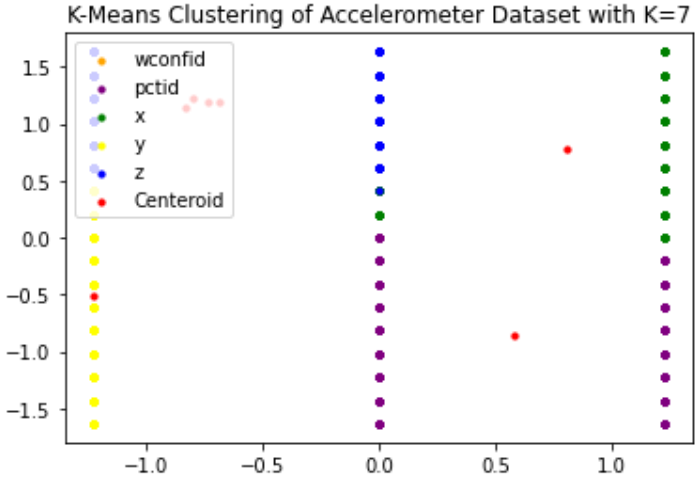


2. Centroid initialization of the K-means

In this Accelerometer dataset, we see that the data is not divided into classes like the Iris dataset, so we decided to implement the K-means using 5 centroids based on the number of attributes. In our opinion, this is maybe the appropriate number for centroid initialization, which can make it easy for clustering.

3. Analyze, compare the results with different K and calculate the clustering quality

Here, we change the value of K to 7 and get the following result:

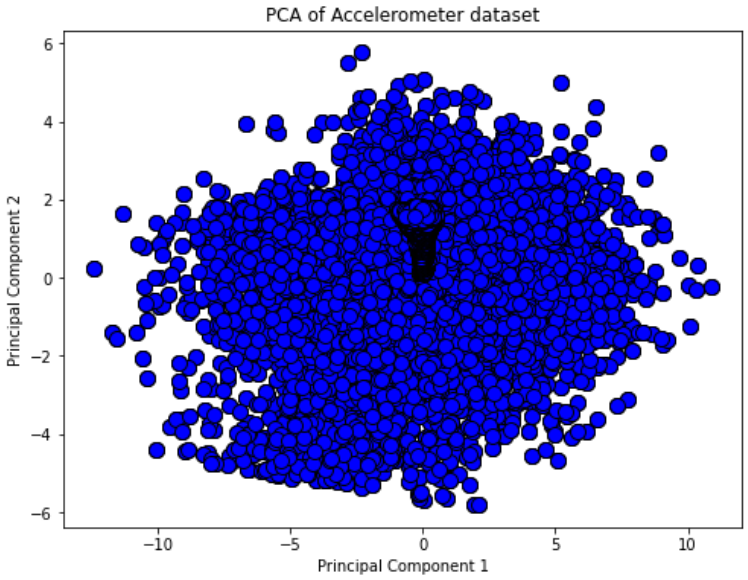


Visually, it is hard to see the difference between the two results. However, the improvement when increasing the centroid number of 2 units is showed specifically in terms of the clustering quality. To be more specific, to calculate the clustering quality, we used the accuracy evaluation to reflect the proportion of objects that were correctly assigned:

- The accuracy when **K = 5** is: **2.00916**
- The accuracy when **K = 7** is: **2.10729**

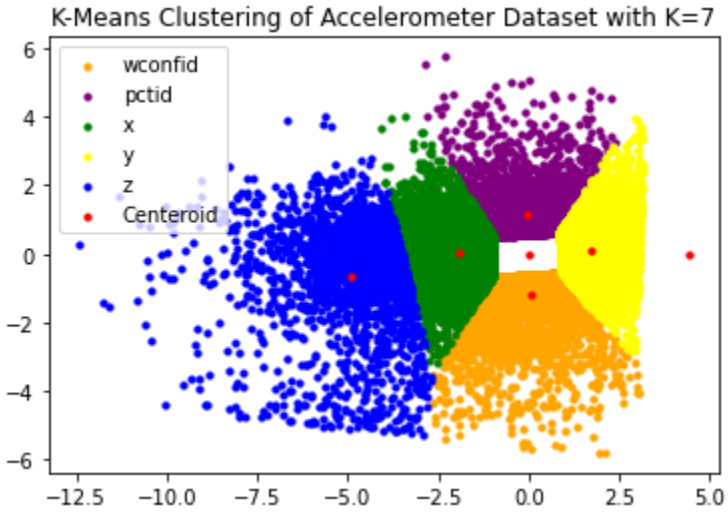
4. Use PCA to visualize the data distribution in 2D

We use 2 components for PCA and here is the result:



5. Apply K-means after PCA

After using PCA, we again apply K-means with K = 7 and get the following result:



Visually, it is clear to see that with PCA before apply K-means, the clustering result is greatly improved and we can easily see several centroids in the convergent state.

In terms of accuracy, after applying PCA then K-means, we can get the result of **2.288**, which is just a little bit higher than the result before dimensionality reduction.

C/ Source code

Here is the Github repository to the source code of our labwork if you want to check:
https://github.com/DaoHaiLong/Machine-Learning-and-Data-Mining-II