Classification I

Report

Machine learning and Data mining II

Nguyễn Mạnh Hưng - 22BI13183

Nguyễn Trọng Minh - 22BI13304

**TABLE OF CONTENT**

1. **K-nearest neighbor classification**
2. Perform K-NN
3. Normalization
4. Apply PCA
5. Validating
6. **Support Vector Machine**
7. Analyze the dataset
8. Set up the SVM
9. SVM’s performance
10. Multi-class datasets
11. **K-nearest neighbor classification**
12. **Perform K-NN**

In this lab work, we chose 2 data sets both are binary classified. One contains data about 100 mushroom individuals and classified into edible or not. The other one is about



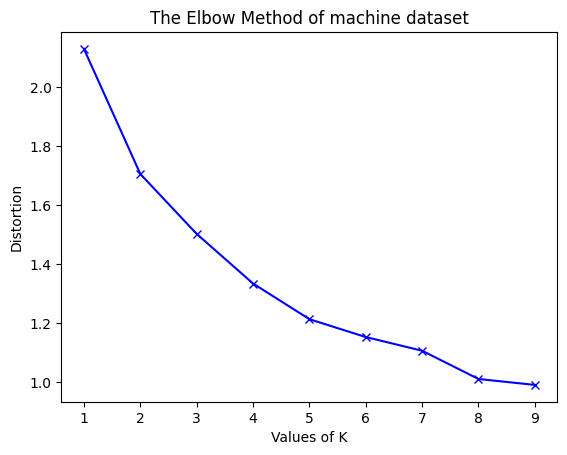
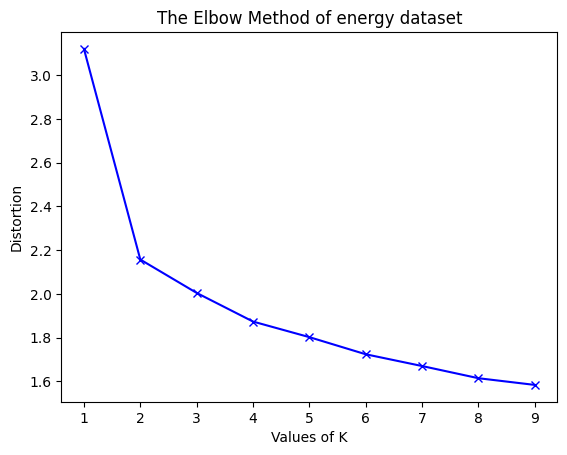
* 1. Data about CPUs performance



* 1. Data about energy efficiency

After a quick look into the data set, we can start applying K-means clustering. Because K-means clustering measures the Euclidian distance between a data point and centroids, we need to standardize data into an appropriate scale. We simply do this by divide each category by their standard deviation. And for later visualization using PCA, I also want to center the data to the origin.

The next step is to determine how many clusters we want to find from this data set. Normally, we can set the number of clusters to any positive integer, but there is a method to find which number give the most efficient. This method is called “Elbow method”. In this method, we continuously iterate through every k from 1 to n, which is set to be 10 in this article. For each iteration, we calculate the within-cluster sum of squares, or cluster’s variance. Then, we plot a graph and surprisingly, this graph looks like an elbow.



As we can see from the graph, we need to pick k when the line starts to look like a straight line, in both cases, I would choose number 4.

Now the data is ready to be separated into 4 clusters with K-means algorithm. It starts by initialize the first 4 centroids, and assigns every data point into its nearest centroid. After that, it stops and calculate a new centroid which is the mean value of every data point inside a cluster. And again, it comes back to step 2, assign data point into a cluster. This loop occurs for several times to find the best cluster centroid with the least data variance.

At the end, we get a clustered data set with labels for each data point.

1. **Centroid initialization**

In the above session, we talked about an overall protocol of K-means clustering, I want to go deeper into the centroid initialization in this part of the report.

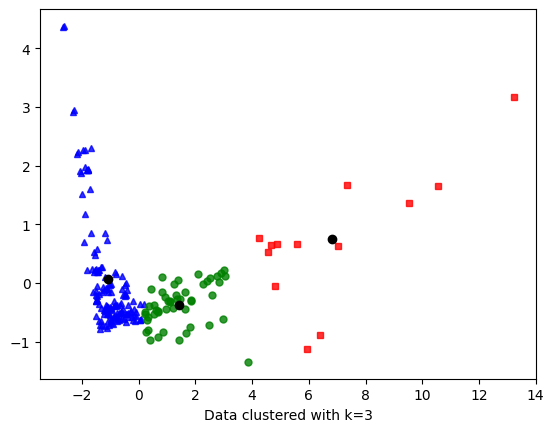
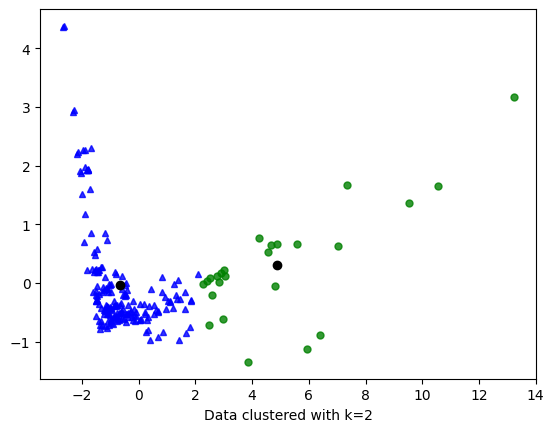
Normally, the first centroids of the algorithm are generated randomly inside data set. But in this experiment, I applied a method to determine those centroids, called “greedy k-means++” published by Arthur and Vassilvitskii in 2007. The greedy k-means++ algorithm samples k initial centers by adaptive sampling, where in each step, 𝑙 possible centroids are chosen, and then among these 𝑙 centers, the one set that decreases the k-mean cost the most is chosen to be initial centroids.

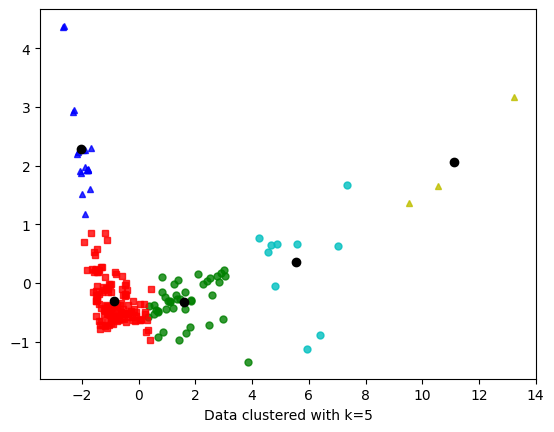
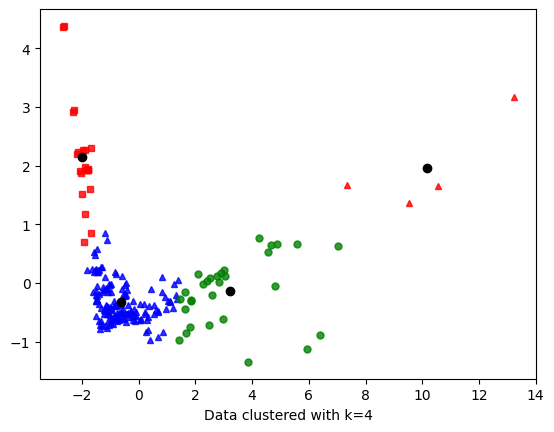
This technique is believed to speed up convergence that provides an

O(log k)-approximation in expectation.

1. **Analyze the result**

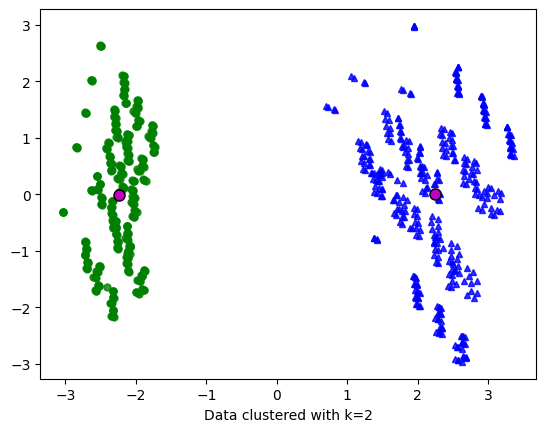
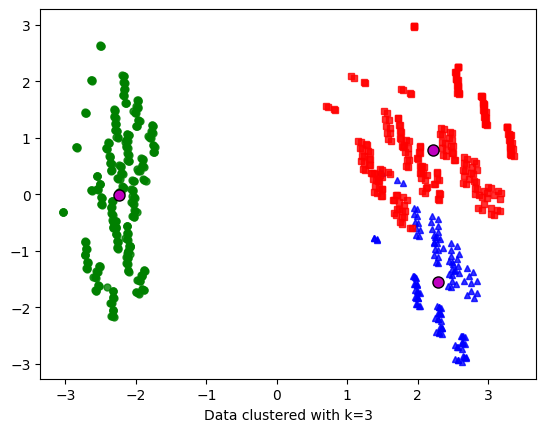
For the CPUs performance dataset, here are the results with different values of k:

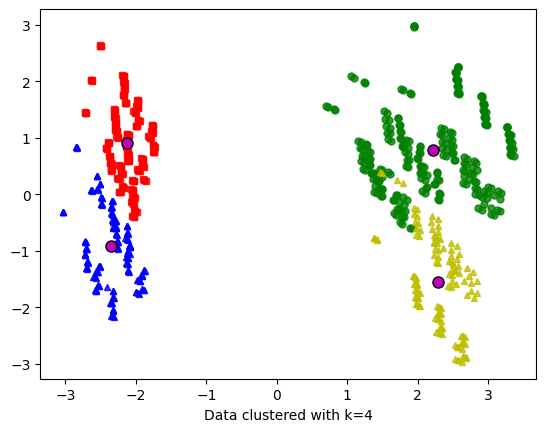
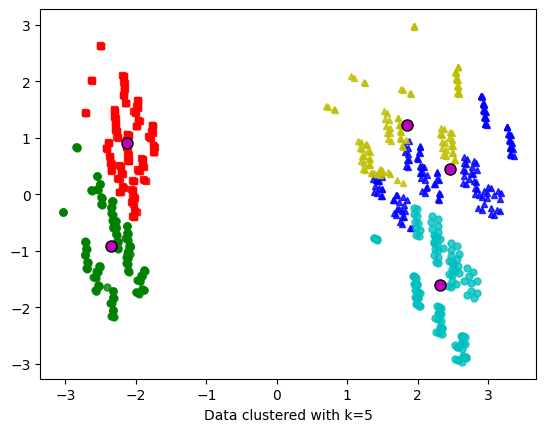




As we can see in case k = 4 and 5, there is a cluster with tiny amount of data points, 4 and 3. These 2 clusters should be merged into bigger cluster for better performance.

For the energy dataset here are the results with different values of k:

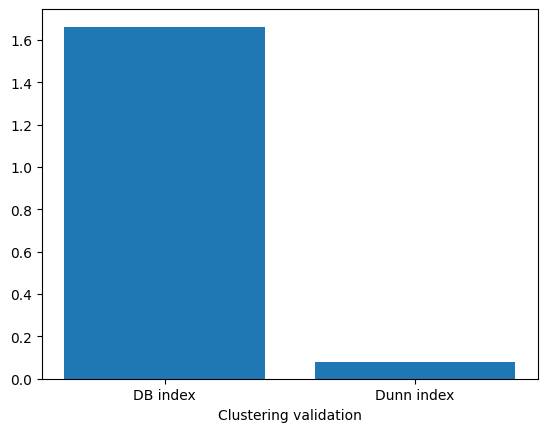
 

The plotted data show that k=2 is the best number of clusters for this dataset. We can also see some data points might be clustered into wrong cluster, but this phenomenal is just data error due to PCA.

1. **Clustering quality**

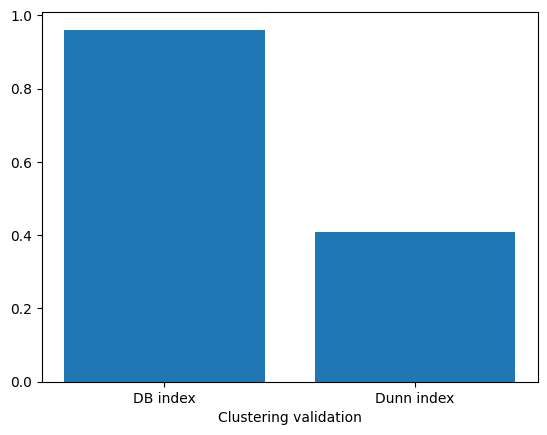
To validate the quality of the cluster, we apply 2 internal validation methods, Davies-Bouldin index and Dunn index.

Here is the validation result for energy dataset with k=4:



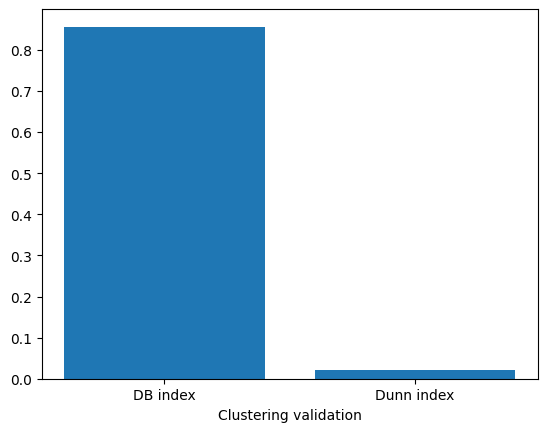
We can see that DB index is quite high, that means the data is not well-separated. This is true according to the plot we have above. Take a look at Dunn index, Dunn index is opposite from DB index, since it need to be large to indicate a good clustering. And the Dunn index in this part is also very small, show us the same result as DB index.

Here is the same energy dataset but with k=2 validation result:



It’s clearly that the DB index and Dunn index got closer with k=2. That means the clusters are now well-separated.

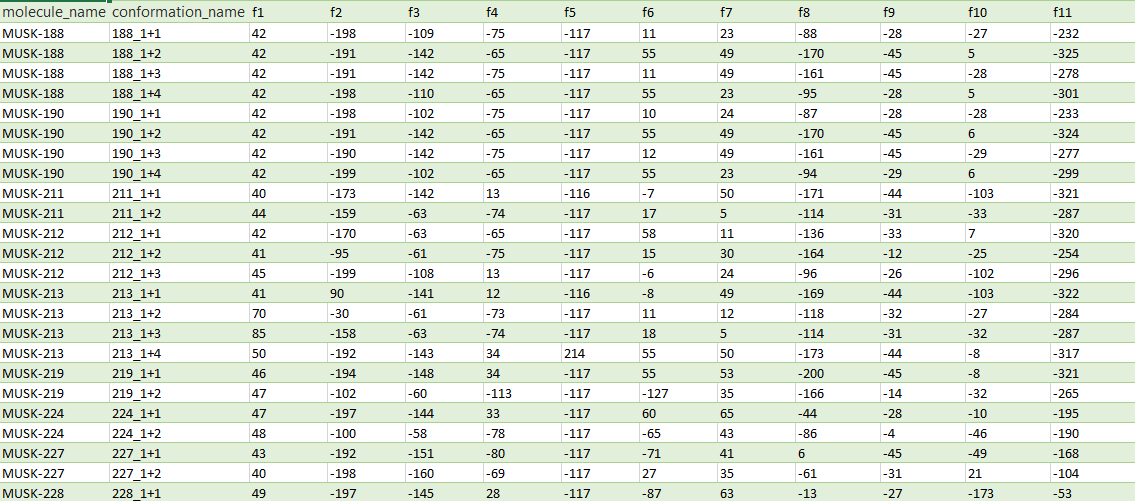
Now we apply the same validation techniques on machine dataset, and got this result with k=4:



The DB index is smaller than 1 but the Dunn index is still very small, this means our clustering attempt failed to create well-separated clusters.

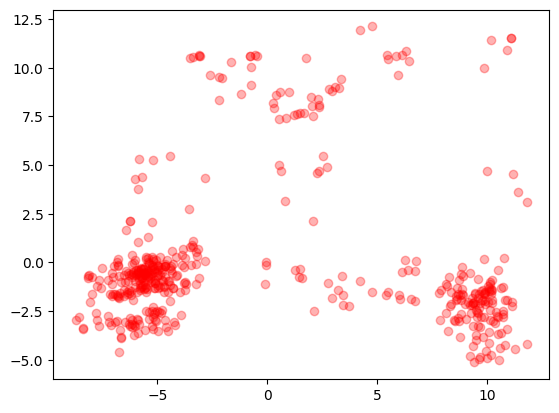
1. **Subspace clustering**
2. Visualize the dataset

For the dataset that has more than 100 features, we chose a dataset about the musk and non-musk molecules.



* 1. Data about musk and non-musk molecules.

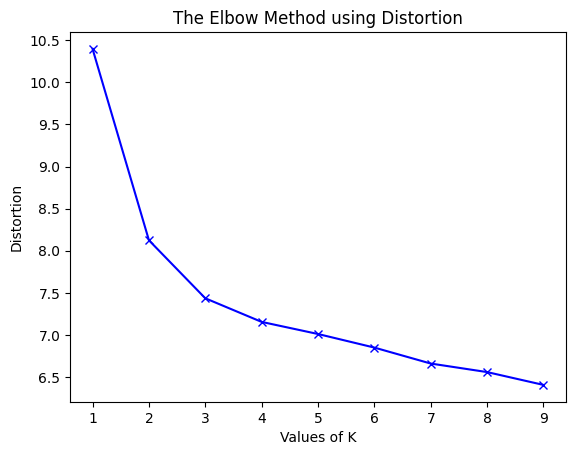
We simply apply PCA method to reduce the dataset into 2 dimensions data. And we got this nicely graph of the dataset.



* 1. 2D visualization of the dataset

1. Apply K-means

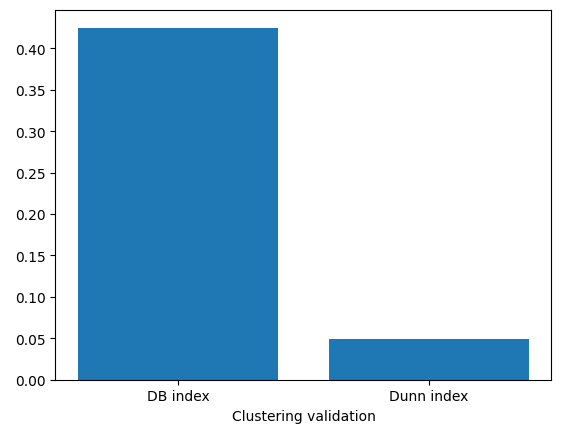
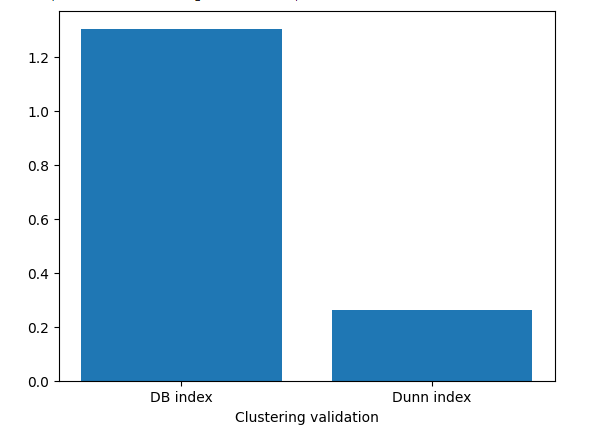
Using the same steps as the two previous datasets, first we will use the “elbow method” to find the suitable number of centroids.



As we see from the graph, we can choose 3 is the number of centroids. After that we assign the data point to the suitable cluster, and we got the result.

1. Analyze result

To compare the performance between two clusters before and after PCA. We can compare the clustering quality using 2 internal validation methods, Davies-Bouldin index and Dunn index.

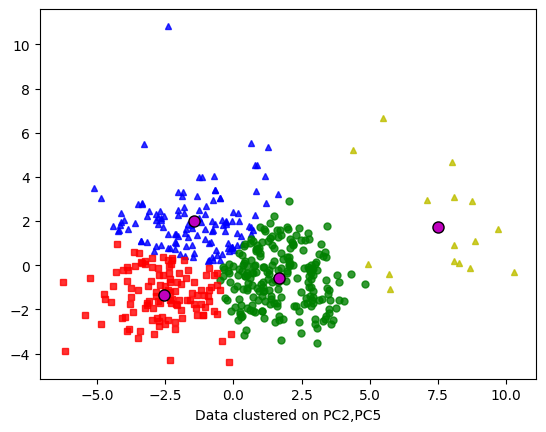


DB and Dunn index before and after PCA

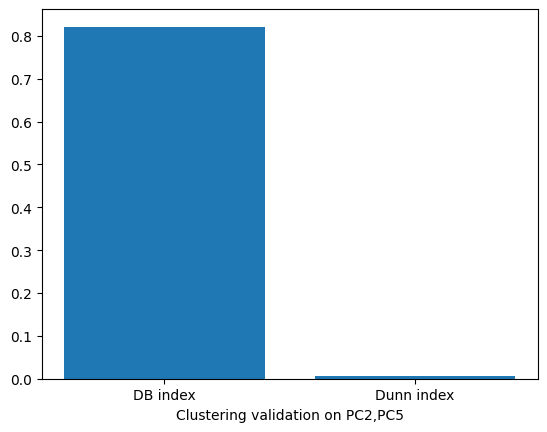
As we can see from the bar charts, DB index after PCA is smaller that means the clusters are separated better compared to before PCA. Meanwhile the Dunn index keep getting smaller, indicate that clustering result is not improved. But we must take in account that the number of dimensions is greatly reduced, so that Dunn index decreasing is normal. In conclusion, the clustering result after PCA has better performance compared to clustering with original dataset.

1. Vary subspaces

We applied PCA with 8 components, and then we chose 2 components number 2 and 5 to be our subspace in this section. Apply K-means with k=4 as we did with the original dataset, we got this plot as a result.



With the following performance calculated:



Compared the performance of clustering on this subspace with the 2D dataset we created in previous section, we can easily notice that the performance is greatly reduced. This is because in the last section, we created a 2D subspace that represent the highest proportion of the original dataset while the subspace in this section is clearly not the best components for 2D visualization.