University of Science and Technology of Hanoi

Clustering report

Machine learning and Data mining II

Nguyễn Mạnh Hưng - 22BI13183

Nguyễn Trọng Minh - 22BI13304

TABLE OF CONTENT

I. K-means clustering

1.Experiment protocol

In this lab work, we chose 2 data sets, one is data about the energy efficiency of the heating load and cooling load requirements of buildings, and the other is CPUs performance data of different machines.

Vendor Nam	Model Name	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP	ERP
amdahl	470v/7	29,00	8000,00	32000,00	32,00	8,00	32,00	269,00	253,00
adviser	32/60	125,00	256,00	6000,00	256,00	16,00	128,00	198,00	199,00
amdahl	470v/7a	29,00	8000,00	32000,00	32,00	8,00	32,00	220,00	253,00
amdahl	470v/7b	29,00	8000,00	32000,00	32,00	8,00	32,00	172,00	253,00
amdahl	470v/7c	29,00	8000,00	16000,00	32,00	8,00	16,00	132,00	132,00
amdahl	470v/b	26,00	8000,00	32000,00	64,00	8,00	32,00	318,00	290,00
amdahl	580-5840	23,00	16000,00	32000,00	64,00	16,00	32,00	367,00	381,00
amdahl	580-5850	23,00	16000,00	32000,00	64,00	16,00	32,00	489,00	381,00
amdahl	580-5860	23,00	16000,00	64000,00	64,00	16,00	32,00	636,00	749,00
amdahl	580-5880	23,00	32000,00	64000,00	128,00	32,00	64,00	1144,00	1238,00

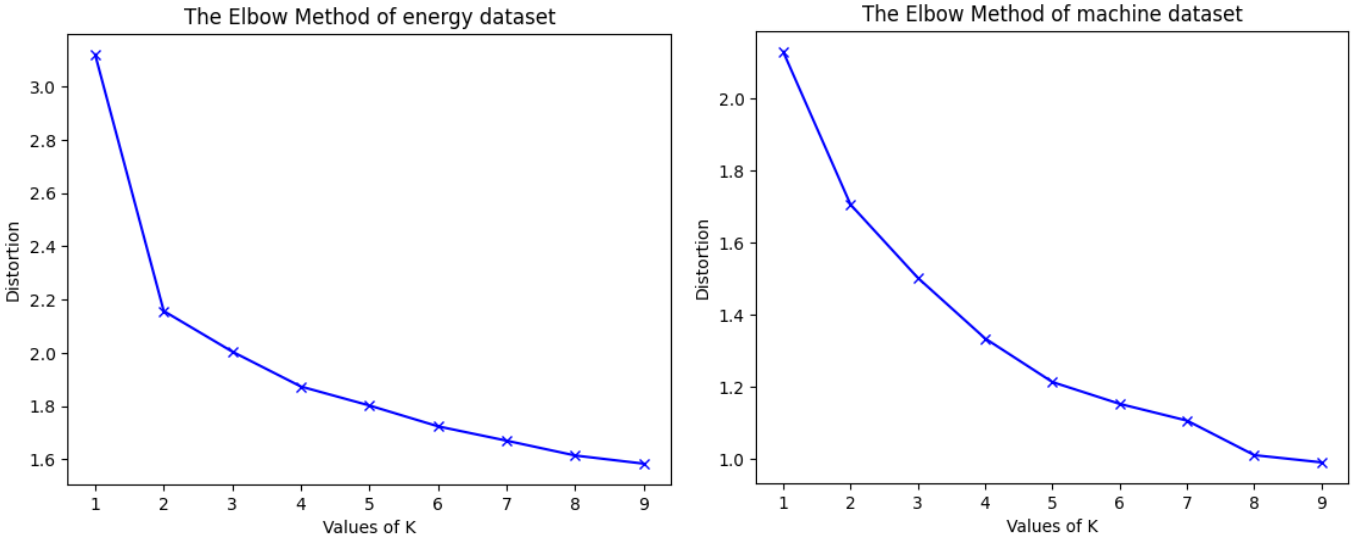
1.1. Data about CPUs performance

Relative Compactness	Surface Area	Wall Area	Roof Area	Overall Height	Orientation	Glazing Area	Glazing Area Distribution	Heating Load	Cooling Load
0,98	514,50	294,00	110,25	7,00	2	0,00	0	15,55	21,33
0,98	514,50	294,00	110,25	7,00	3	0,00	0	15,55	21,33
0,98	514,50	294,00	110,25	7,00	4	0,00	0	15,55	21,33
0,98	514,50	294,00	110,25	7,00	5	0,00	0	15,55	21,33
0,90	563,50	318,50	122,50	7,00	2	0,00	0	20,84	28,28
0,90	563,50	318,50	122,50	7,00	3	0,00	0	21,46	25,38
0,90	563,50	318,50	122,50	7,00	4	0,00	0	20,71	25,16
0,90	563,50	318,50	122,50	7,00	5	0,00	0	19,68	29,60

1.2. 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.

2.Centroid initialization

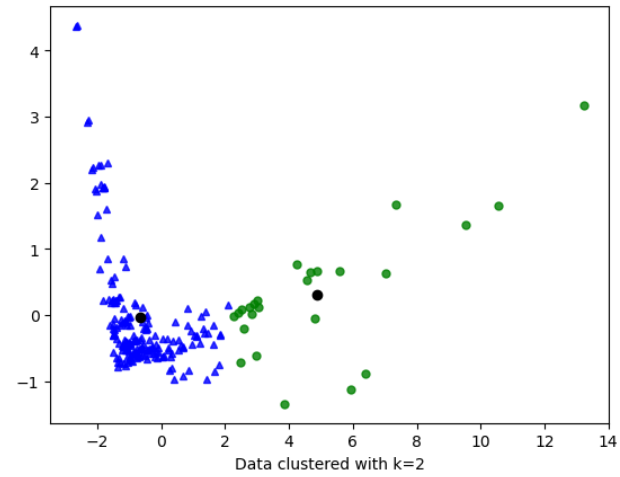
In the above session, we talked about an overall protocol of K-means clustering, I want to go deeper into the centroid initialization in this part of the report.

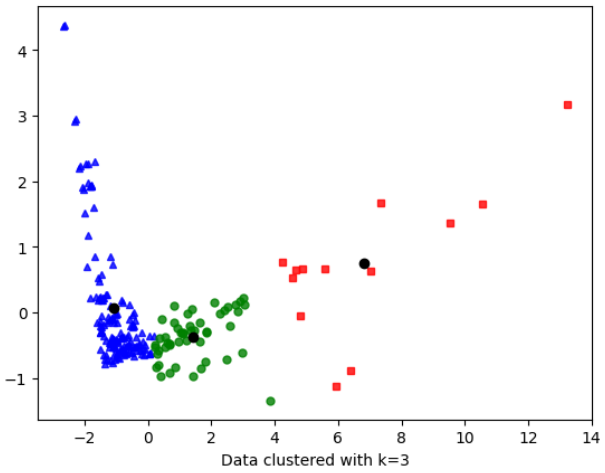
Normally, the first centroids of the algorithm are generated randomly inside data set. But in this experiment, I applied a method to determine those centroids, called “greedy k-means++” published by Arthur and Vassilvitskii in 2007. The greedy k-means++ algorithm samples k initial centers by adaptive 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.

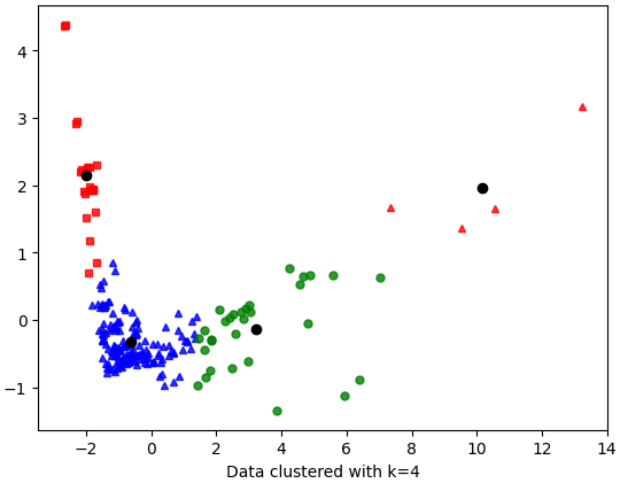
This technique is believed to speed up convergence that provides an O(log k)-approximation in expectation.

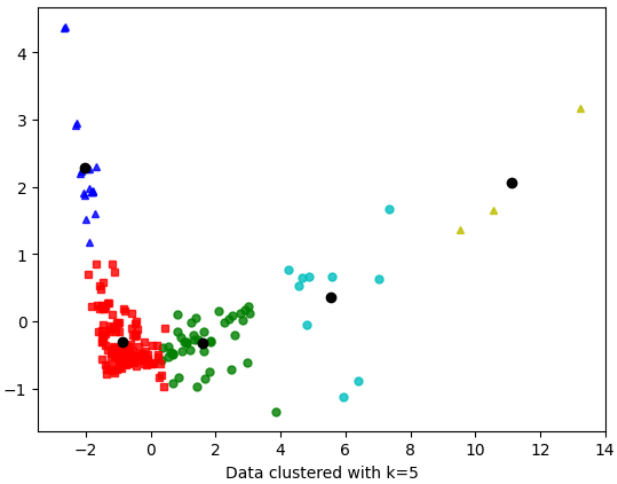
3.Analyze the result

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



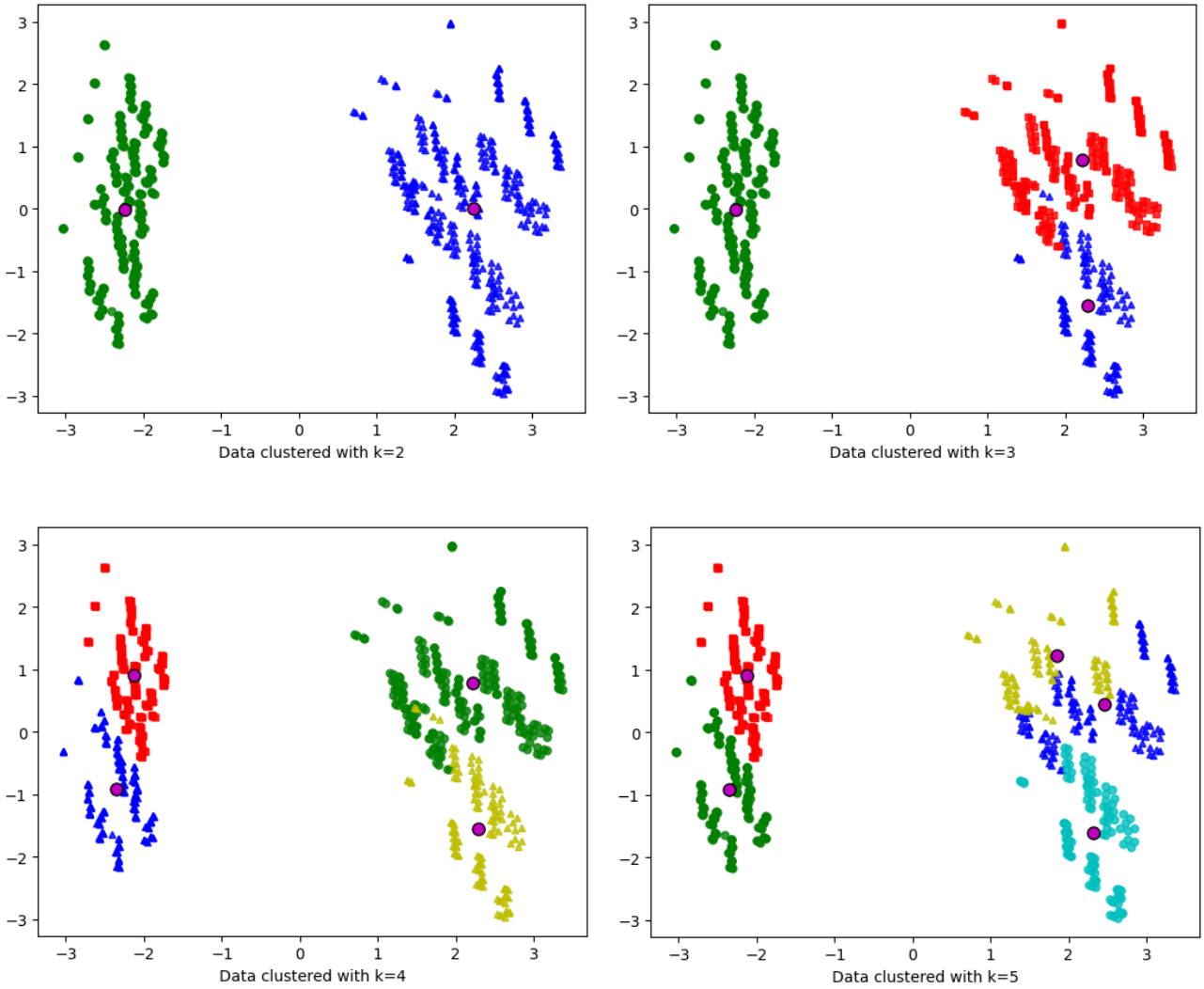






As we can see in case k = 4 and 5, there is a cluster with tiny amount of data points, 4 and 3. These 2 clusters should be merged into bigger cluster for better performance.

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

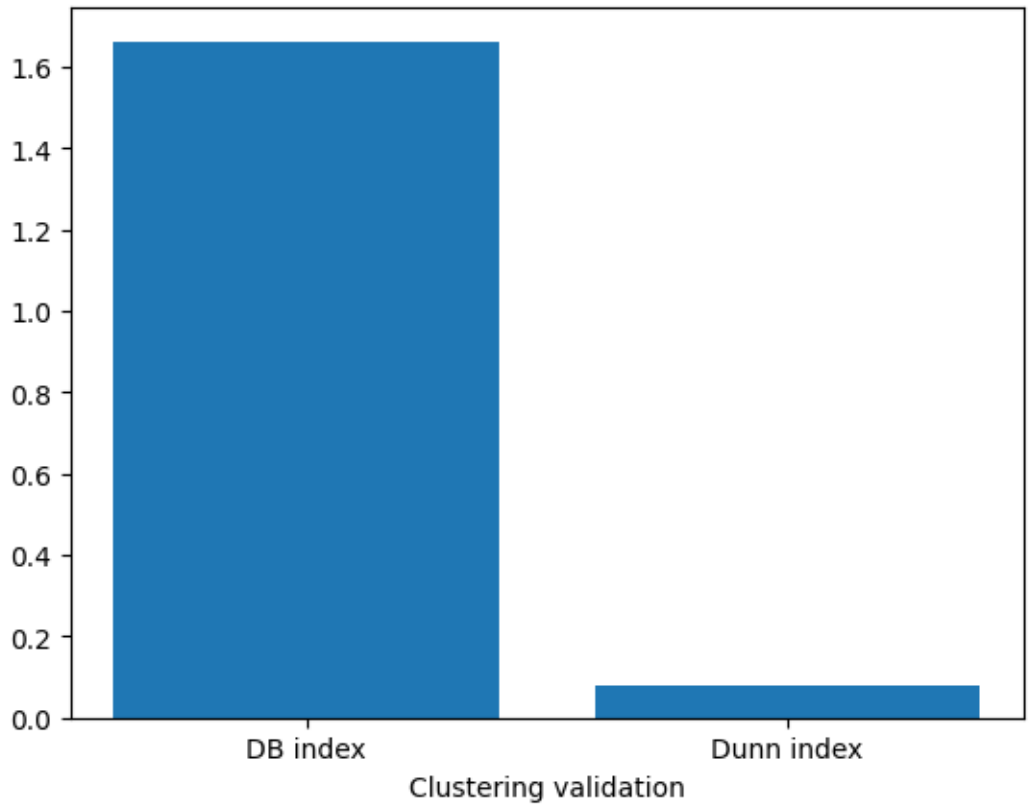


The plotted data show that k=2 is the best number of clusters for this dataset. We can also see some data points might be clustered into wrong cluster, but this phenomenal is just data error due to PCA.

4. Clustering quality

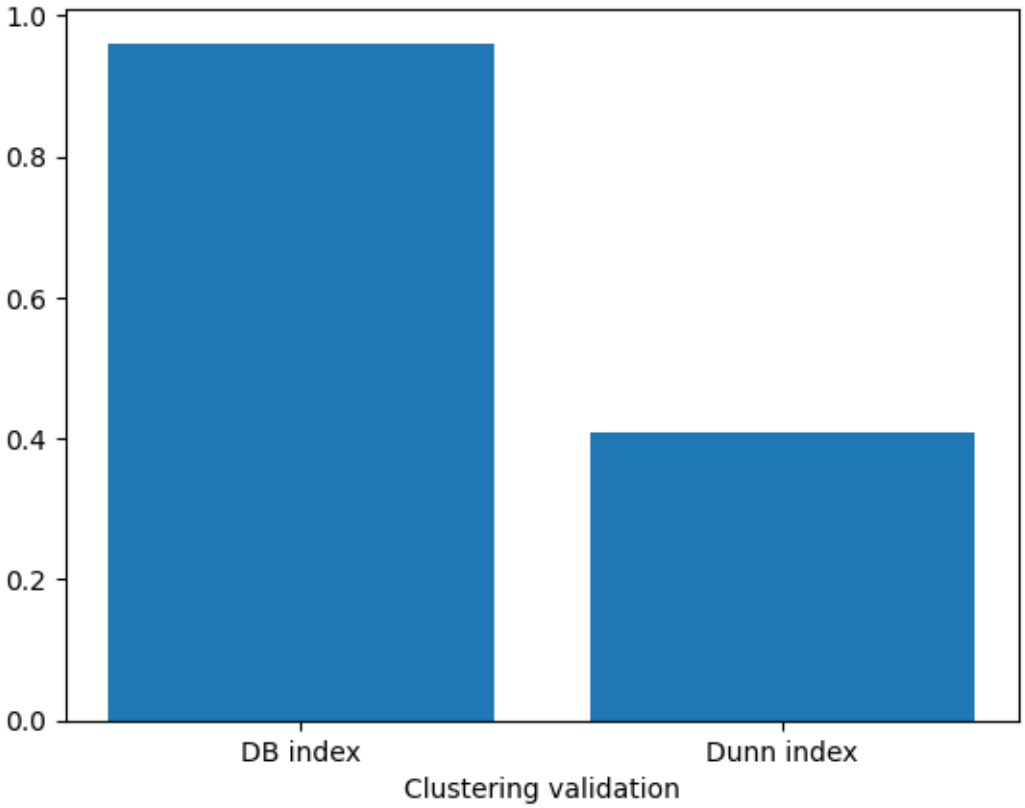
To validate the quality of the cluster, we apply 2 internal validation methods, Davies-Bouldin index and Dunn index.

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



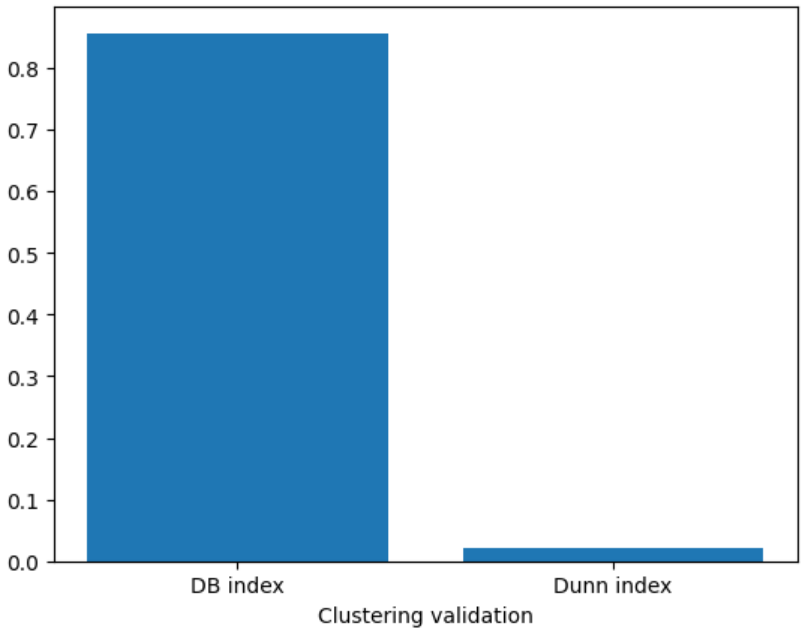
We can see that DB index is quite high, that means the data is not well-separated. This is true according to the plot we have above. Take a look at Dunn index, Dunn index is opposite from DB index, since it need to be large to indicate a good clustering. And the Dunn index in this part is also very small,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

II. Subspace clustering

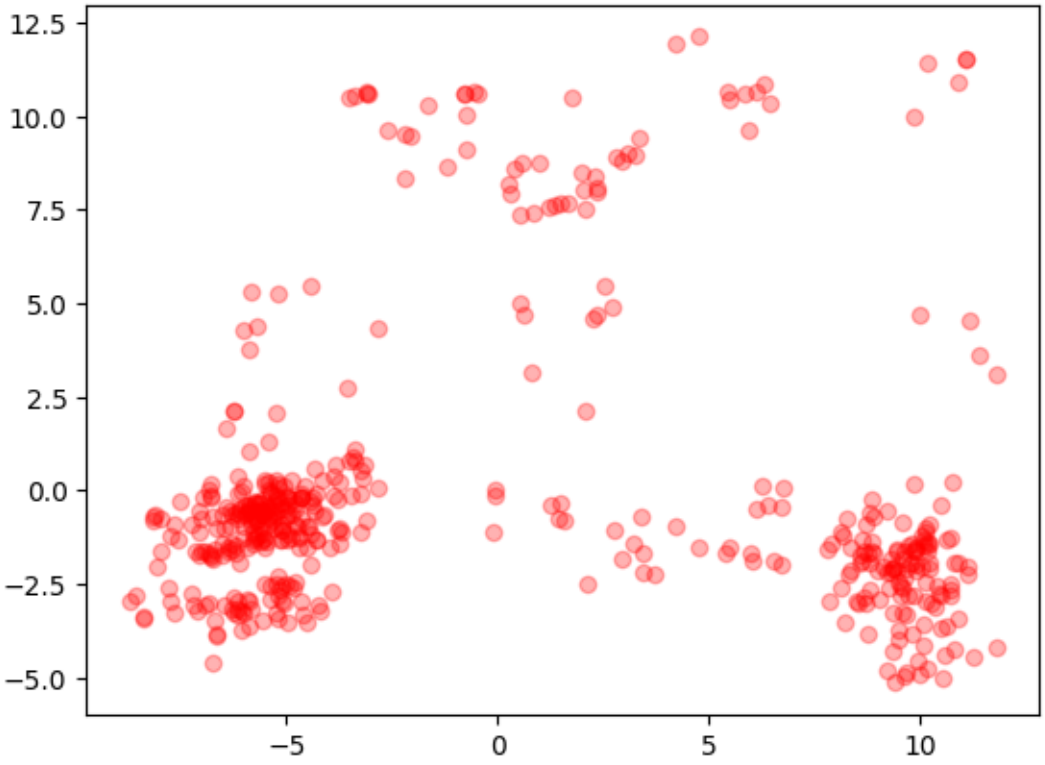
1. Visualize the dataset

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

molecule_name	conformation_name	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11
MUSK-188	188_1+1	42	-198	-109	-75	-117	11	23	-88	-28	-27	-232
MUSK-188	188_1+2	42	-191	-142	-65	-117	55	49	-170	-45	5	-325
MUSK-188	188_1+3	42	-191	-142	-75	-117	11	49	-161	-45	-28	-278
MUSK-188	188_1+4	42	-198	-110	-65	-117	55	23	-95	-28	5	-301
MUSK-190	190_1+1	42	-198	-102	-75	-117	10	24	-87	-28	-28	-233
MUSK-190	190_1+2	42	-191	-142	-65	-117	55	49	-170	-45	6	-324
MUSK-190	190_1+3	42	-190	-142	-75	-117	12	49	-161	-45	-29	-277
MUSK-190	190_1+4	42	-199	-102	-65	-117	55	23	-94	-29	6	-299
MUSK-211	211_1+1	40	-173	-142	13	-116	-7	50	-171	-44	-103	-321
MUSK-211	211_1+2	44	-159	-63	-74	-117	17	5	-114	-31	-33	-287
MUSK-212	212_1+1	42	-170	-63	-65	-117	58	11	-136	-33	7	-320
MUSK-212	212_1+2	41	-95	-61	-75	-117	15	30	-164	-12	-25	-254
MUSK-212	212_1+3	45	-199	-108	13	-117	-6	24	-96	-26	-102	-296
MUSK-213	213_1+1	41	90	-141	12	-116	-8	49	-169	-44	-103	-322
MUSK-213	213_1+2	70	-30	-61	-73	-117	11	12	-118	-32	-27	-284
MUSK-213	213_1+3	85	-158	-63	-74	-117	18	5	-114	-31	-32	-287
MUSK-213	213_1+4	50	-192	-143	34	214	55	50	-173	-44	-8	-317
MUSK-219	219_1+1	46	-194	-148	34	-117	55	53	-200	-45	-8	-321
MUSK-219	219_1+2	47	-102	-60	-113	-117	-127	35	-166	-14	-32	-265
MUSK-224	224_1+1	47	-197	-144	33	-117	60	65	-44	-28	-10	-195
MUSK-224	224_1+2	48	-100	-58	-78	-117	-65	43	-86	-4	-46	-190
MUSK-227	227_1+1	43	-192	-151	-80	-117	-71	41	6	-45	-49	-168
MUSK-227	227_1+2	40	-198	-160	-69	-117	27	35	-61	-31	21	-104
MUSK-228	228_1+1	49	-197	-145	28	-117	-87	63	-13	-27	-173	-53

1.1. Data about musk and non-musk molecules.

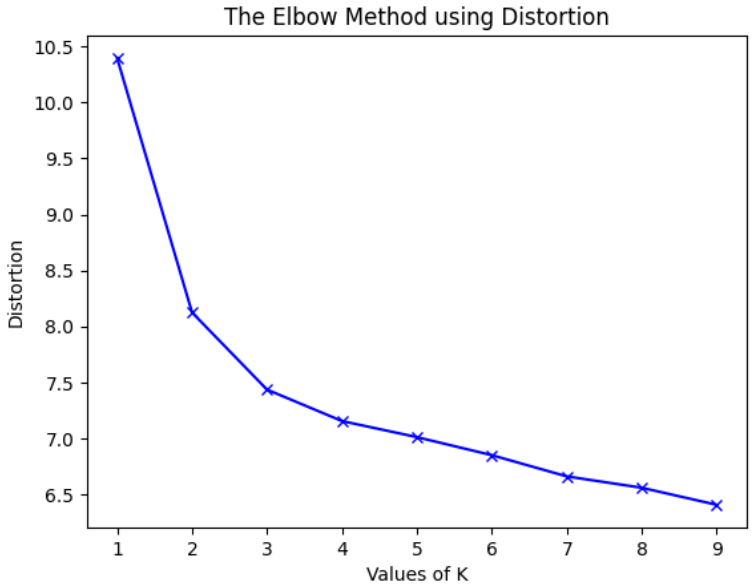
We simply apply PCA method to reduce the dataset into 2 dimensions data. And we got this nicely graph of the dataset.



1.2. 2D visualization of the dataset

2. Apply K-means

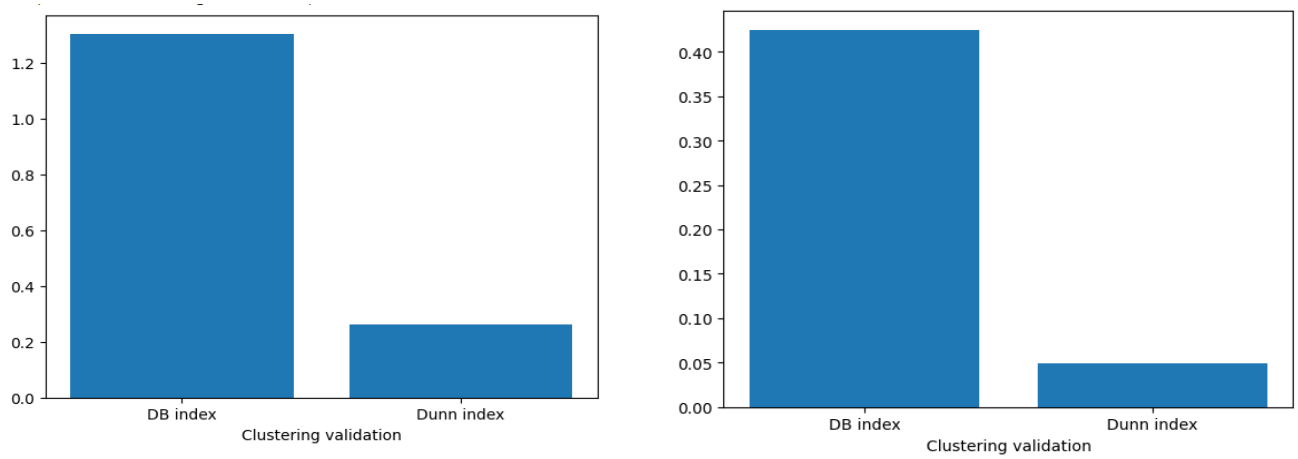
Using the same steps as the two previous datasets, first we will use the “elbow method” to find the suitable number of centroids.



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.

3. 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.

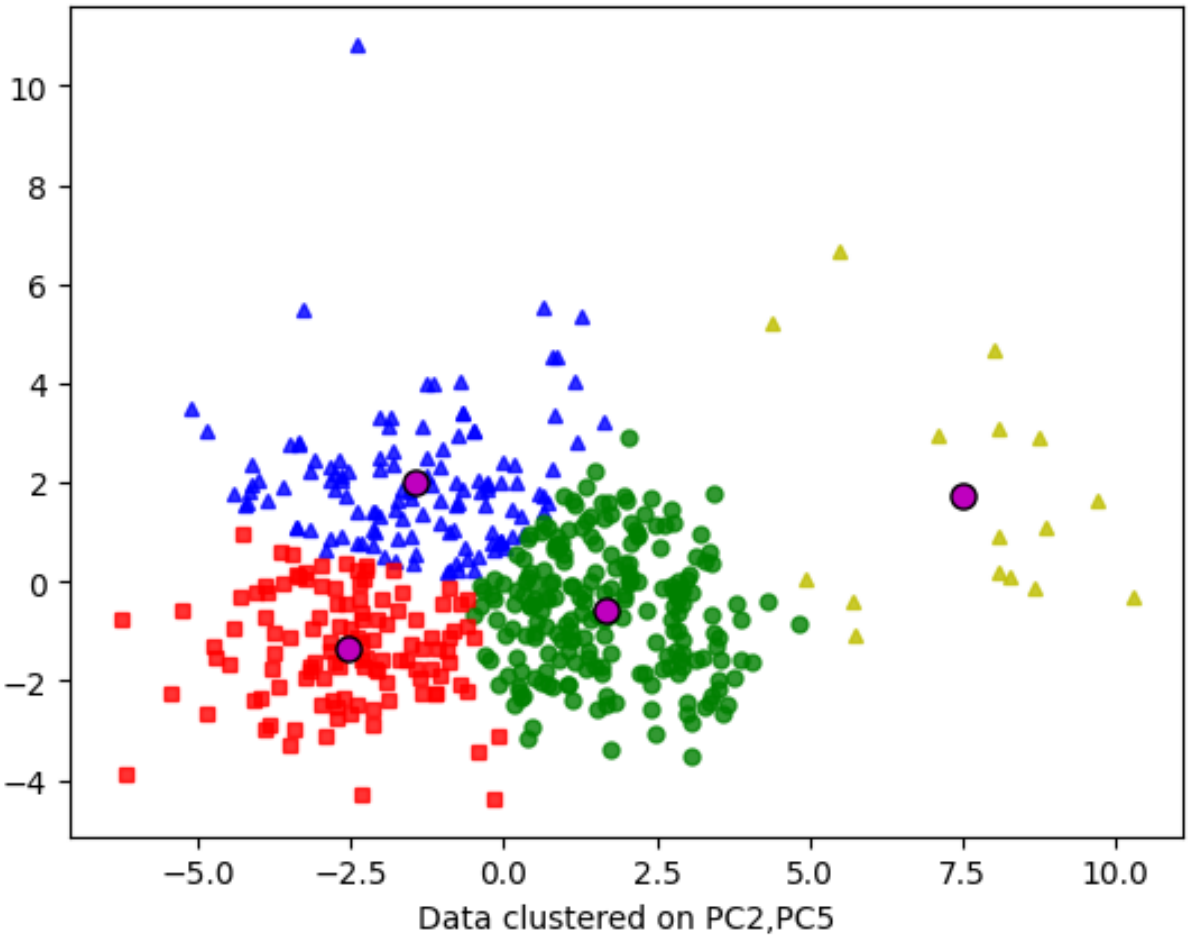


DB and Dunn index before and after PCA

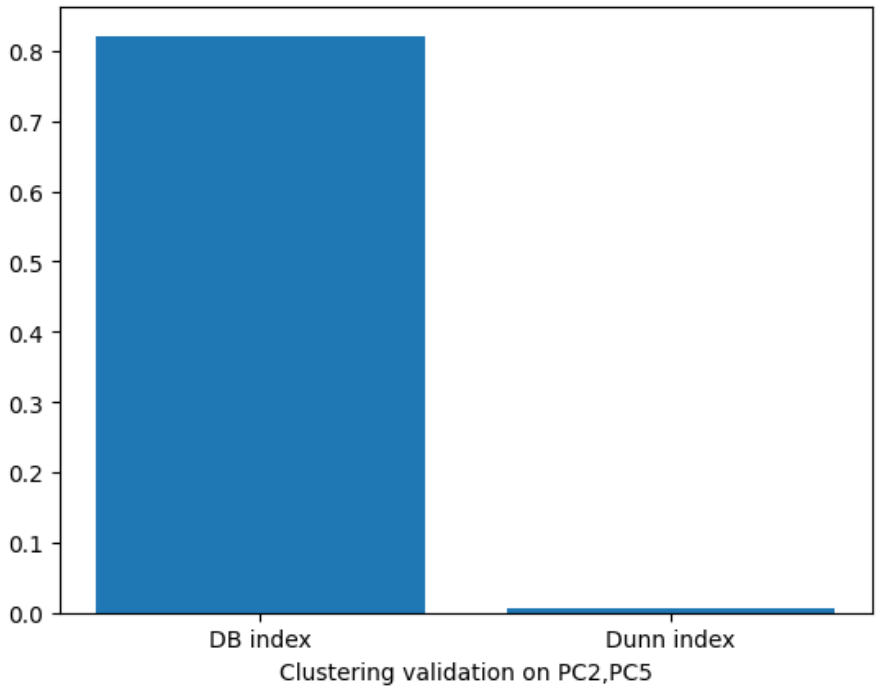
As we can see from the bar charts, DB index after PCA is smaller that means the clusters are separated better compared to before PCA. Meanwhile the Dunn index keep getting smaller, indicate that clustering result is not improved. But we must take in account that the number of dimensions is greatly reduced,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

4. 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.