

Cluster Analysis -- Basic Concepts and Algorithms

Peerapon S.

Machine Learning (1/67)

Topics

Basic K-means algorithm

Choosing number of clusters

Insights from clustering

Clustering

Questions

What are customer behaviors?

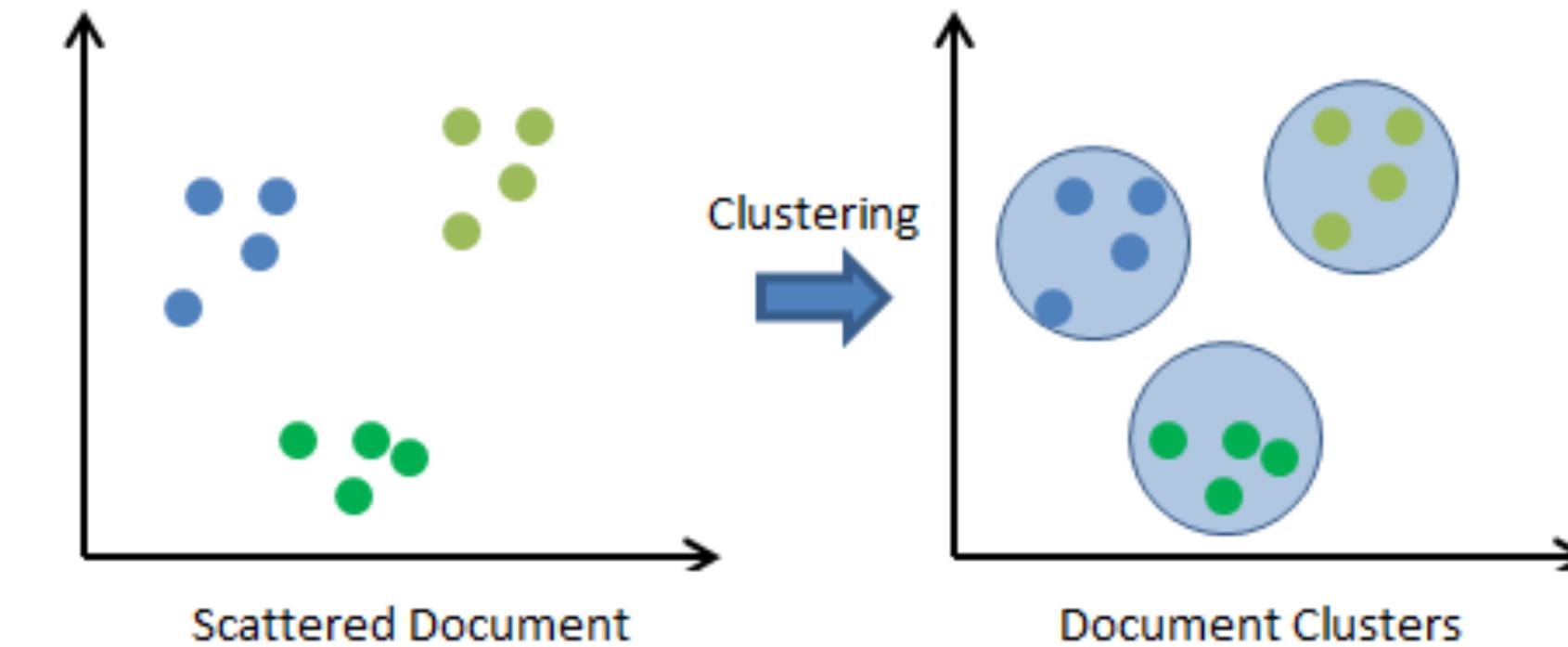
Who are our customers?

What are groups of species or syndromes, documents, etc. ?

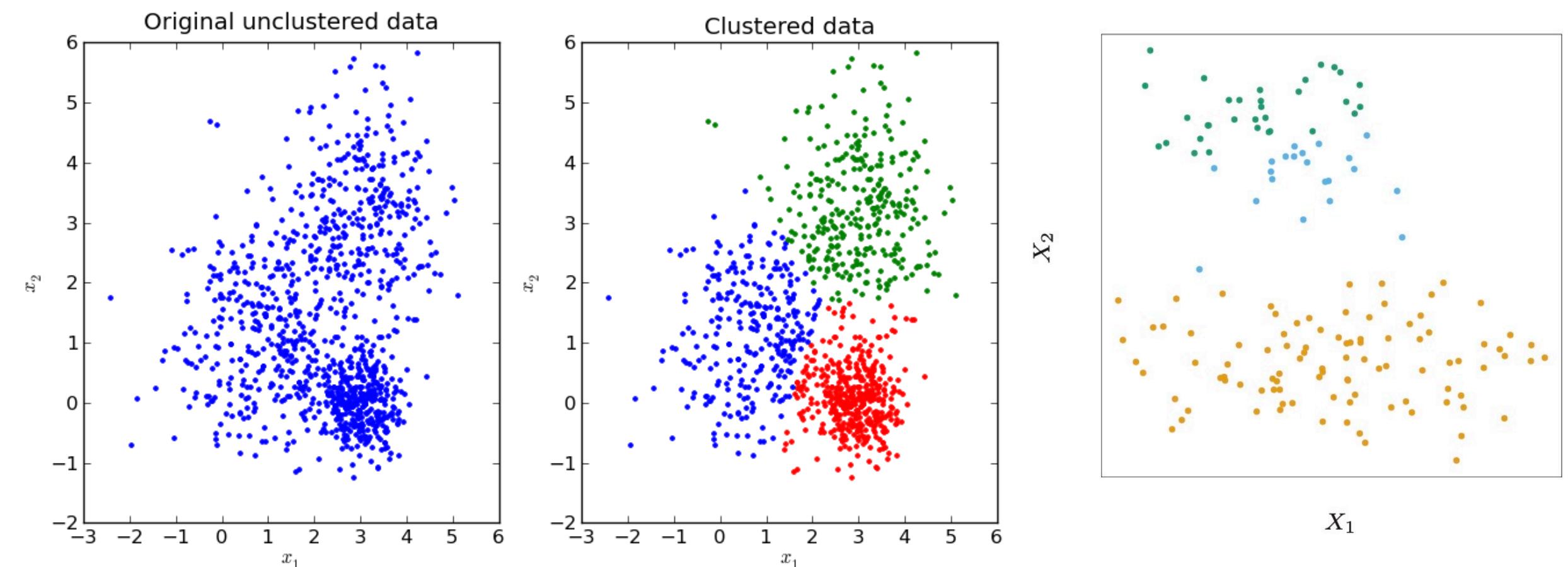
No Y data !

Goal

Find natural grouping of data



A Clustering =
A set of clusters



Centroid of a Cluster

Imaginary location representing the center of a cluster.

For a d -dimensional data set,

- The centroid of a cluster is the vector with d elements.
- Average of (scaled) numeric or (encoded) ordinal variables

In representative/prototype-based clustering, objects are assigned to their closest centroids.

	Age	Height	Weight
Person 1	45	170	64
Person 2	31	172	77
Person 3	29	159	56
Person 4	64	163	72

$$\text{Centroid } \mu = \{42.25, 166, 67.25\}$$

K-Means

- Initial centroid (k)
- Find distance ទំនាក់ទំនង រវាងក្នុងក្នុង cluster និង គ្រប់គ្រងទំនាក់ទំនង
- Assign ទូទៅនៃទំនាក់ទំនង នៃជំនួយទំនាក់ទំនង
- Calculate mean ទំនាក់ទំនង នៃជំនួយទំនាក់ទំនង
- Repeat 2-4 ដល់មិនមែន ផ្តល់ជំនួយទំនាក់ទំនង

K-Means Clustering

Given n data points of dimension d and a clustering $C = \{C_1, C_2, \dots, C_K\}$ with centroids μ_k , $k = 1, 2, \dots, K$.

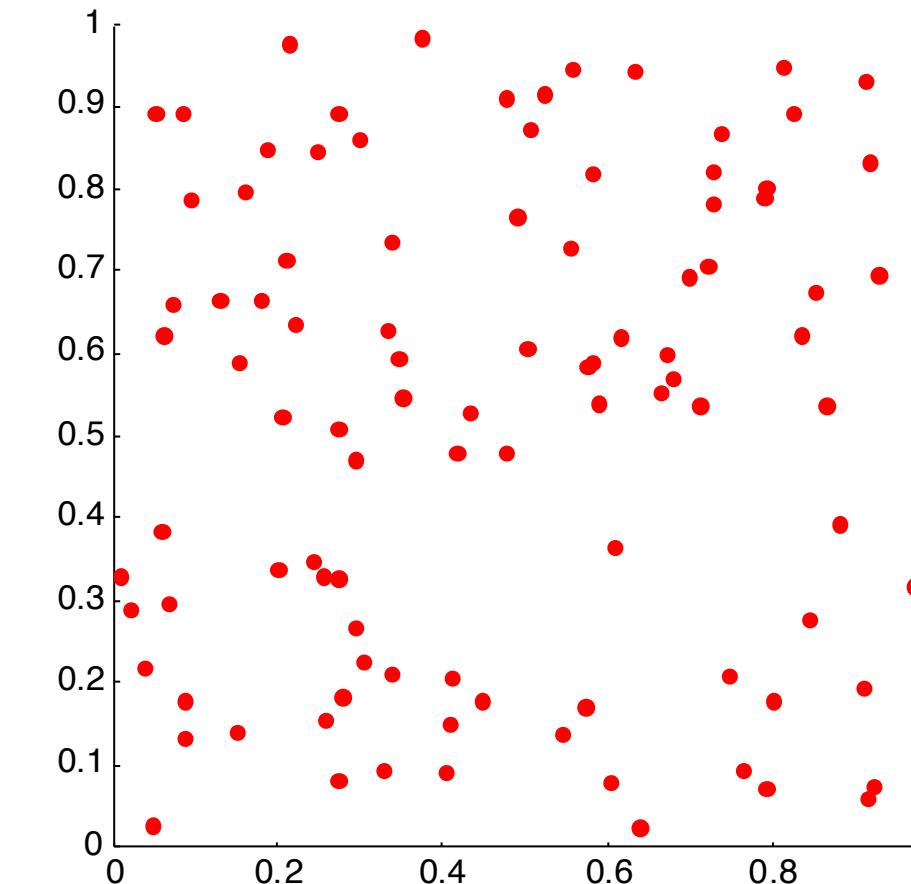
Define the sum of squared errors of cluster as



$$\text{SSE}(\mathcal{C}) = \sum_{k=1}^K \sum_{x_j \in C_k} \|x_j - \mu_k\|^2$$

Find the clustering $C^* = \{C_1, C_2, \dots, C_K\}$ that minimizes SSE: $\mathcal{C}^* = \arg \min_{\mathcal{C}} \text{SSE}(\mathcal{C})$

Optimal clustering requires exhaustive search !

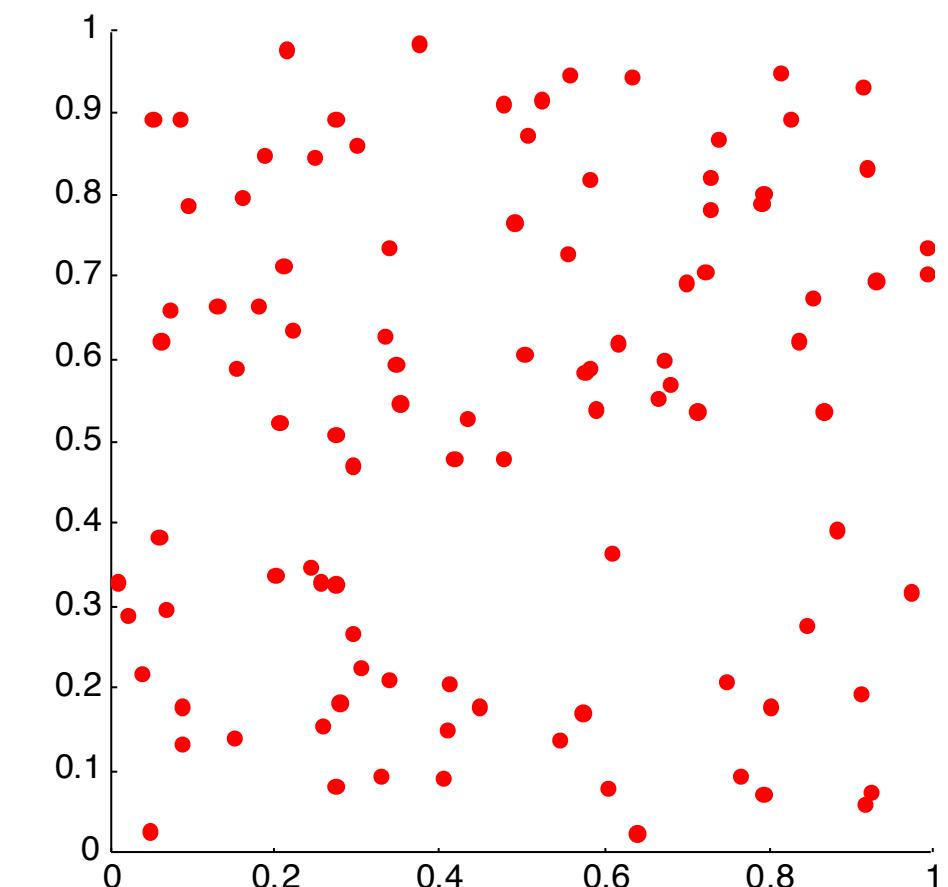


Greedy iterative approach is used to find a clustering that minimizes the SSE objective.

Provide good-enough solutions but not guarantee optimality.

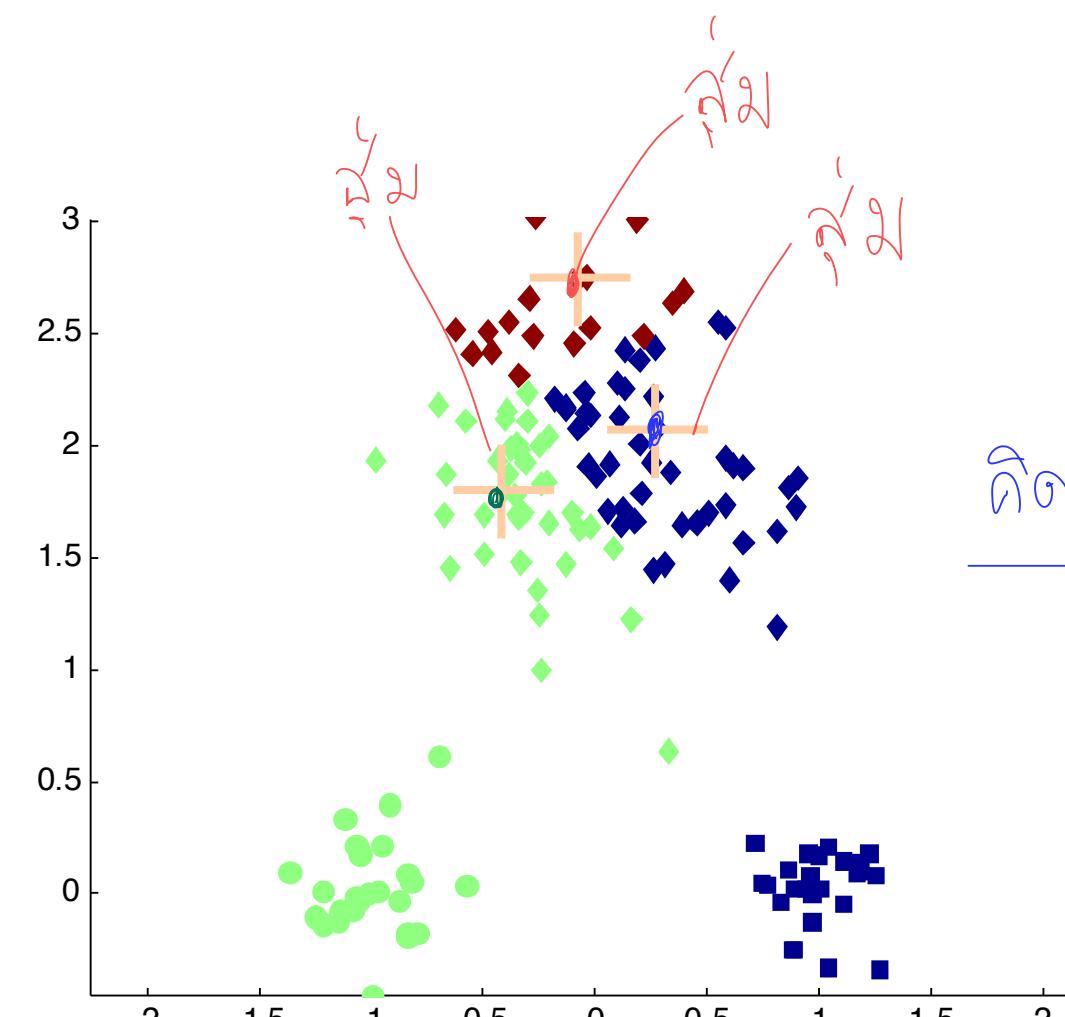
Lloyd's K-means algorithm

- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

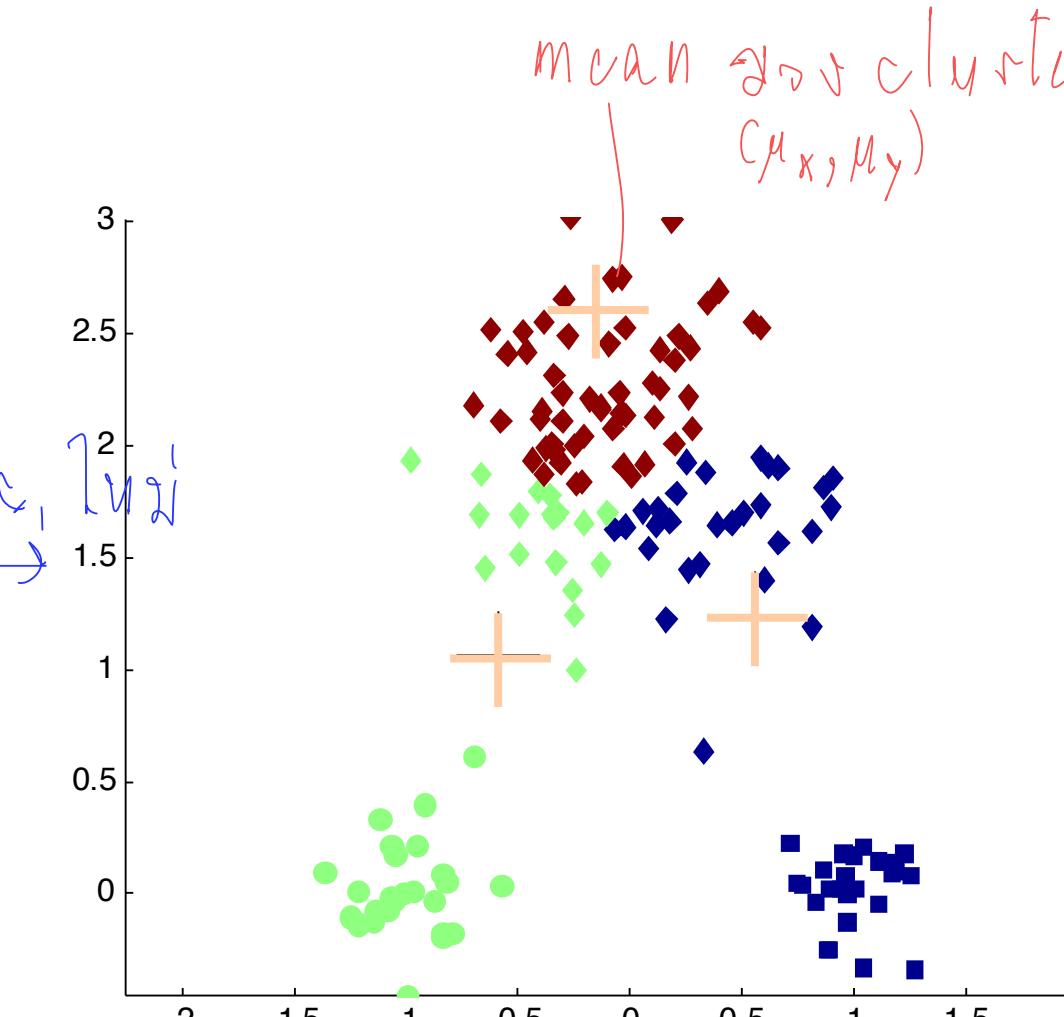


$$K = 3$$

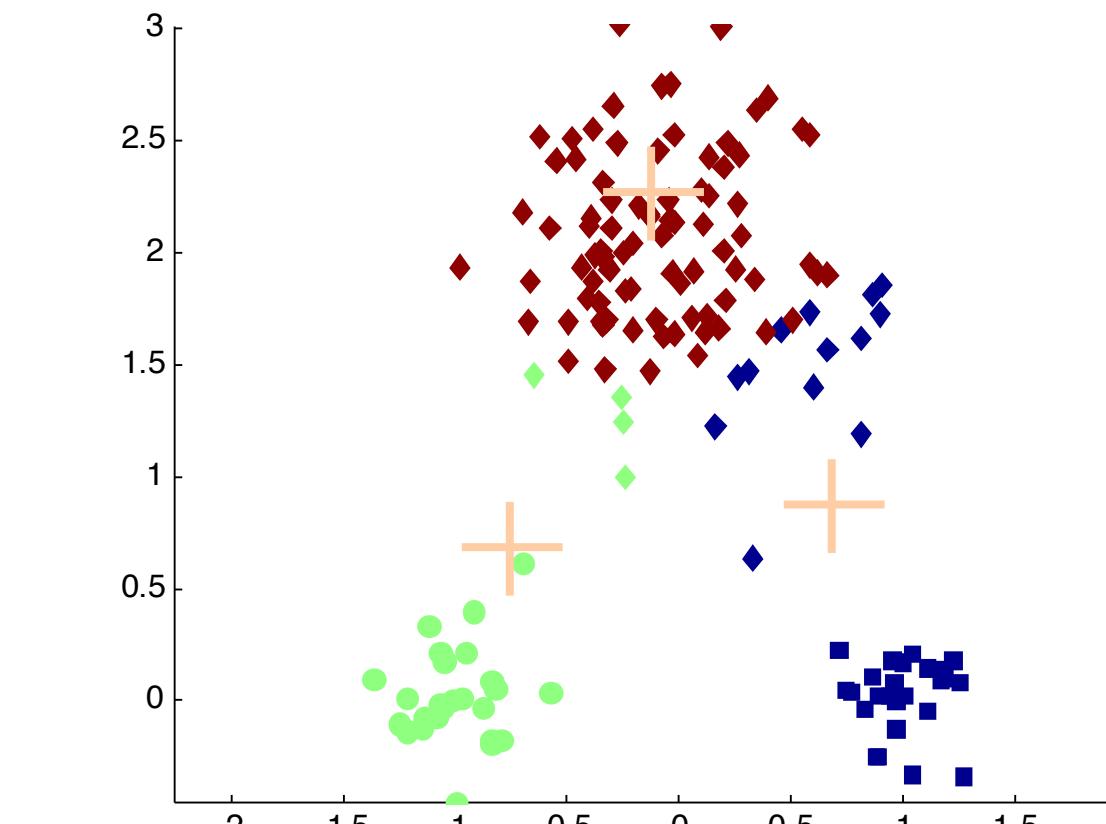
3 cluster



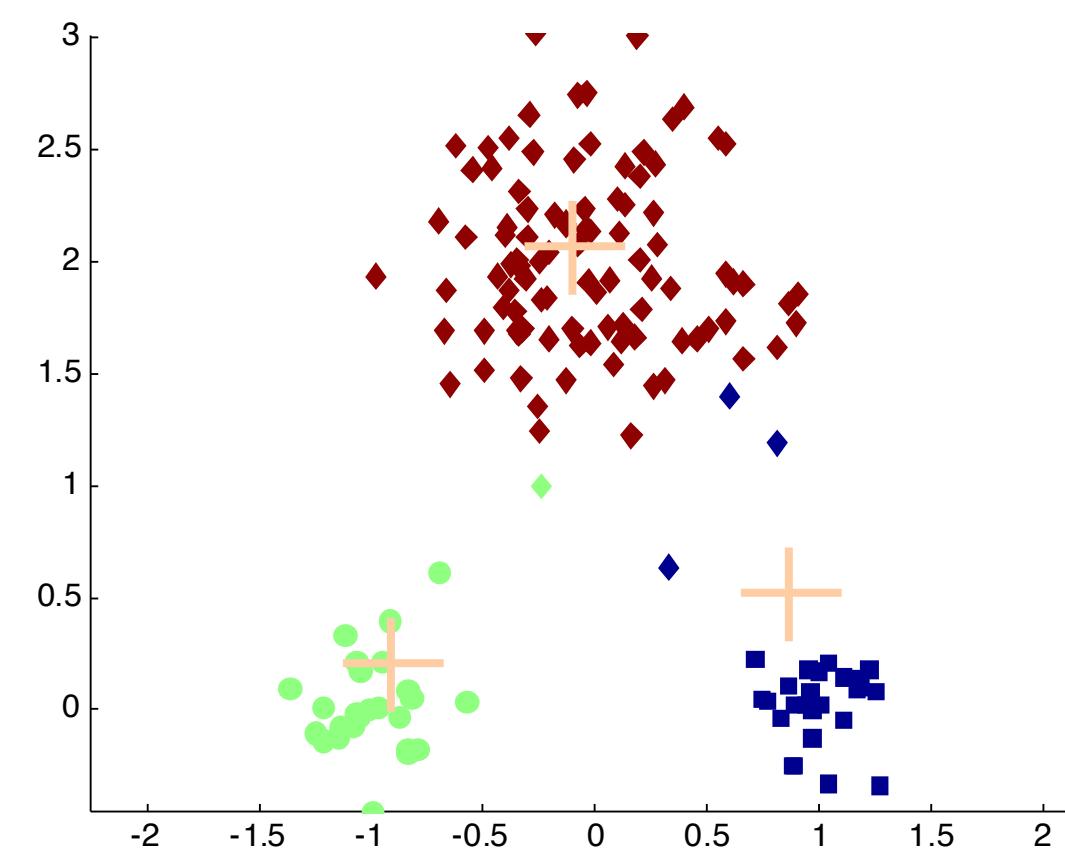
Iteration 1



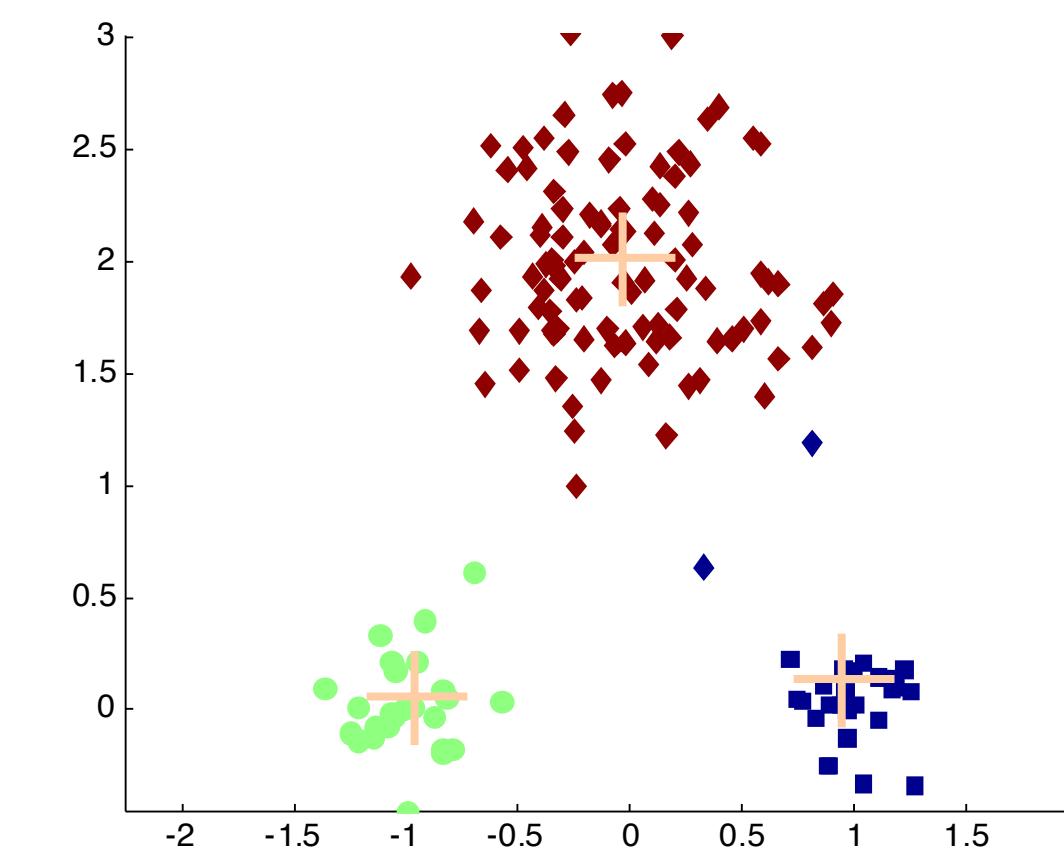
Iteration 2



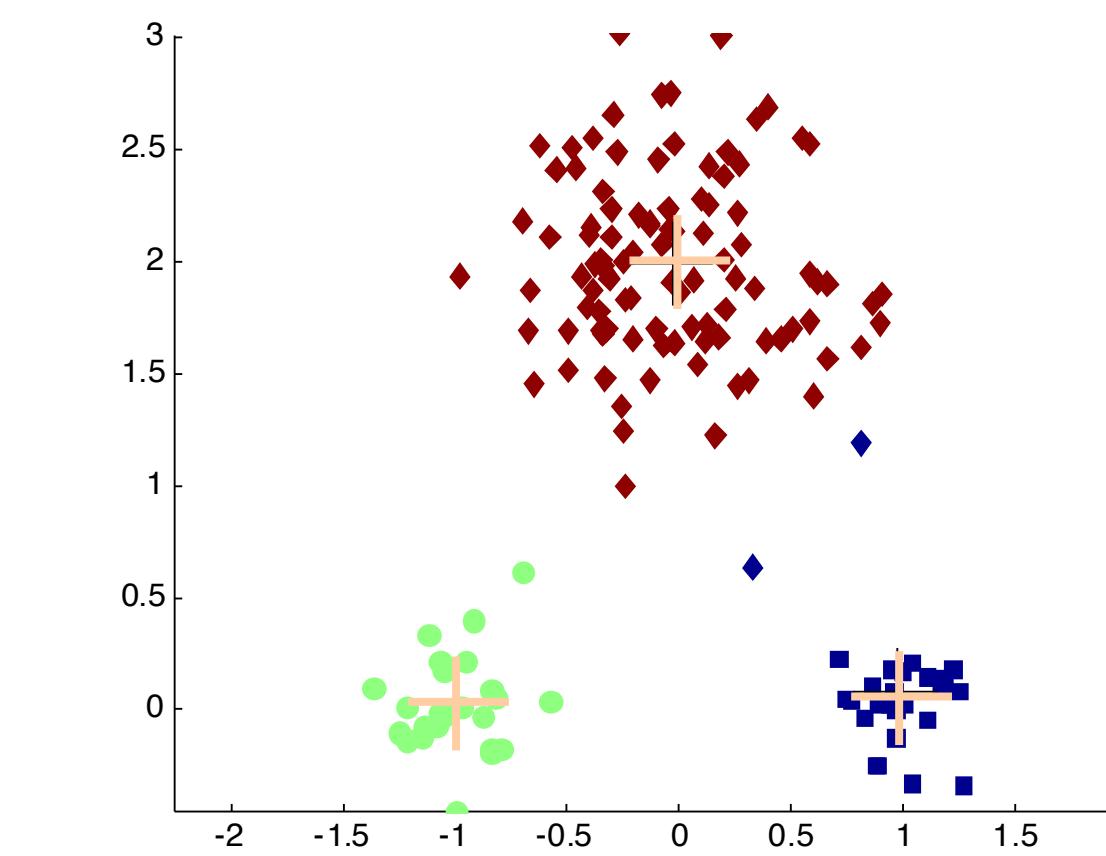
Iteration 3



Iteration 4

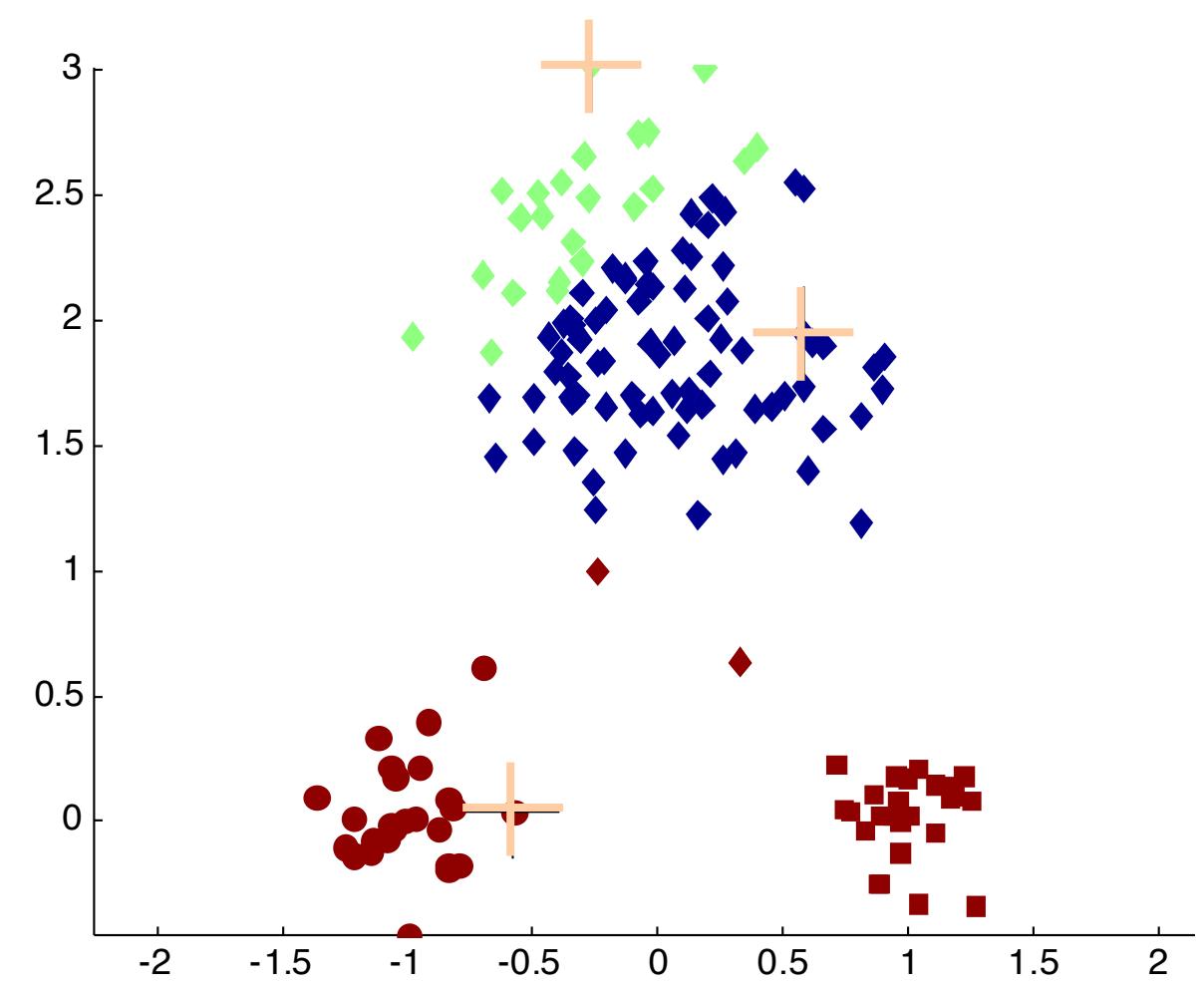


Iteration 5

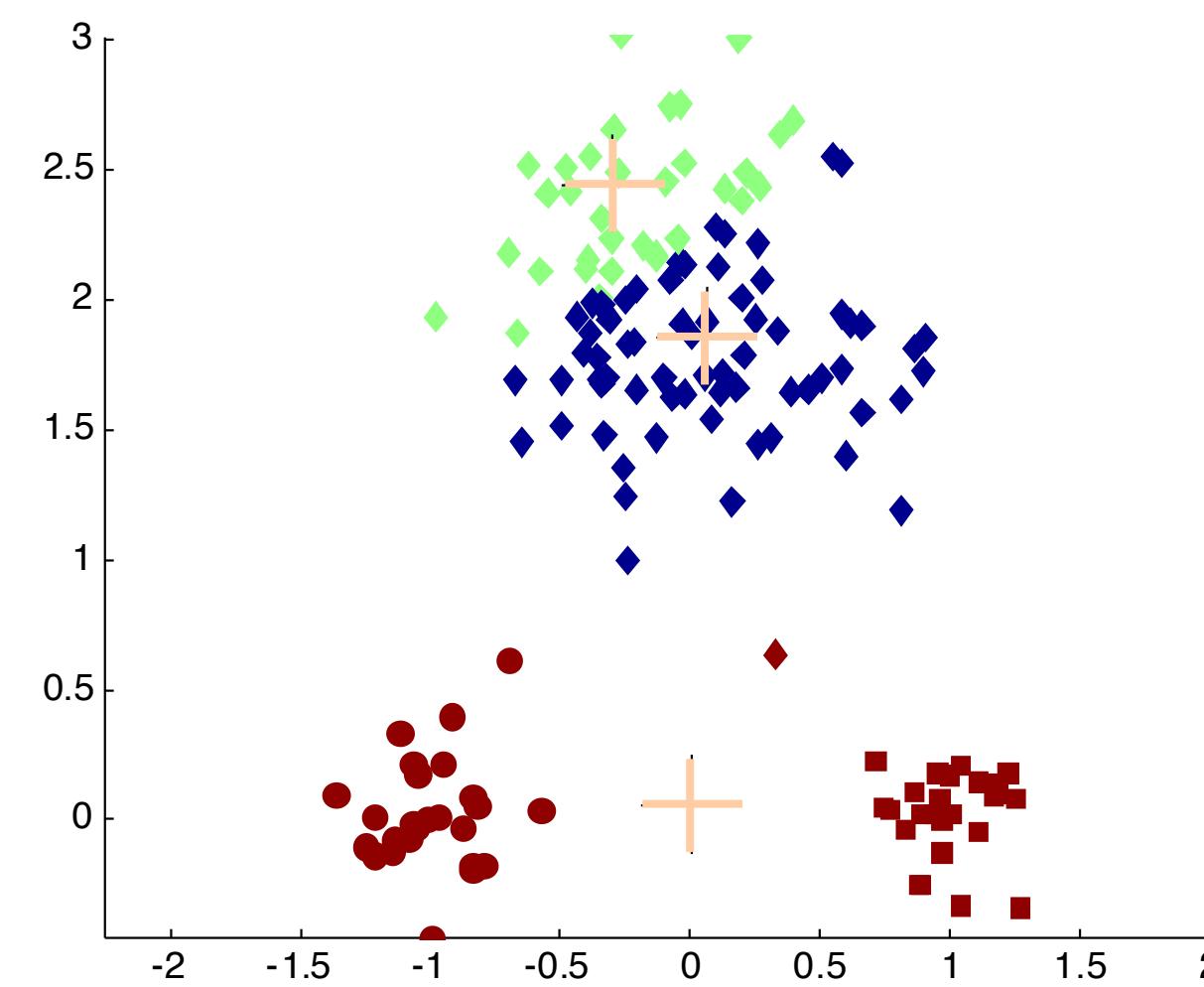


Iteration 6

Importance of Initial Conditions



Initialization



Termination



```
from sklearn.datasets import make_blobs

X, _ = make_blobs(n_samples=800, centers=4, random_state=42)
plt.scatter(X[:, 0], X[:, 1]);

from sklearn.cluster import KMeans
# Number of clusters
K = 6

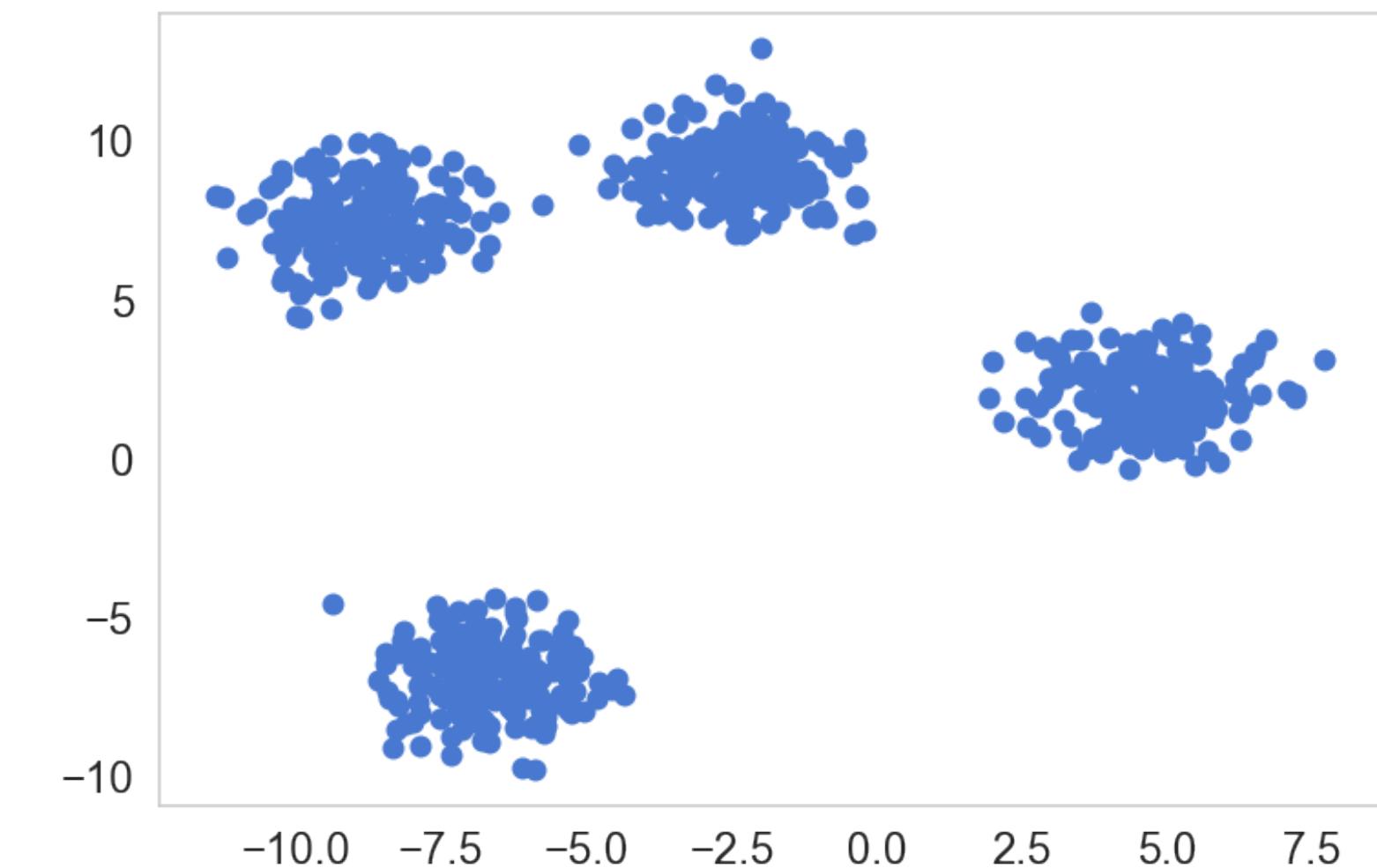
# Fit the input data
kmeans = KMeans(n_clusters=K, n_init=10).fit(X)

# Get the cluster labels
labels = kmeans.predict(X) # or use kmeans.labels

# Get sum of squares distance of all points
sse = kmeans.inertia_

# Centroid values
centroids = kmeans.cluster_centers_

np.set_printoptions(precision=3)
print(f"Sum of squared errors : {sse:.3f}")
print(f"\nCentroids : \n {centroids}")
```



```
Sum of squared errors : 1273.890

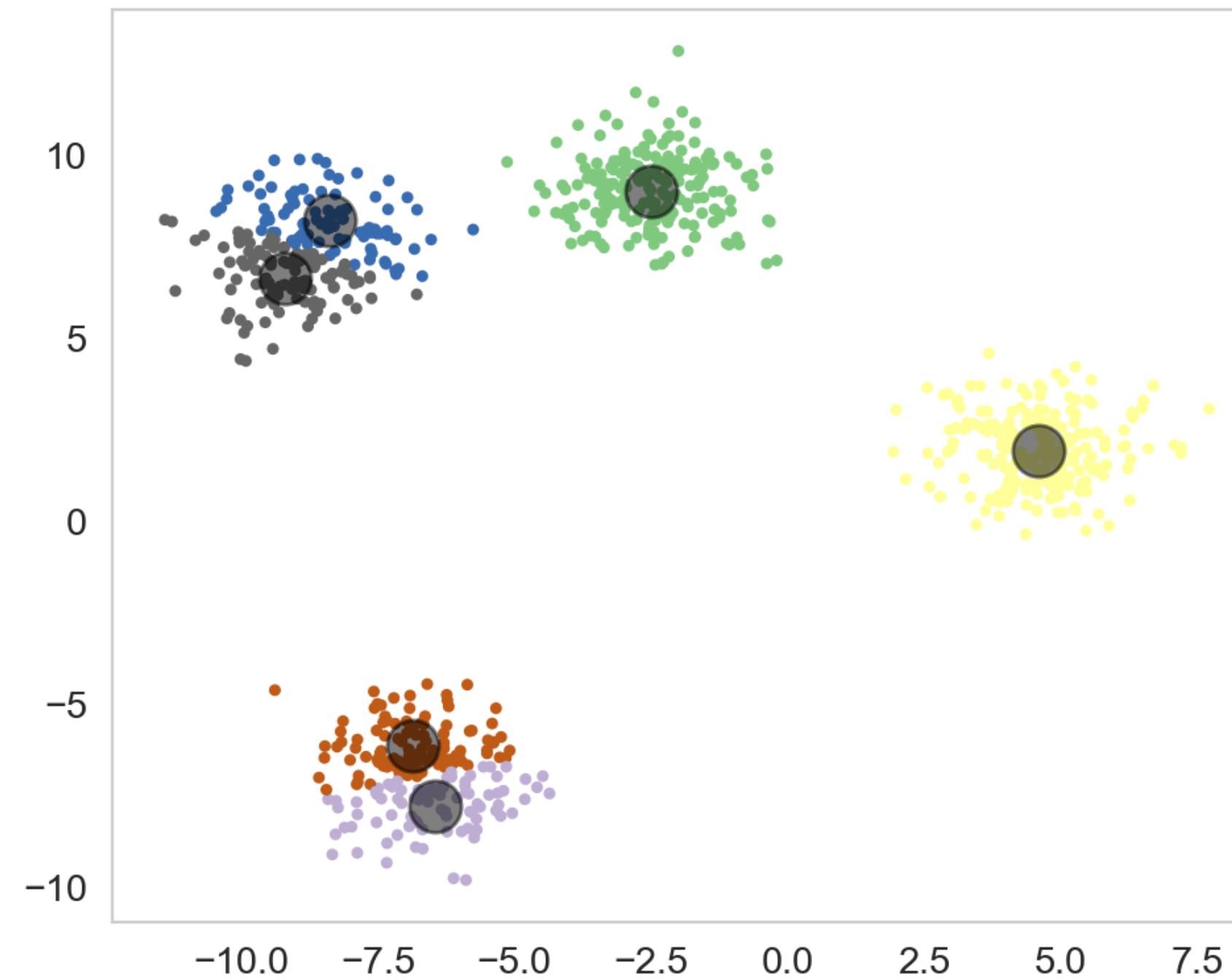
Centroids :
[[ -8.262  8.17 ]
 [ -6.682 -6.81 ]
 [  5.041  1.4  ]
 [ -2.501  9.053]
 [  3.96   2.726]
 [ -9.276  6.737]]
```



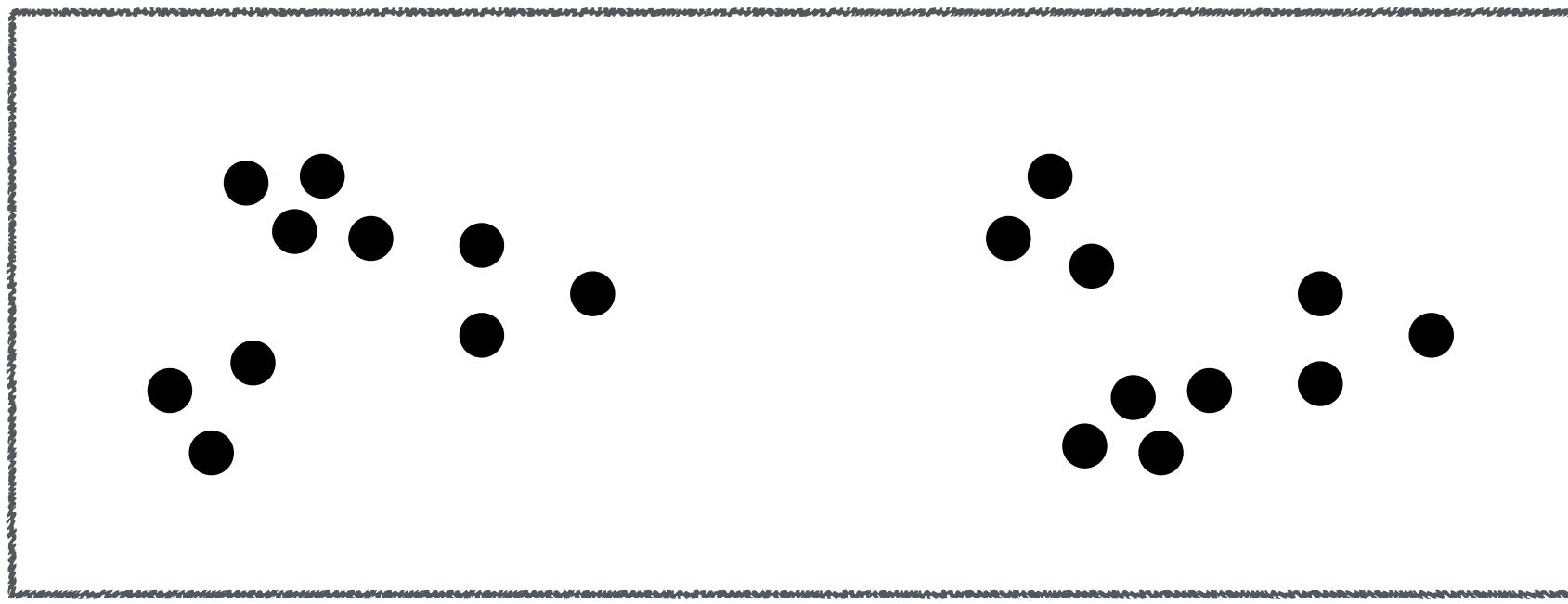
```
import matplotlib.pyplot as plt

plt.scatter(X[:, 0], X[:, 1], c=labels, s=5, cmap='Accent')

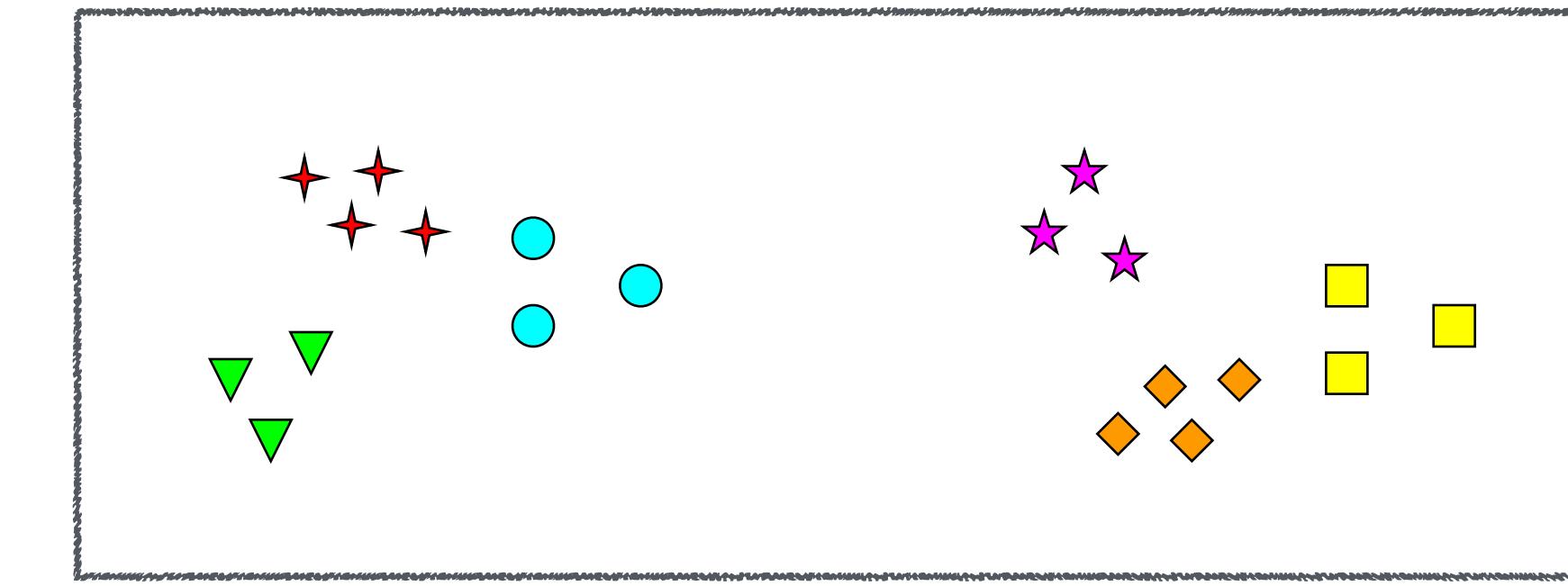
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```



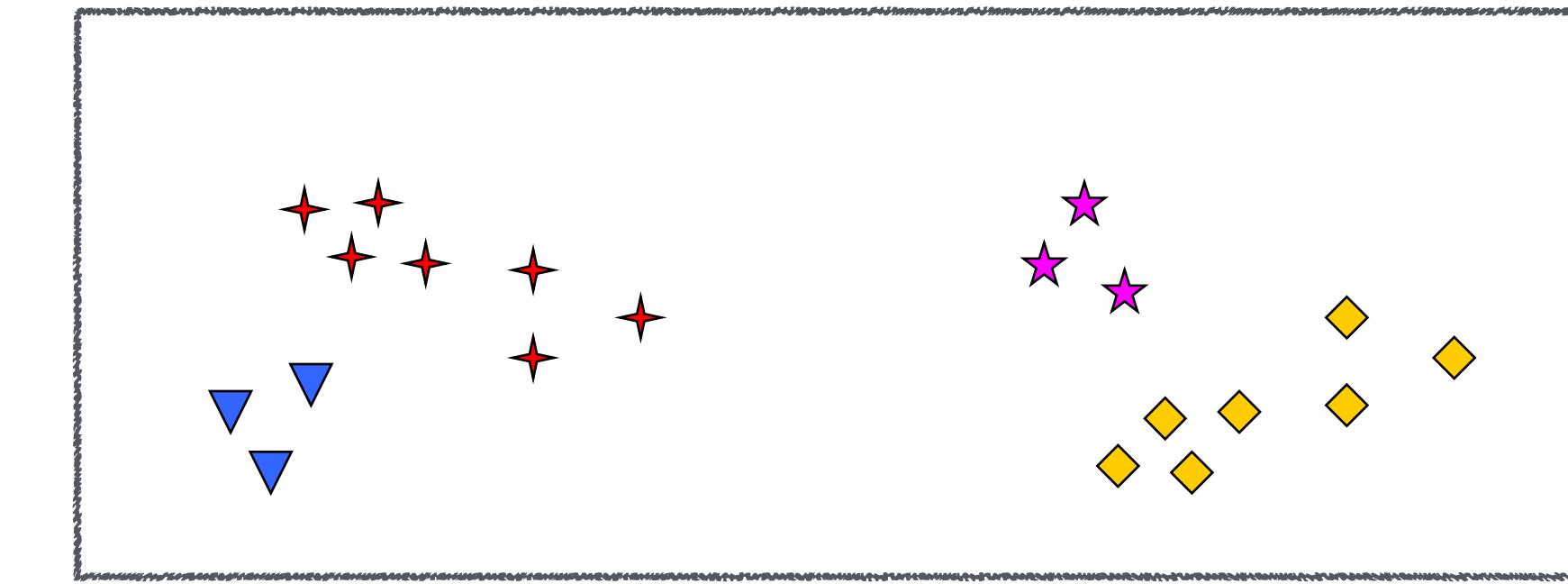
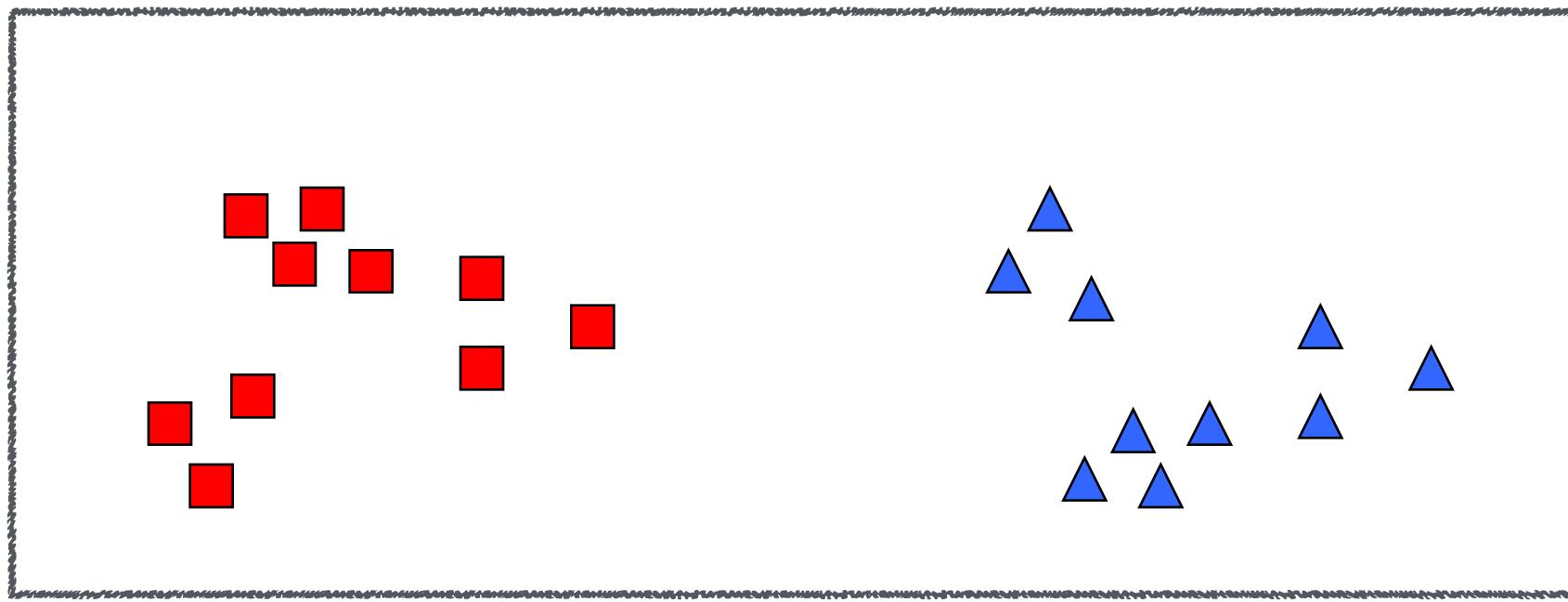
How many clusters ?



Two Clusters



Six Clusters

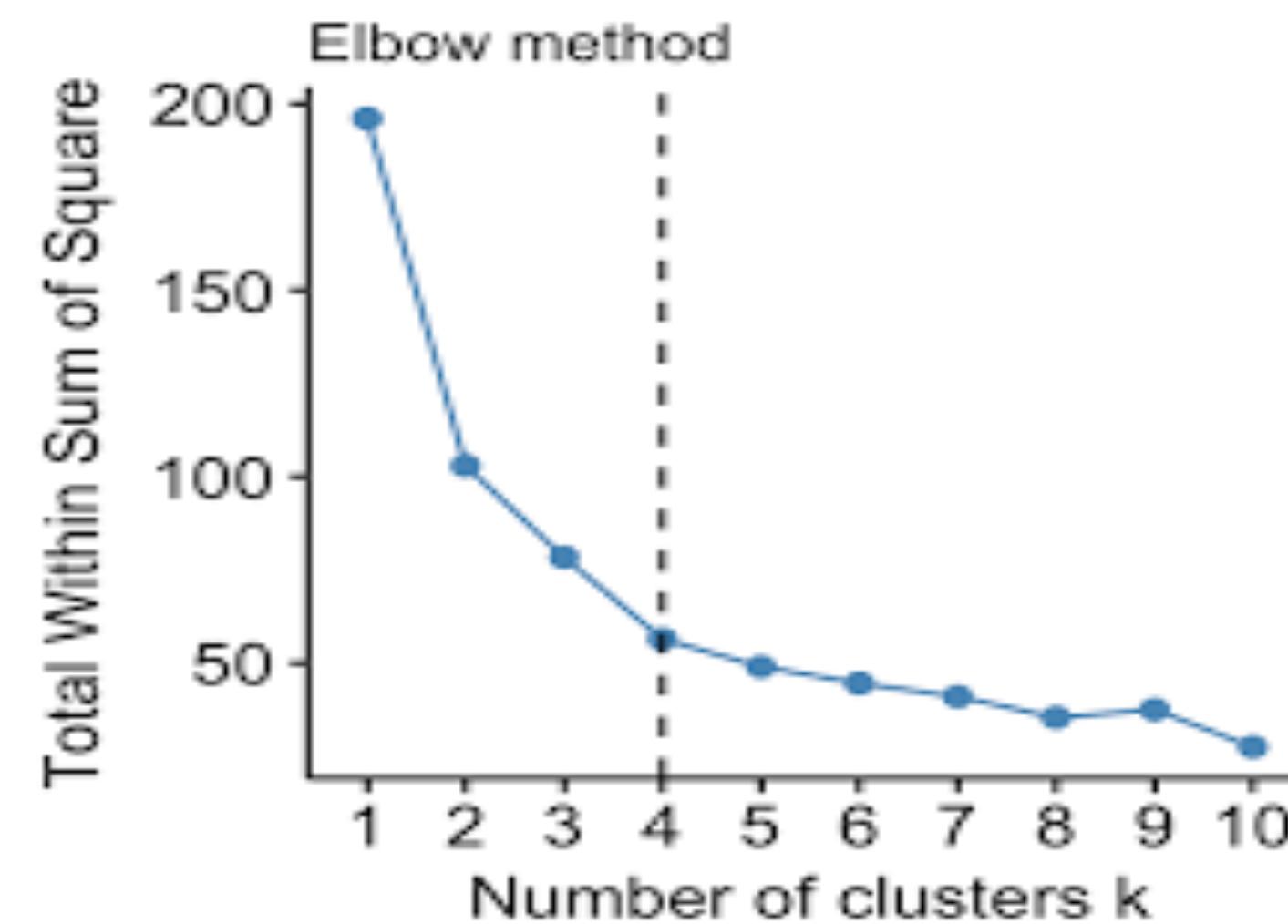
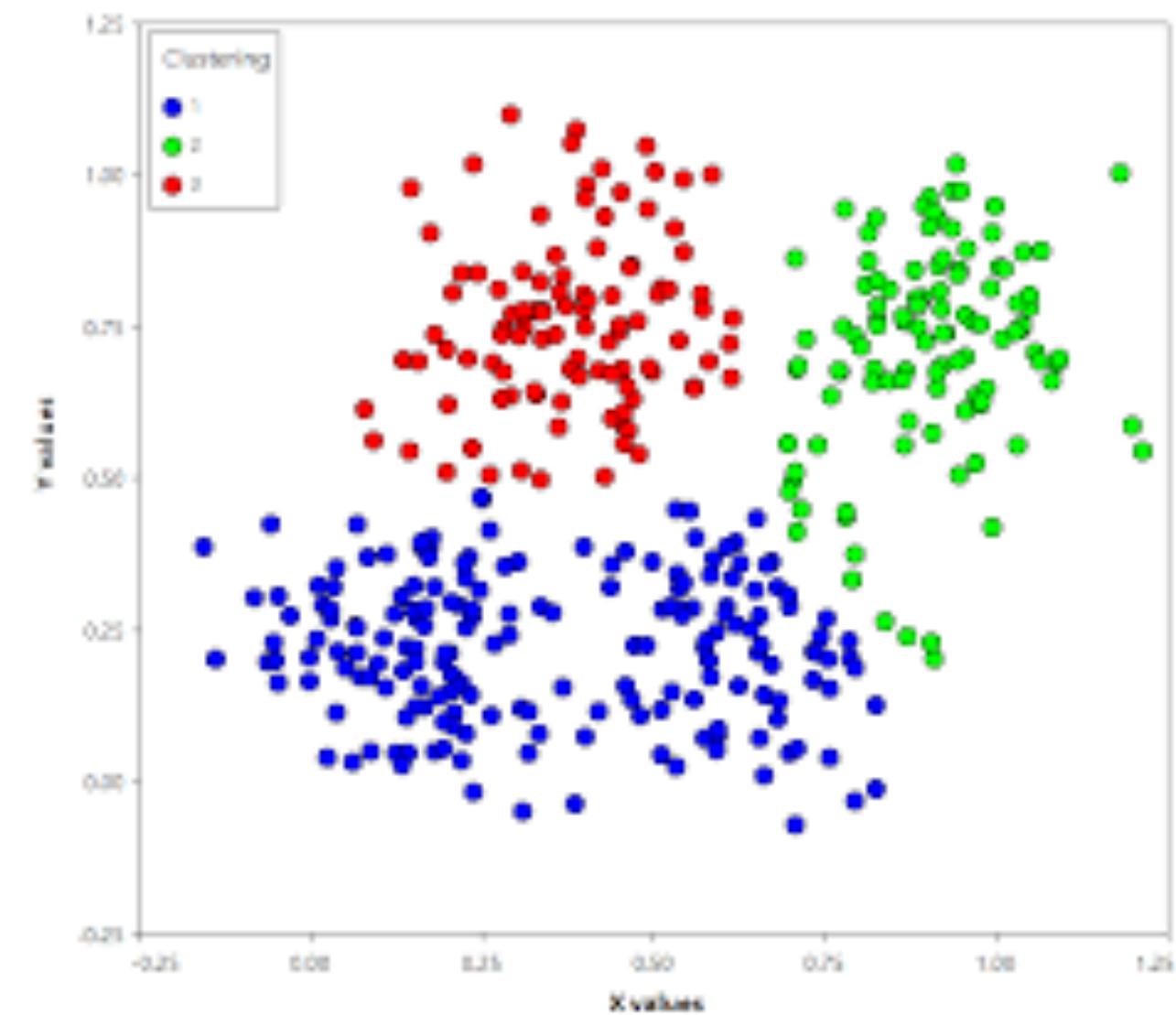


Four Clusters

Choosing the Right K : Elbow Method

Look at the smallest possible K where SSE no longer has significant improvement, i.e., the **knee point**.

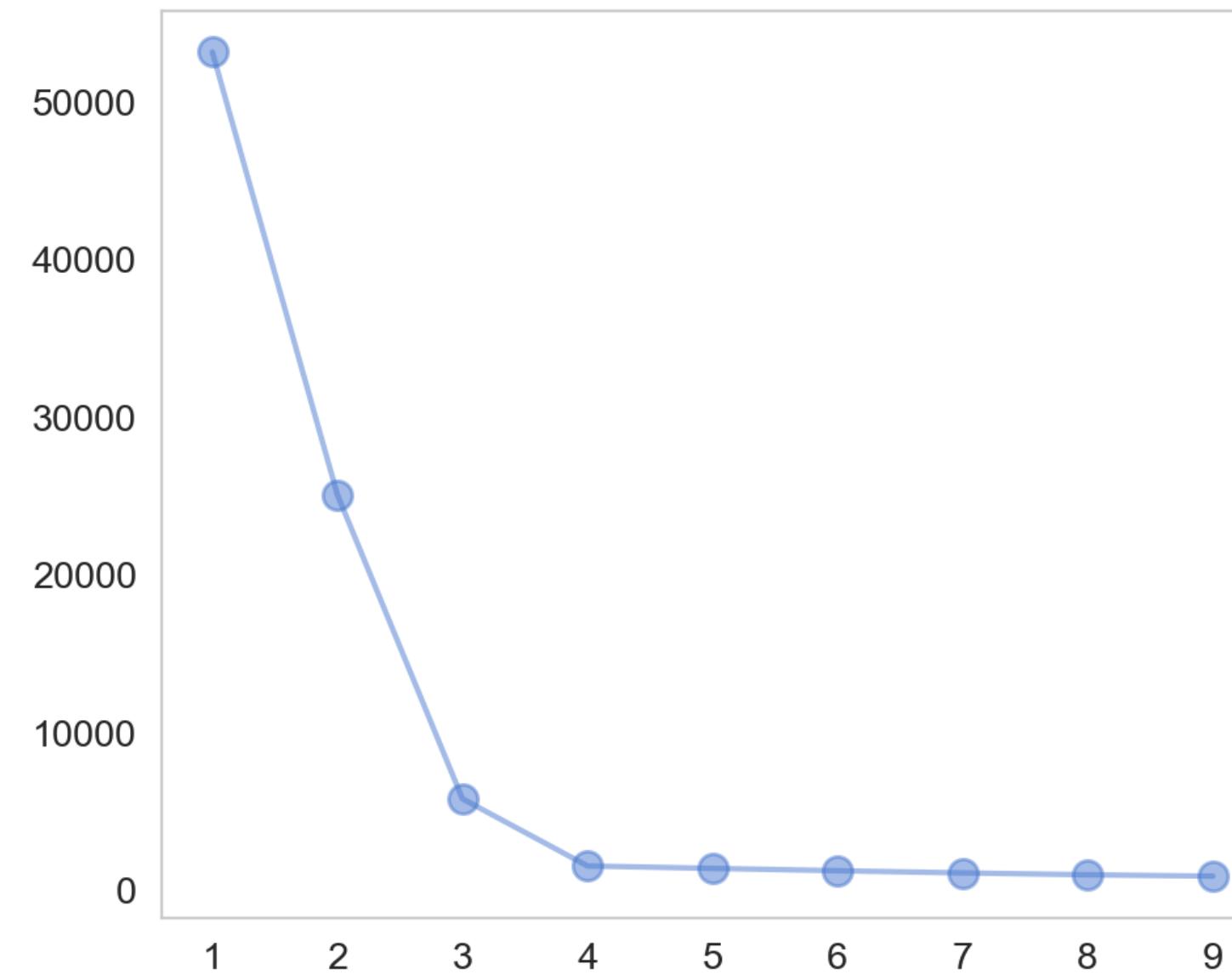
WCSS (Within Cluster Sum Square)



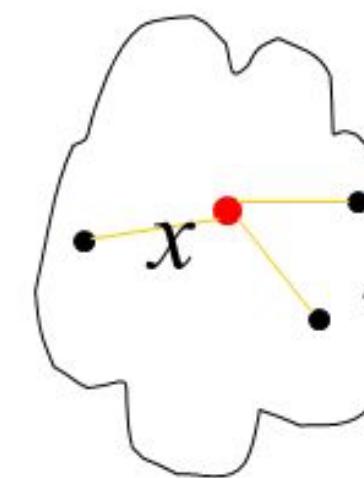


```
sse = {}
for k in range(1, 10):
    kmeans = KMeans(n_clusters=k).fit(X)
    sse[k] = kmeans.inertia_

_,ax=plt.subplots(figsize=(5,4))
ax.plot(list(sse.keys()), list(sse.values()), marker='o', alpha=0.5, ms=8);
plt.tight_layout();
```

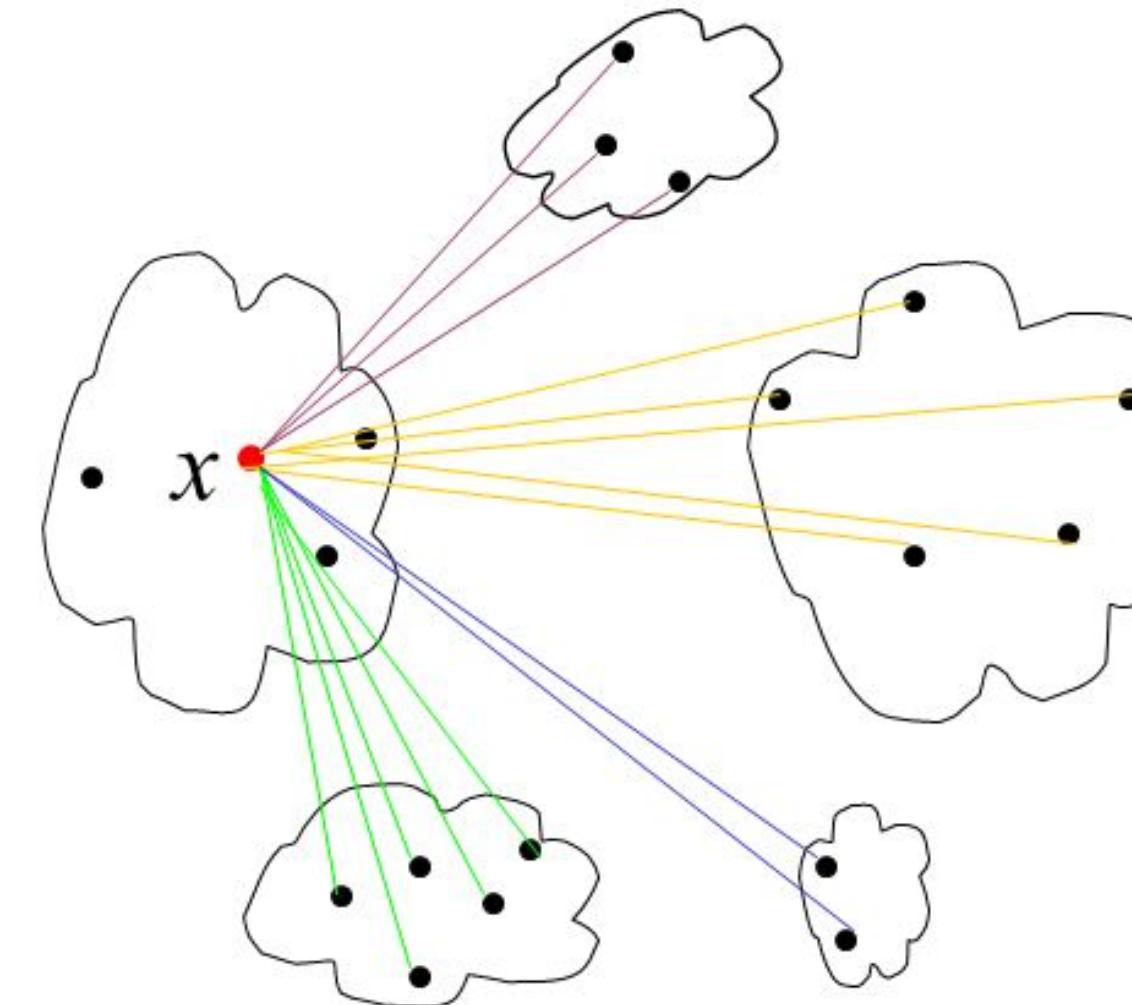


Silhouette Coefficient



Cohesion

a_i : Average distance to
points within cluster
intraclass-distance



Separation *inter-distance*

b_i : Minimum average distance to
points in a different cluster

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} = \begin{cases} 1 - a_i/b_i & \text{if } a_i < b_i \\ 0 & \text{if } a_i = b_i \\ b_i/a_i - 1 & \text{if } a_i > b_i \end{cases}$$

$$\text{Silhouette coefficient (SC)} = n^{-1} \sum_{i=1}^n s_i$$

Calinski Harabasz Index

Also known as *Variance Ratio Criterion* (higher is better).

Ratio of sum of *between-clusters dispersion* and *within-clusters dispersion* for all clusters.

Define

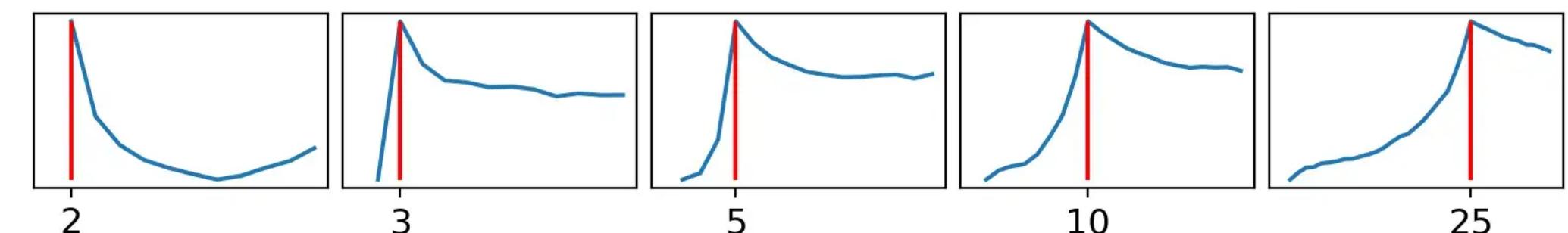
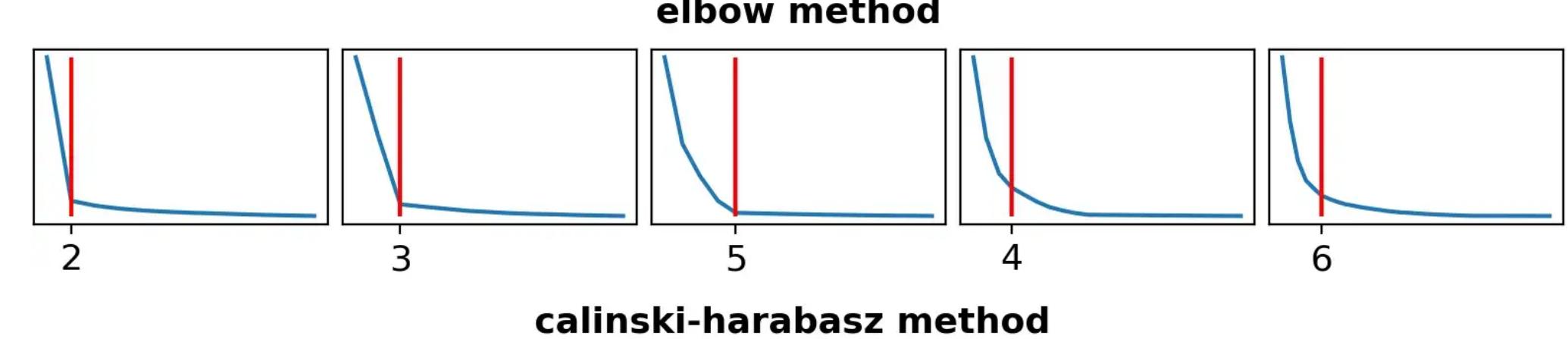
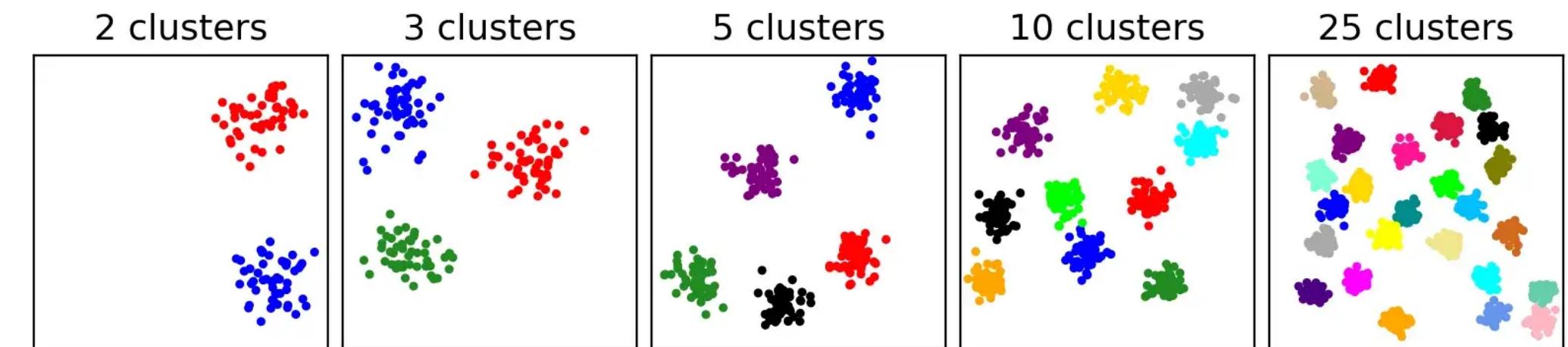
- N data points with *global center* \mathbf{c}_G .
- K clusters, each with N_q data points and center \mathbf{c}_q in cluster q .

The Calinski Harabasz index is given by

$$S = \frac{B_K / (K - 1)}{W_K / (N - K)}$$

$$B_K = \sum_{q=1}^K N_q \|\mathbf{c}_q - \mathbf{c}_G\|^2 \quad \text{Between-cluster dispersion}$$

$$W_K = \sum_{q=1}^K \sum_{x \in C_q} \|x - \mathbf{c}_q\|^2 \quad \text{Within-cluster dispersion}$$

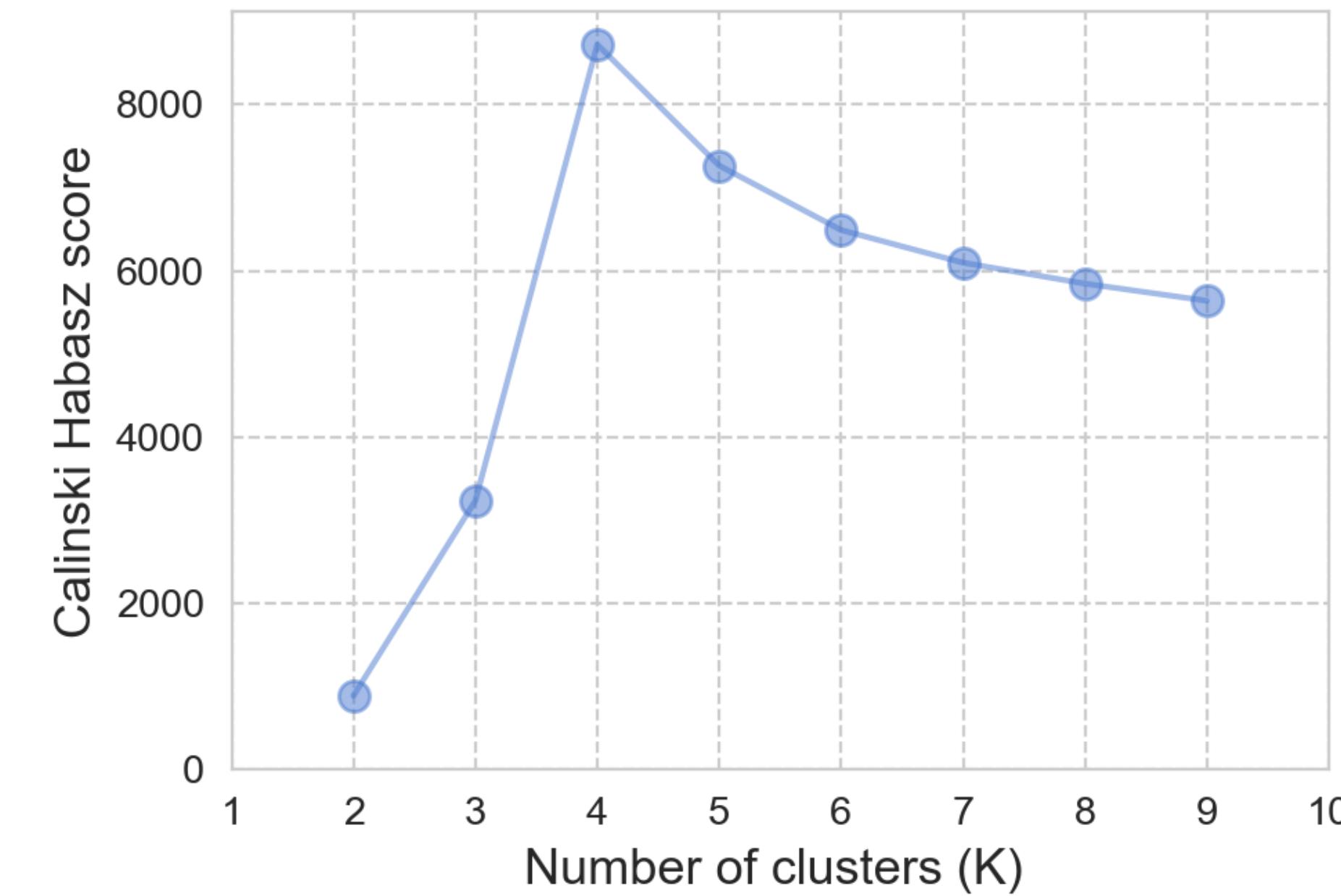
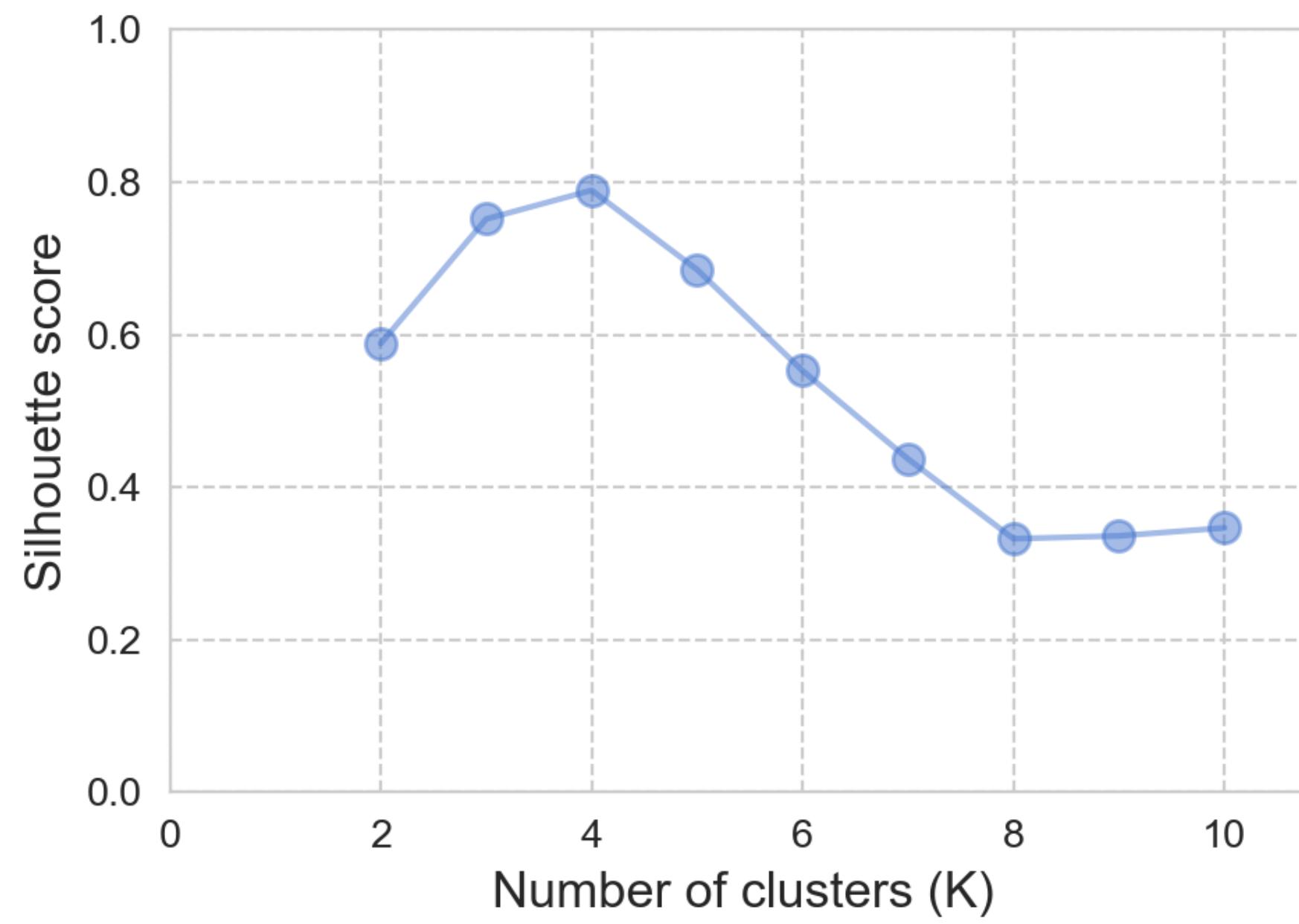




```
from sklearn.metrics import silhouette_score  
silhouette_score(X, labels)
```

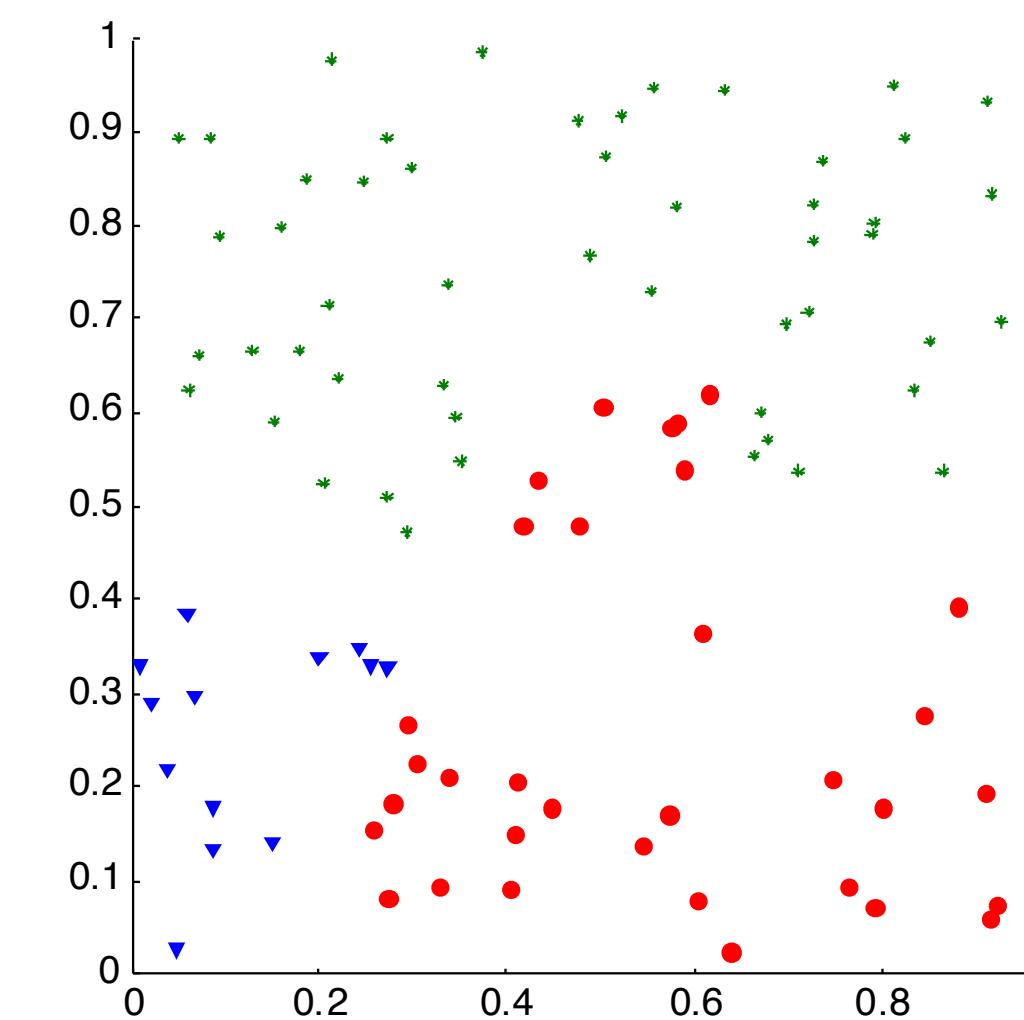
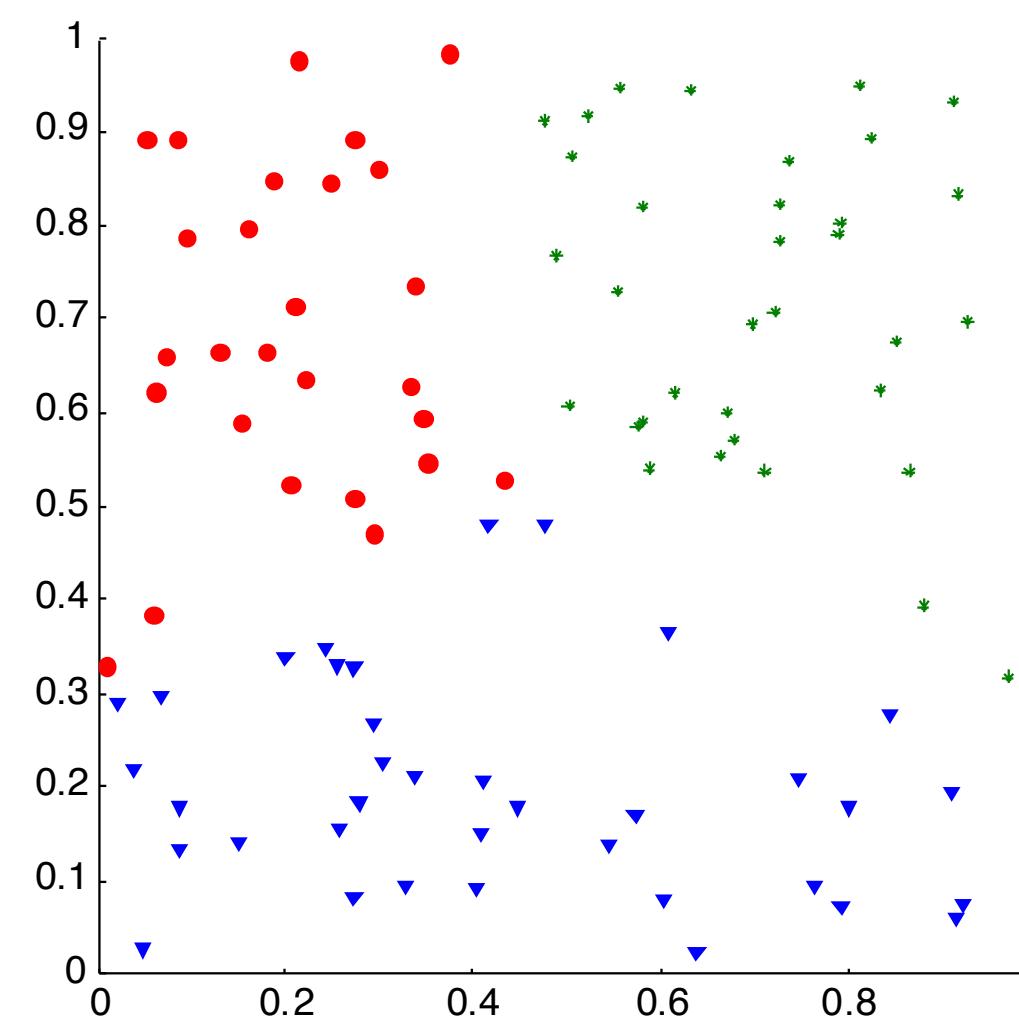
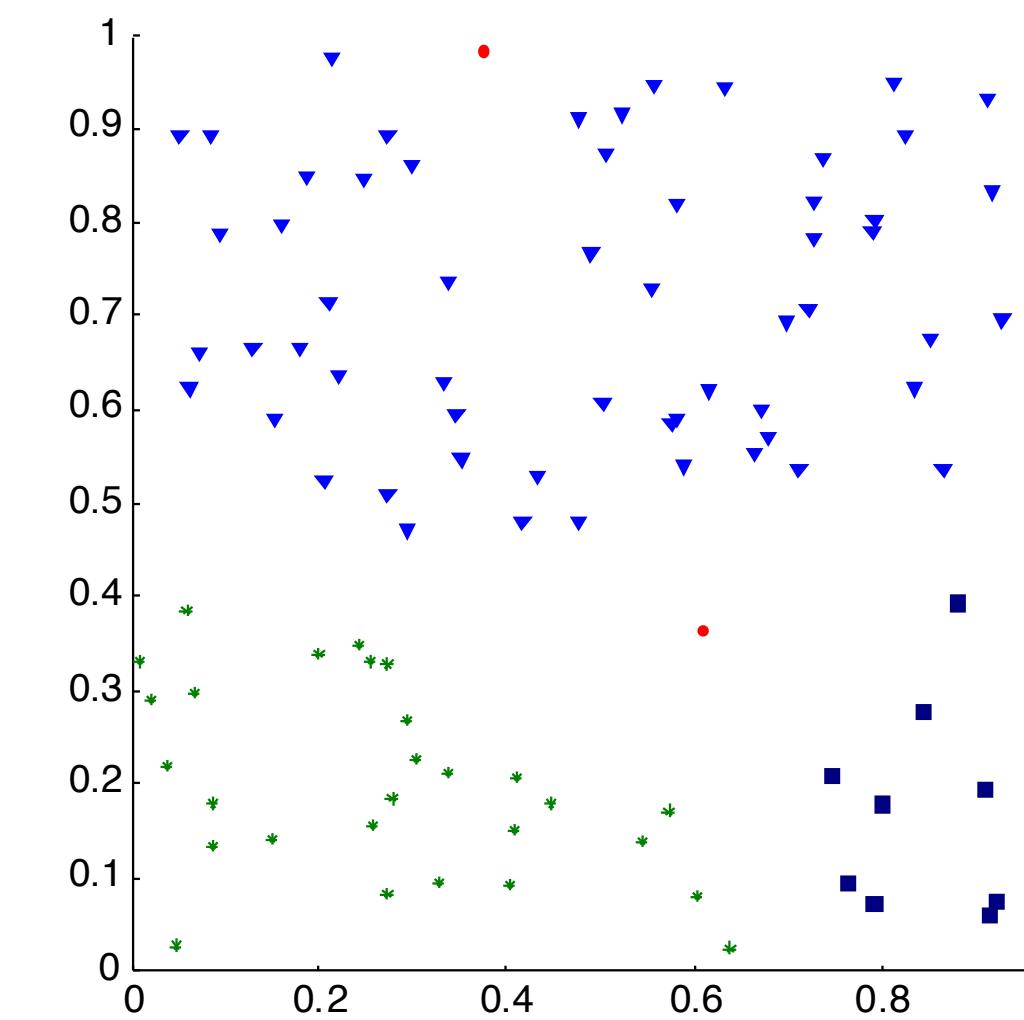
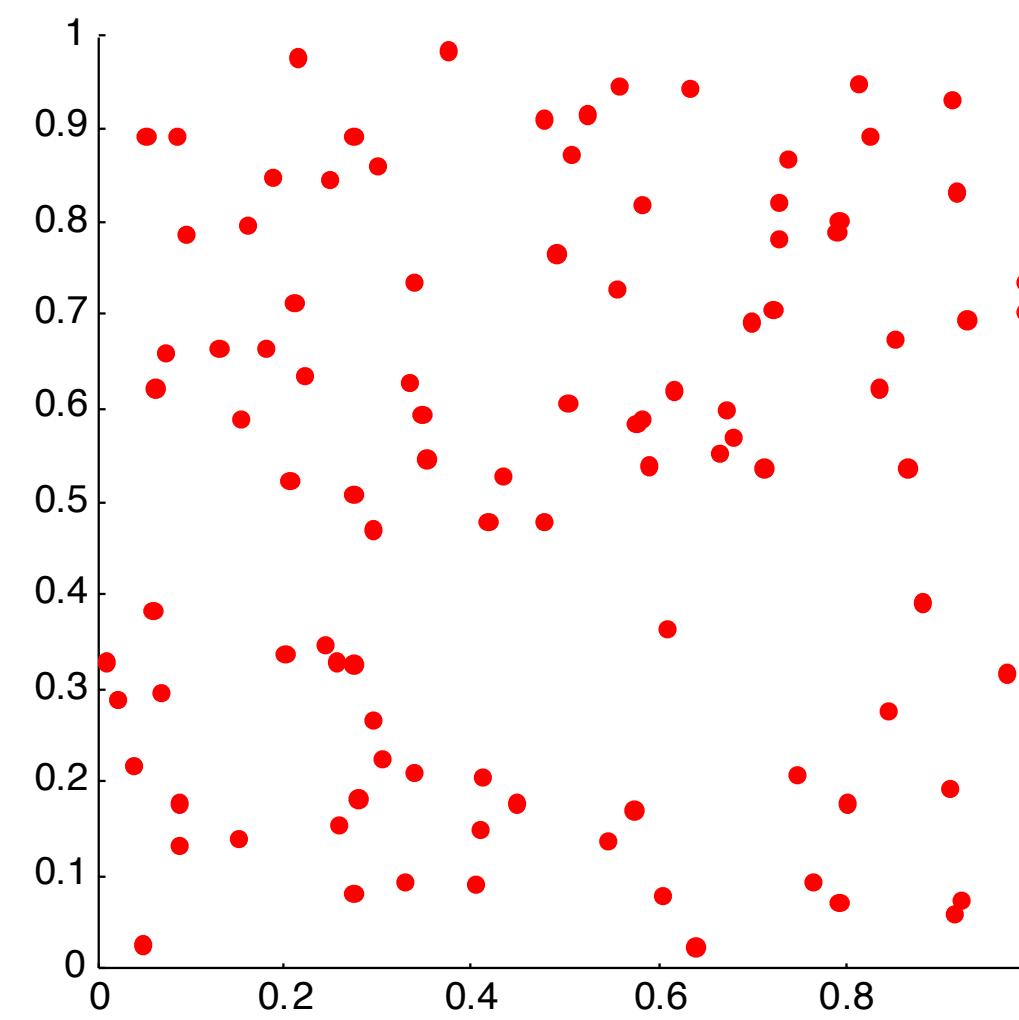
data point cluster labels

```
from sklearn.metrics import calinski_harabasz_score  
calinski_harabasz_score(X, labels)
```



Clusters found in Random Data

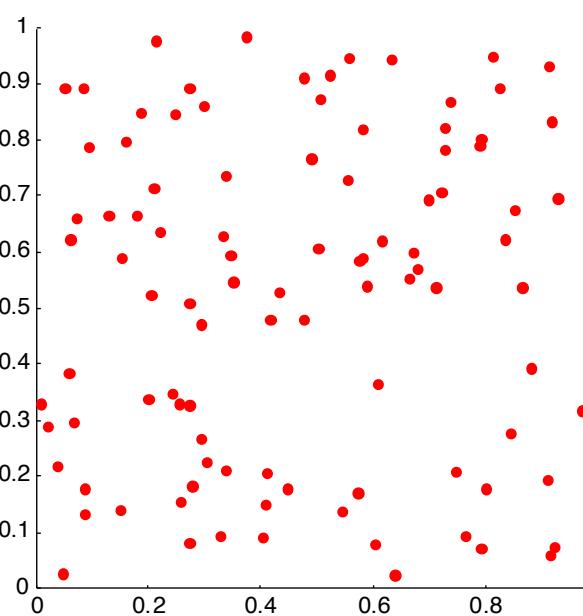
**Random
Points**



Hopkins Statistics

nearest neighbor

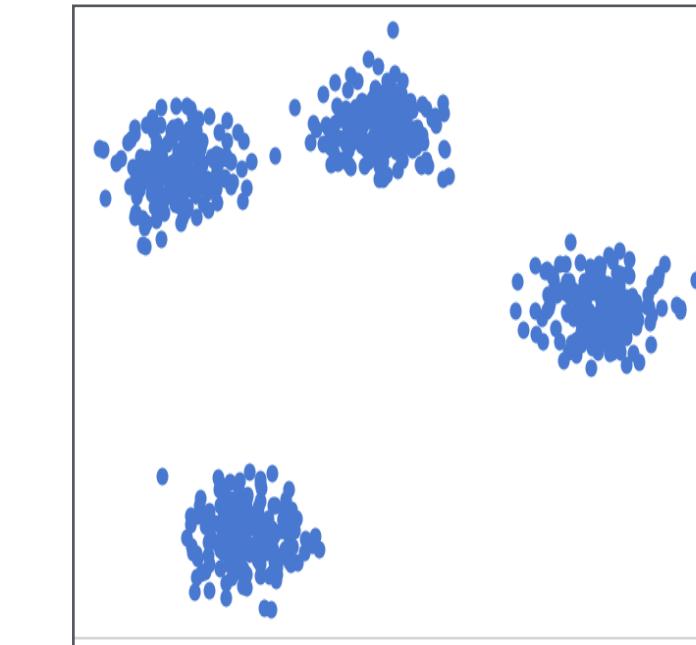
Random data



Sample p points

For each point i ,
find distance u_i to its nearest
neighbor in actual data

Actual data



Sample p points

For each point i ,
find distance w_i to its nearest
neighbor in actual data

$$H = \frac{\sum_{i=1}^p w_i}{\sum_{i=1}^p u_i + \sum_{i=1}^p w_i}$$

环绕着 $\sum_{i=1}^p w_i$ 的红色圆圈注释为 "ระยะทางจากจุด ซึ่งมีลักษณะ NN"

环绕着 u_i 的红色圆圈注释为 "random data"

$u_i \rightarrow 0$

Taking Insights from Clustering

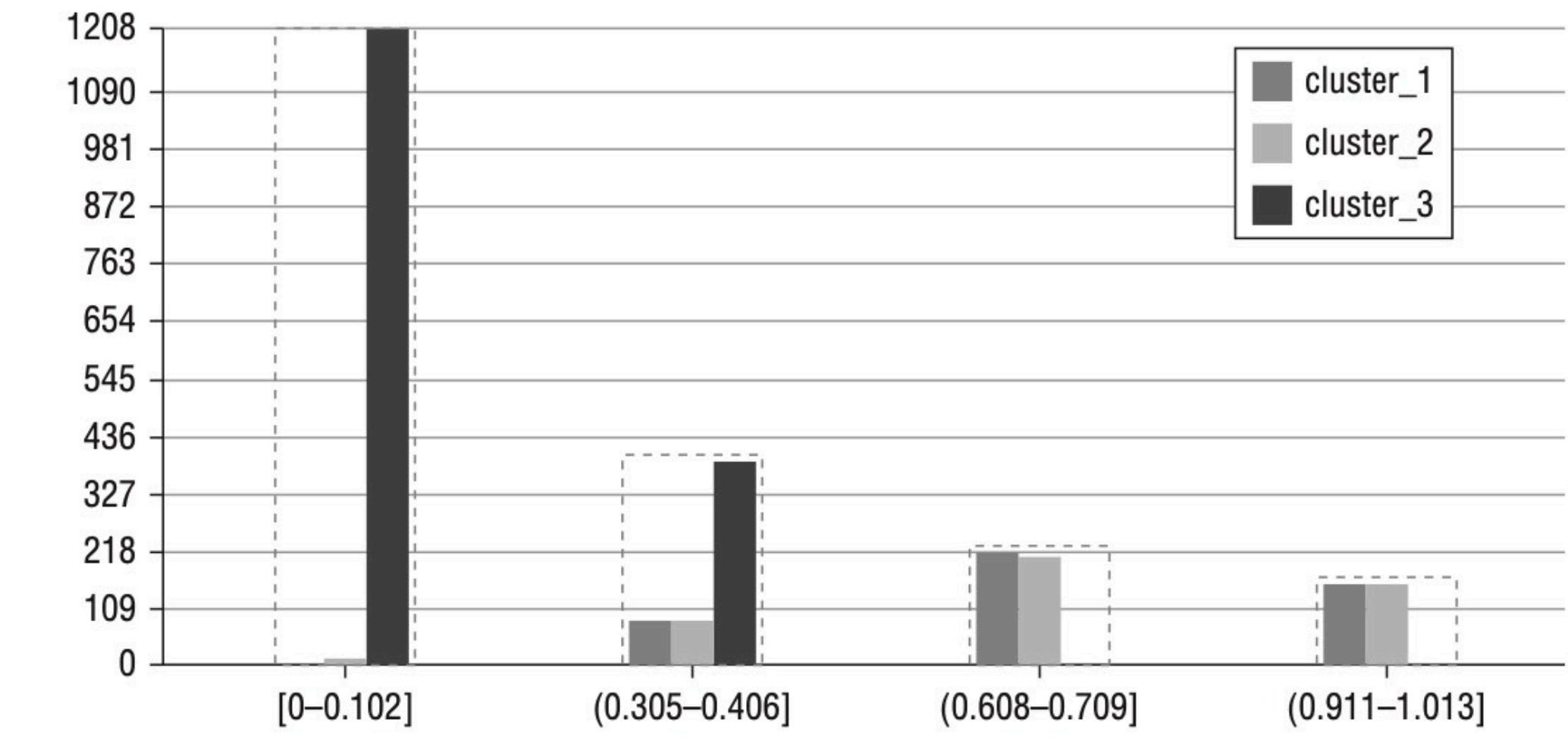
Count of records in each cluster

Summary statistics of (descaled) variables in each cluster (Mean vectors = Centroids)

ANOVA to find which variables have significant difference in means among clusters.

↳ Analysis of variance

VARIABLE	CLUSTER 1	CLUSTER 2	CLUSTER 3	OVERALL
# Records in Cluster	8,538	8,511	30,656	47,705
LASTDATE	0.319	0.304	0.179	0.226
FISTDATE	0.886	0.885	0.908	0.900
RFA_2F	0.711	0.716	0.074	0.303
D_RFA_2A	0.382	0.390	0.300	0.331
E_RFA_2A	0.499	0.500	0.331	0.391
F_RFA_2A	0.369	0.366	0.568	0.496
DOMAIN3	0.449	0.300	0.368	0.370
DOMAIN2	0.300	0.700	0.489	0.493
DOMAIN1	0.515	0.300	0.427	0.420
NGIFTALL_log10	0.384	0.385	0.233	0.287
LASTGIFT_log10	0.348	0.343	0.430	0.400



Identifying Key Variables in Forming Clusters

May not be feasible to examine graphs/tables of every variable in every cluster.

Assign cluster labels to data and build a decision tree to find feature importance.

CLUSTER LABEL	RULE	NUMBER OF RECORDS MATCHING RULE	NUMBER OF RECORDS WITH CLUSTER LABEL IN RULE	TREE ACCURACY
Cluster 1, Rule 1	DOMAIN2 = 0 and RFA_2F > 2	6,807	6,807	100%
Cluster 1, Rule 2	DOMAIN2 = 0 and RFA_2F = 2 and E_RFA_2A = 1	1,262	1,262	100%
Cluster 2, Rule 1	DOMAIN2 = 1 and RFA_2F > 2	6,751	6,751	100%
Cluster 2, Rule 2	DOMAIN2 = 1 and RFA_2F = 2 and E_RFA_2A = 1	1,304	1,304	100%
Cluster 3, Rule 1	RFA_2F = 1 and E_RFA_2A = 0	21,415	21,415	100%
Cluster 3, Rule 2	RFA_2F = 1 and E_RFA_2A = 1	2,432	2,411	99%
Cluster 3, Rule 3	RFA_2F = 2 and E_RFA_2A = 0 and F_RFA_2A = 1	5,453	5,344	98%

Identifying Cluster Prototypes

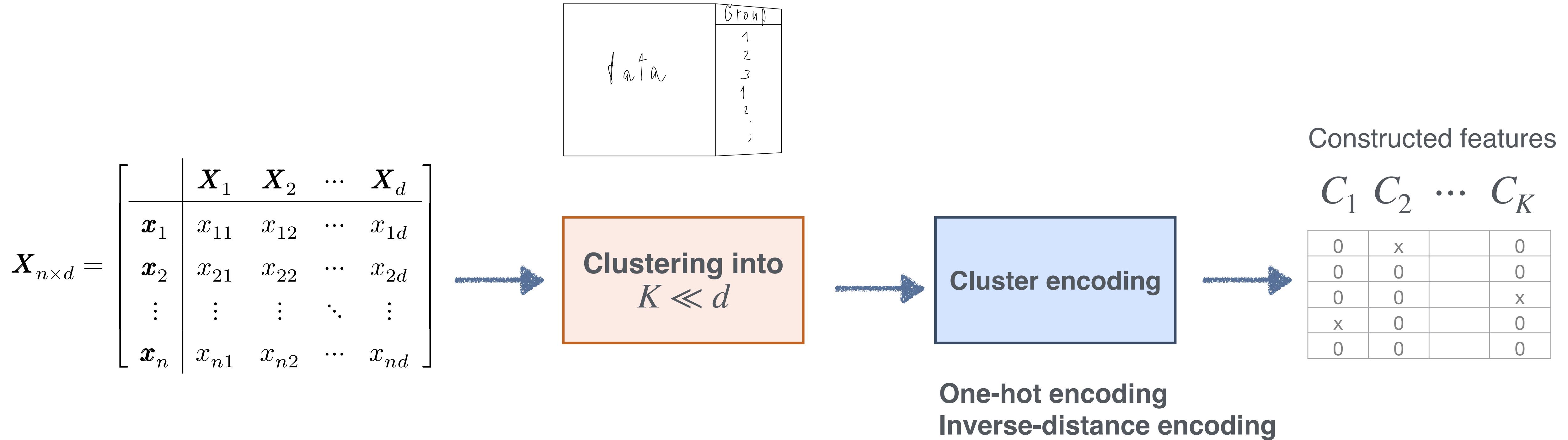
Representative record (in natural units) of each cluster (one with closest distance to centroid)

Helpful in gaining additional insight into the meaning of each cluster.

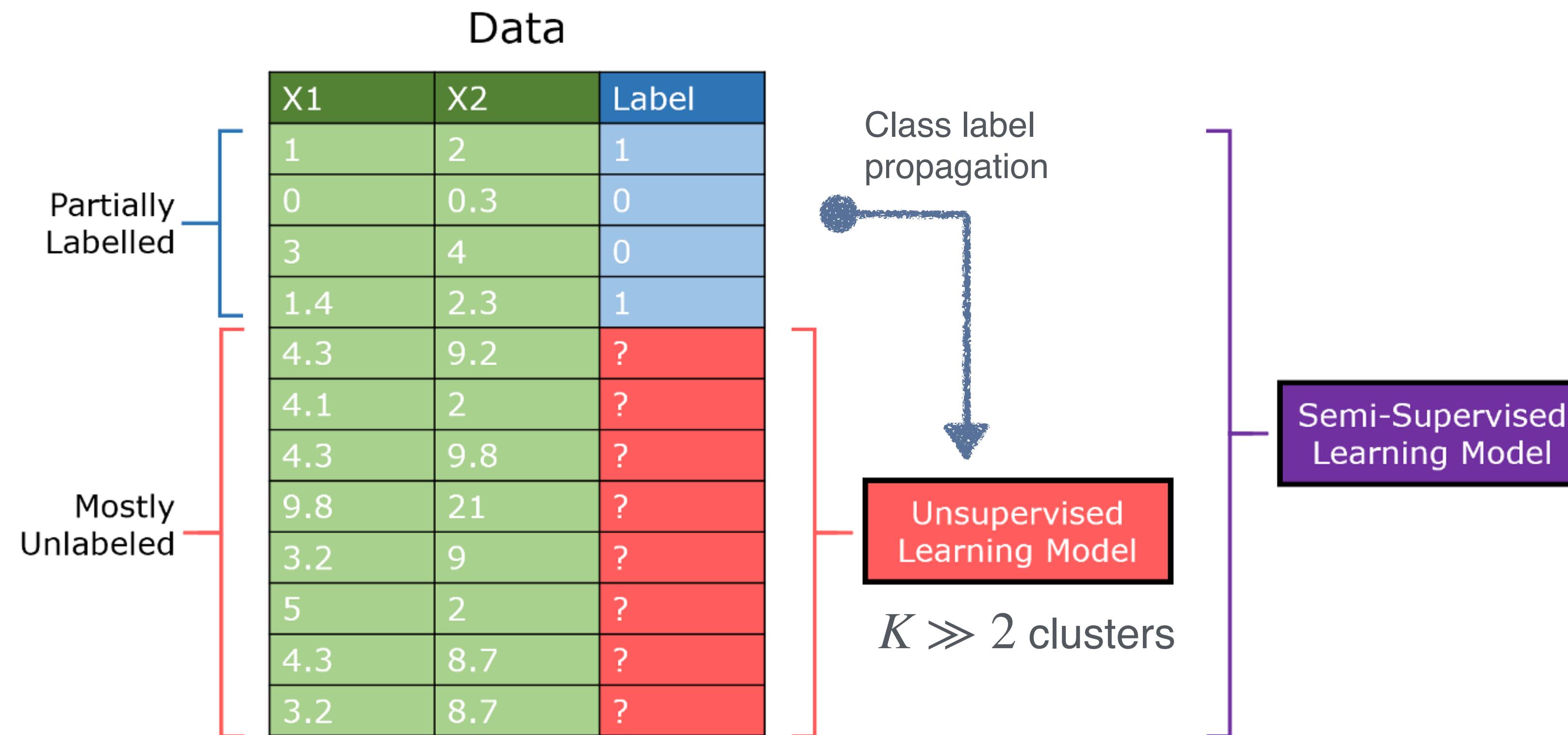
Can examine 2nd or 3rd smallest distance as well.

FIELD	CLUSTER 1, PROTOTYPE 1	CLUSTER 2, PROTOTYPE 1	CLUSTER 3, PROTOTYPE 1
Distance	0.375	0.301	0.313
LASTDATE	9601	9601	9512
FISTDATE	9111	9109	9203
RFA_2F	3	3	1
D_RFA_2A	0	0	0
E_RFA_2A	1	1	0
F_RFA_2A	0	0	1
DOMAIN3	0	0	0
DOMAIN2	0	1	0
DOMAIN1	1	0	0
NGIFTALL	10	10	19
LASTGIFT	12	11	5

Clustering App: Feature Augmentation

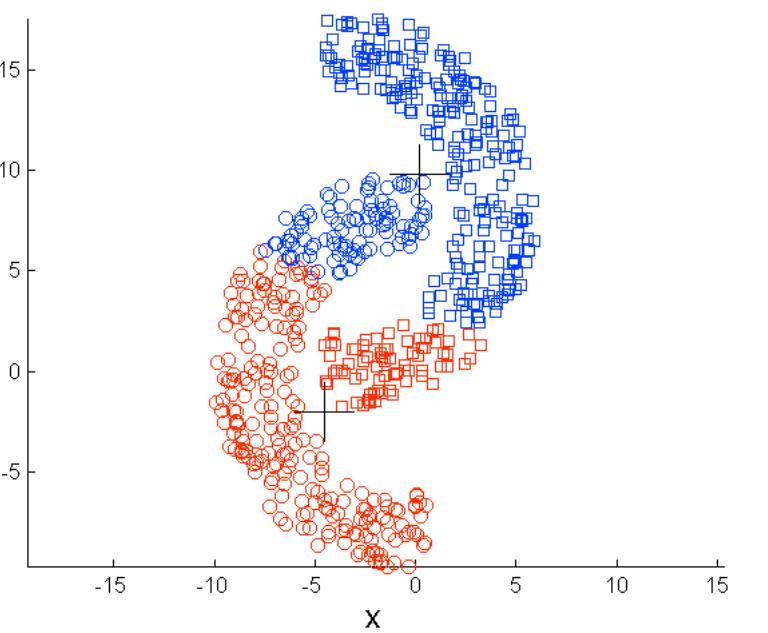


Clustering App: Semi-Supervised Learning

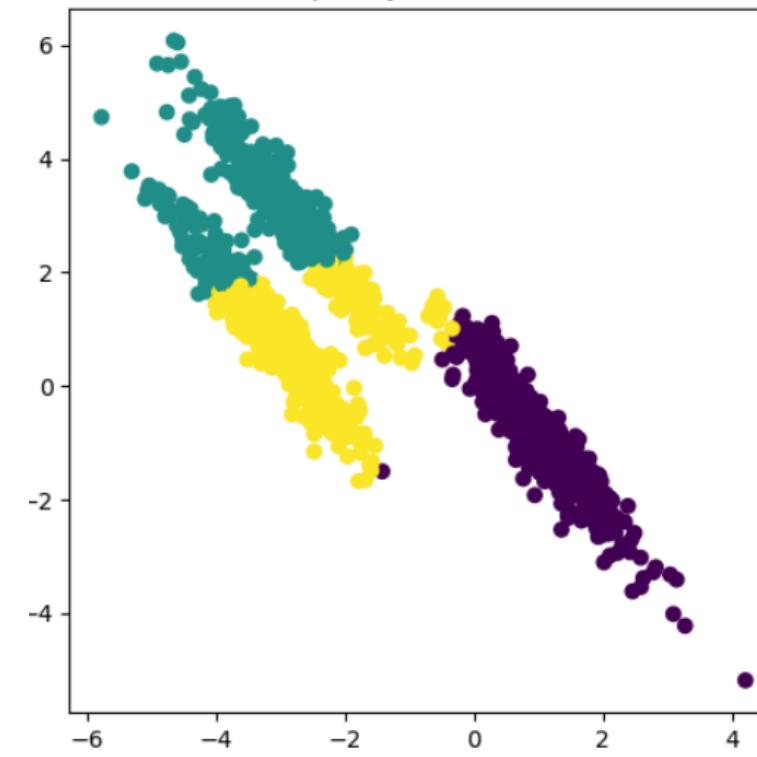


Limitations of K-Means

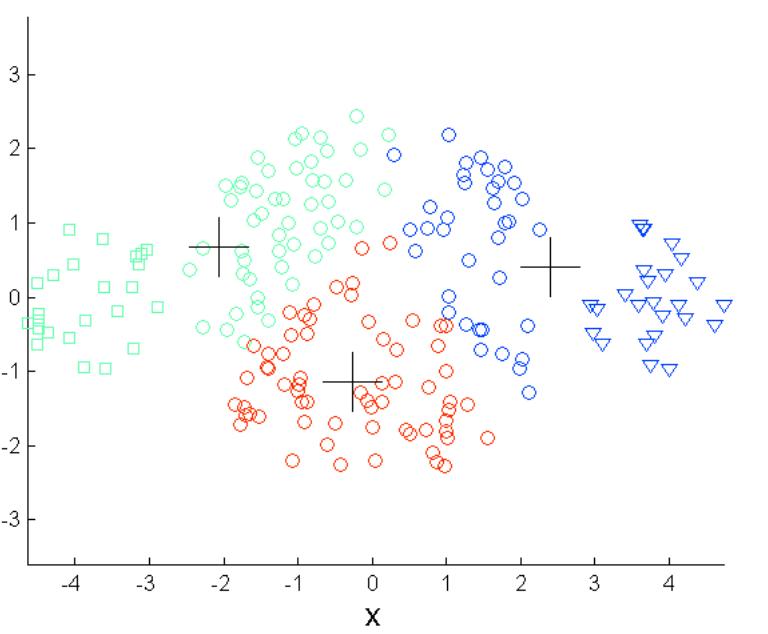
Non-convex shapes



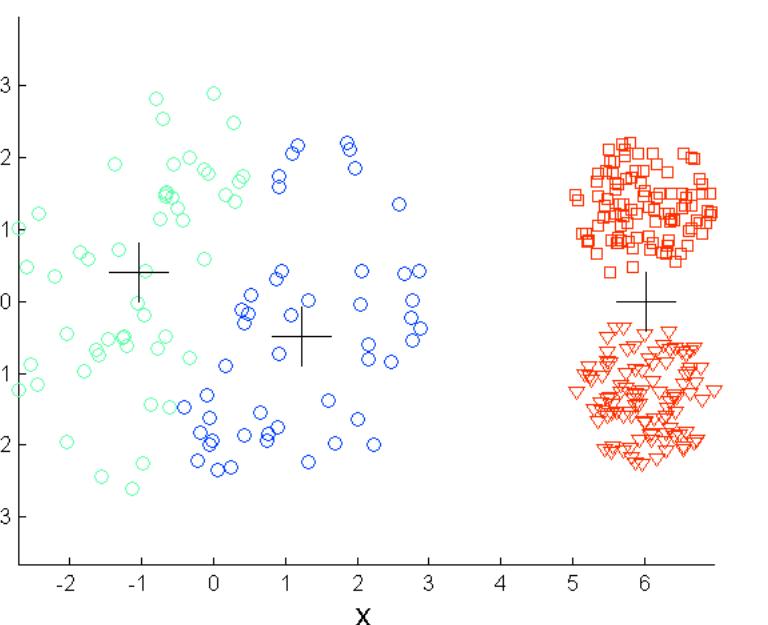
Non-spherical shapes



Different sizes



Different densities
(Unequal variances)



Summary

Clustering to find natural groupings of data points

Require the calculation of dissimilarity among objects.

K-means clustering

- ◆ Approximate clusterings that minimize SSE between objects and their centroids.
- ◆ Elbow method or Silhouette coefficients to determine the number of clusters

Need to interpret clustering results

1. From a one-dimensional data set {2,4,10,12,3,20}, if we apply K-means clustering algorithm using K=2, and that we randomly pick the initial centroid as mu1=1, mu2 = 6. What is the centroid of the 1st cluster in the next iteration ?

(1 Point) *

2.5

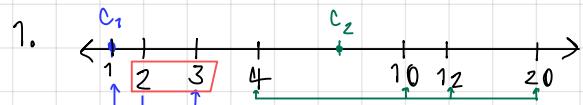
2. From the previous question, what is the centroid of the 2nd cluster in the next iteration ?

(1 Point) *

11.5

3. ถ้าเรามี cluster ของข้อมูลเป็น {(5,4), (3,1), (1,1)} ค่า SSE ของ cluster มีค่าเท่าใด
(1 Point) *

14

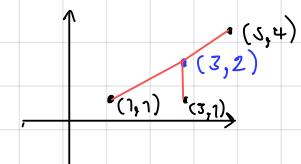


$$\text{new centroid} = \frac{2+3}{2} = 2.5$$

$$c_2 = \frac{4+10+12+20}{4} = 11.5$$

$$3. \text{ centroid } (\mu) = \left(\frac{5+3+1}{3}, \frac{4+1+1}{3} \right) = (3, 2)$$

$$\therefore \text{SSE} = (2^2 + 2^2) + (0^2 + 1^2) + (2^2 + 1^2) = 14$$





Clustering Lab

(Group of 4 to 6)
