Machine Learning

Amin Golzari Oskouei

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei

Azarbaijan Shahid Madani University
2023

Unsupervised Learning : Clustering

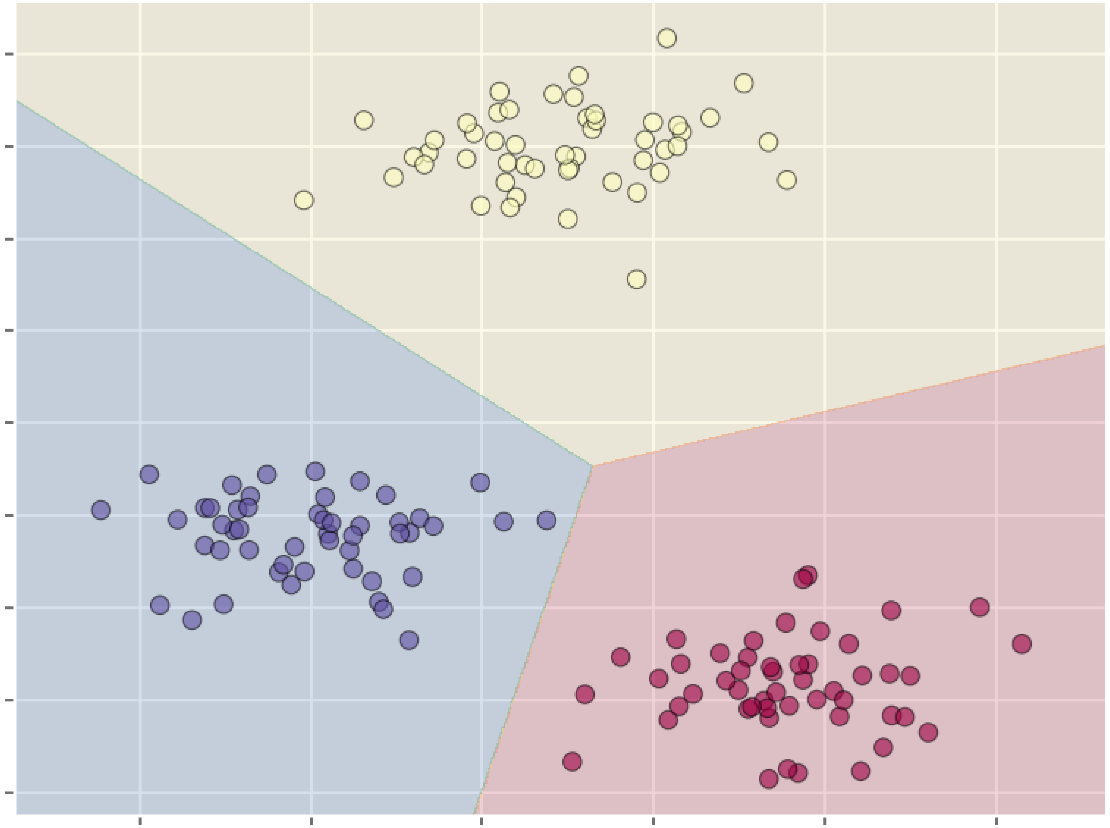
Table of Contents

Reminder: Supervised learning

- Supervised Learning. For each example, the correct answer is given.

$$\boxed{\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\cdots,(x^{(m)},y^{(m)})\}}$$

Training set



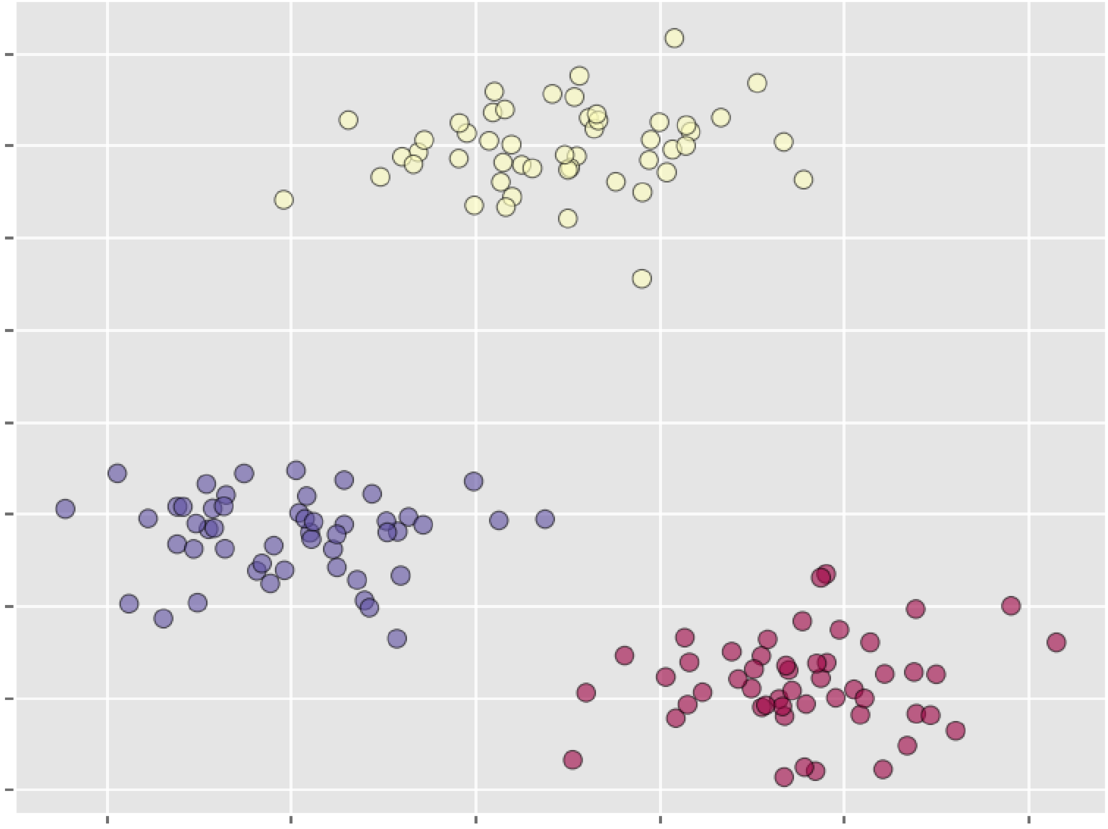
- Types Of Supervised Learning.
  - Regression: Estimation of a continuous quantity.
  - Classification: Estimation of a discrete quantity.

Unsupervised Learning

- Unsupervised Learning. Not knowing the correct answers.

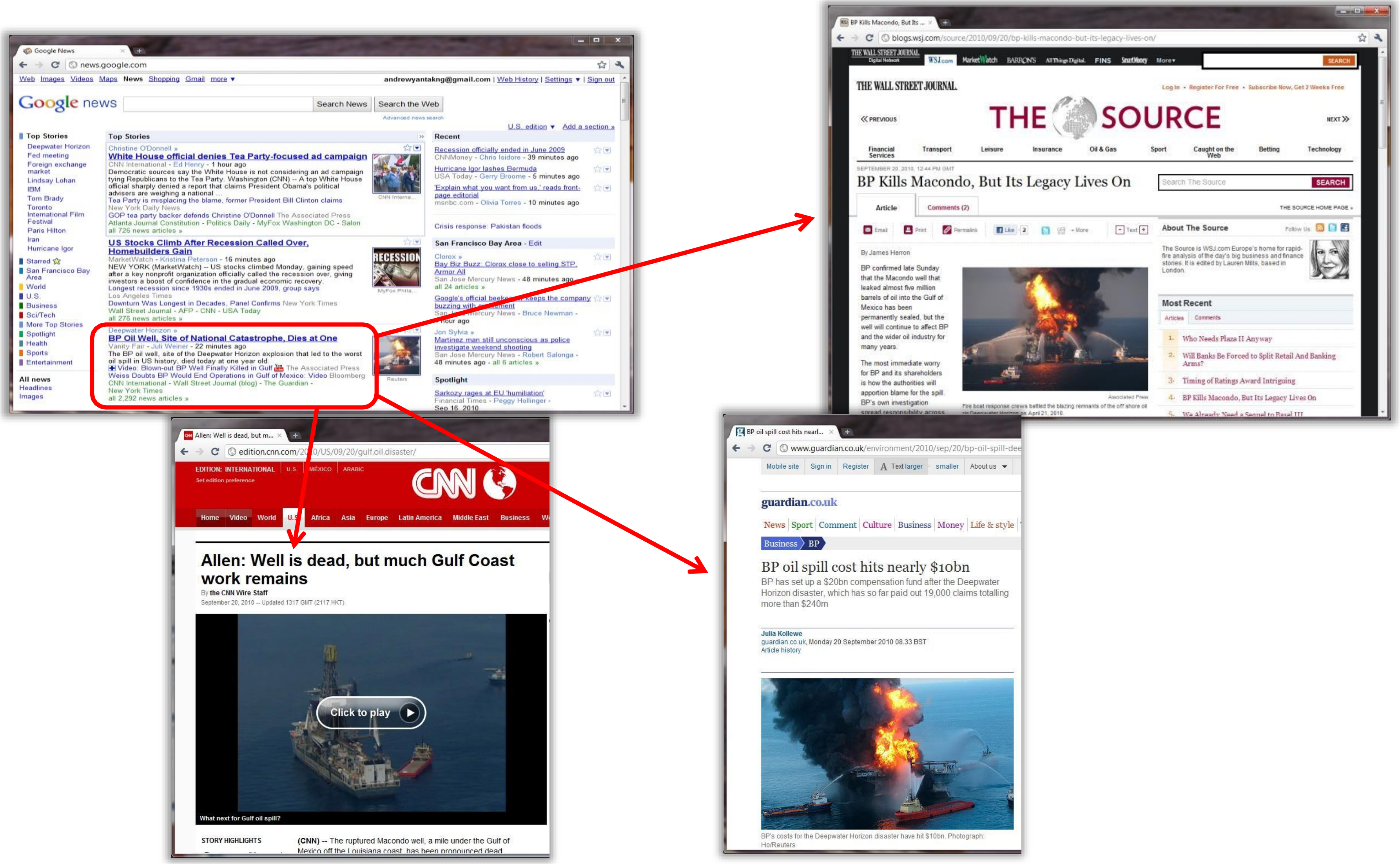
$$\boxed{\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(m)}\}}$$

Training set



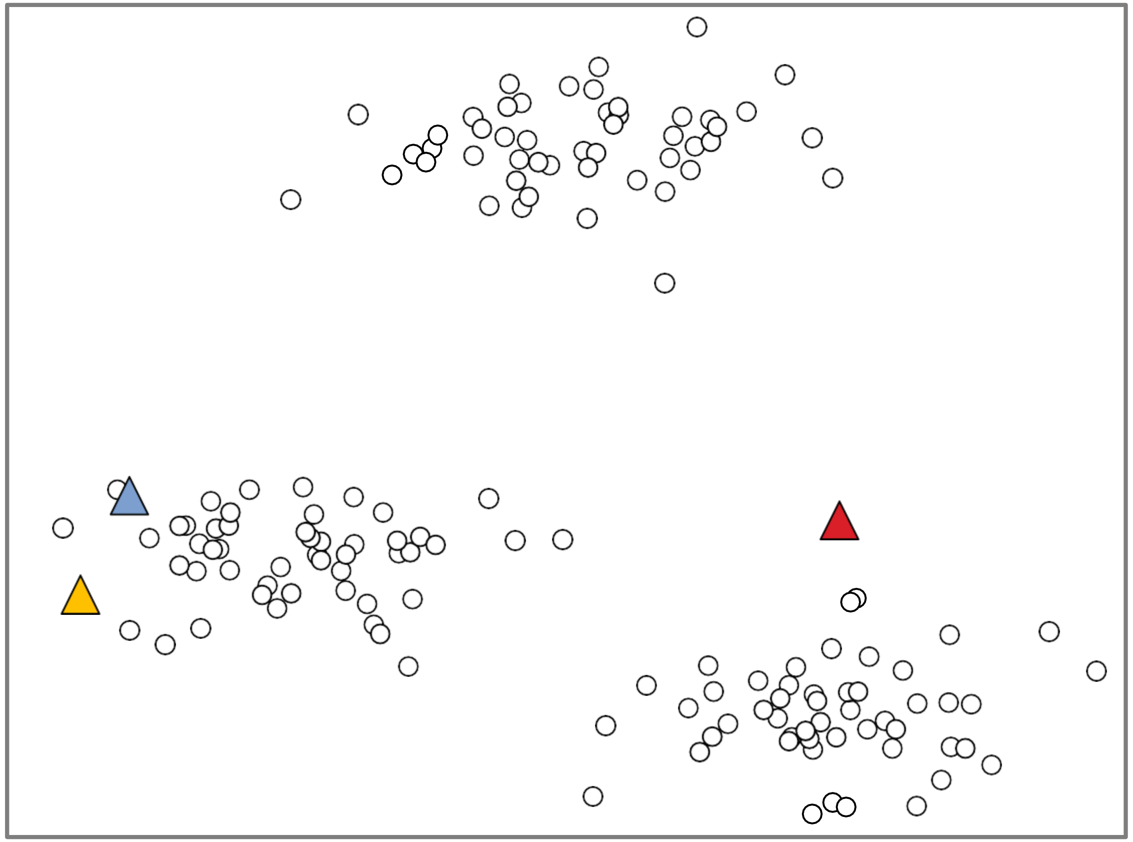
- Goal. Structure detection in input data

Applications of clustering: grouping related news



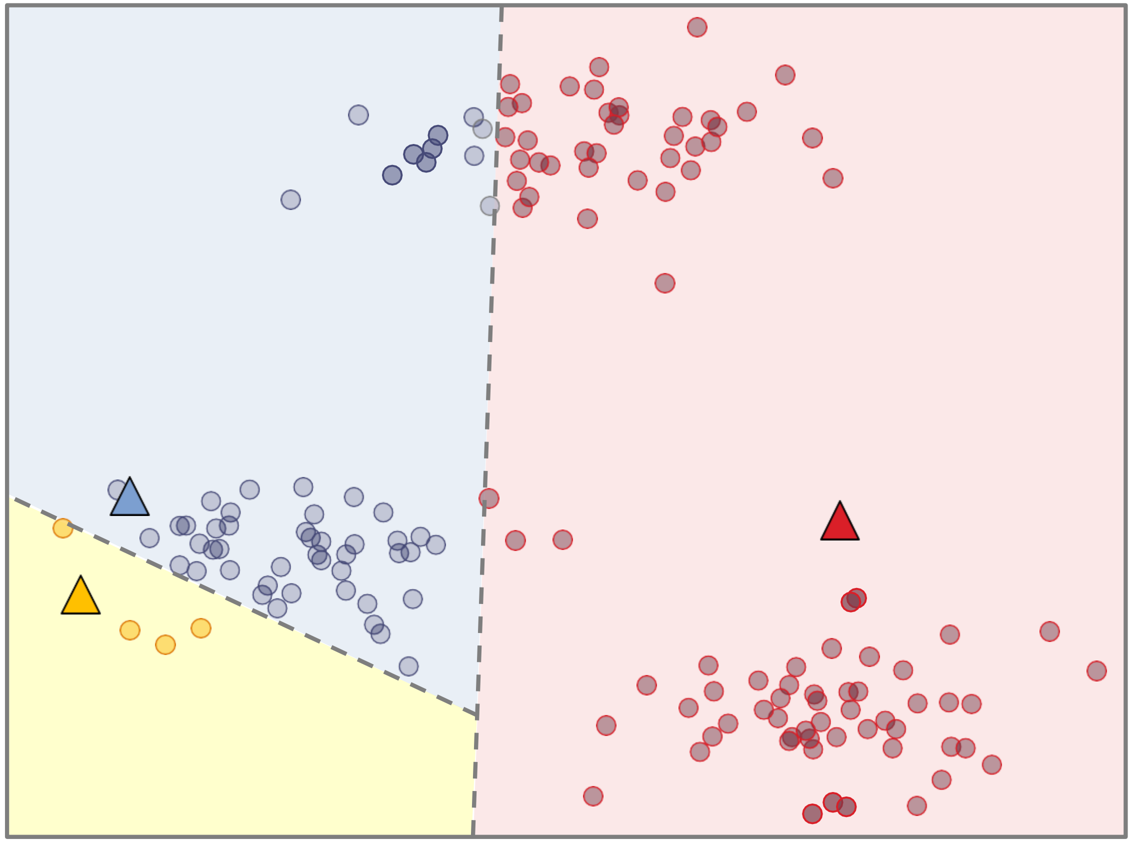
K-means Clustering Algorithm

- Its an iterative clustering algorithm
  - It choosing K points randomly as the center of the clusters.
  - Repeat the following steps:
    - Assign each data to a cluster with the closest center.
    - Update the centroid of each cluster by averaging the data assigned to that cluster
  - Stop time : When no data changes its cluster in an iteration.



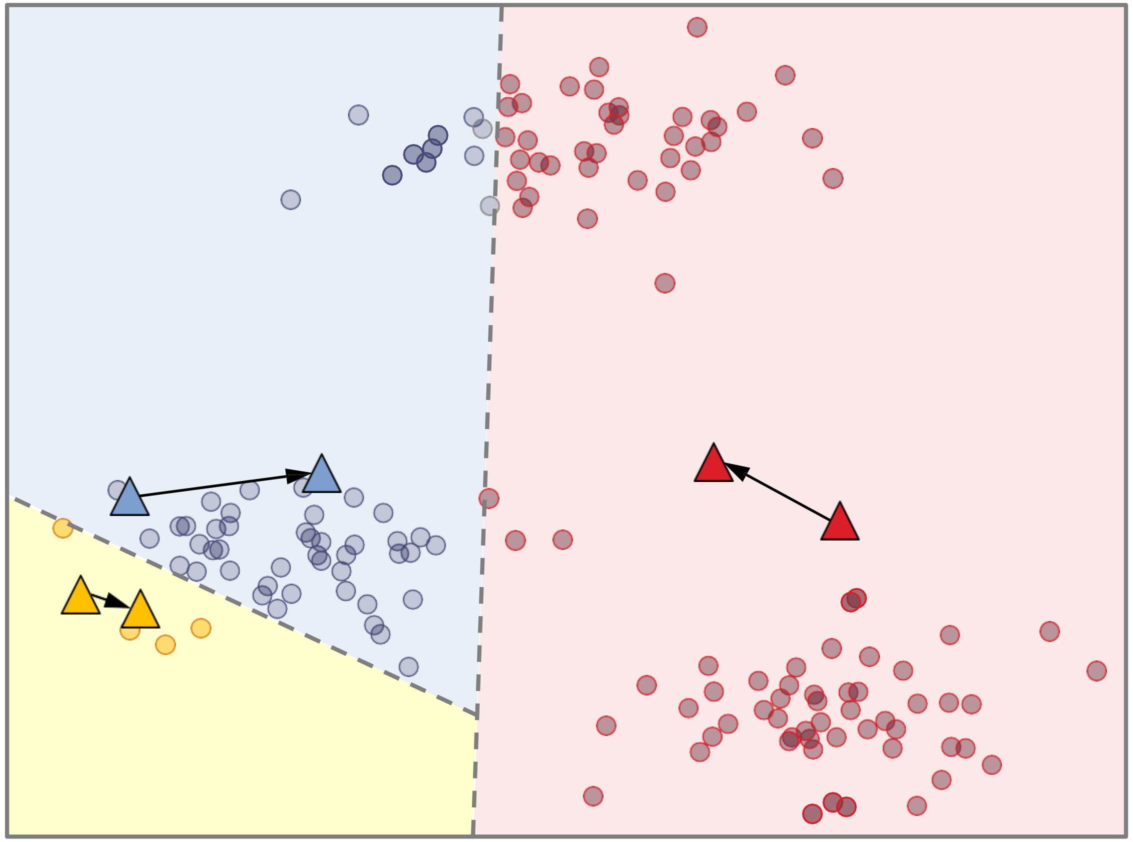
K-means Clustering Algorithm

- Its an iterative clustering algorithm
  - It choosing K points randomly as the center of the clusters.
  - Repeat the following steps:
    - Assign each data to a cluster with the closest center.
    - Update the centroid of each cluster by averaging the data assigned to that cluster
  - Stop time : When no data changes its cluster in an iteration.

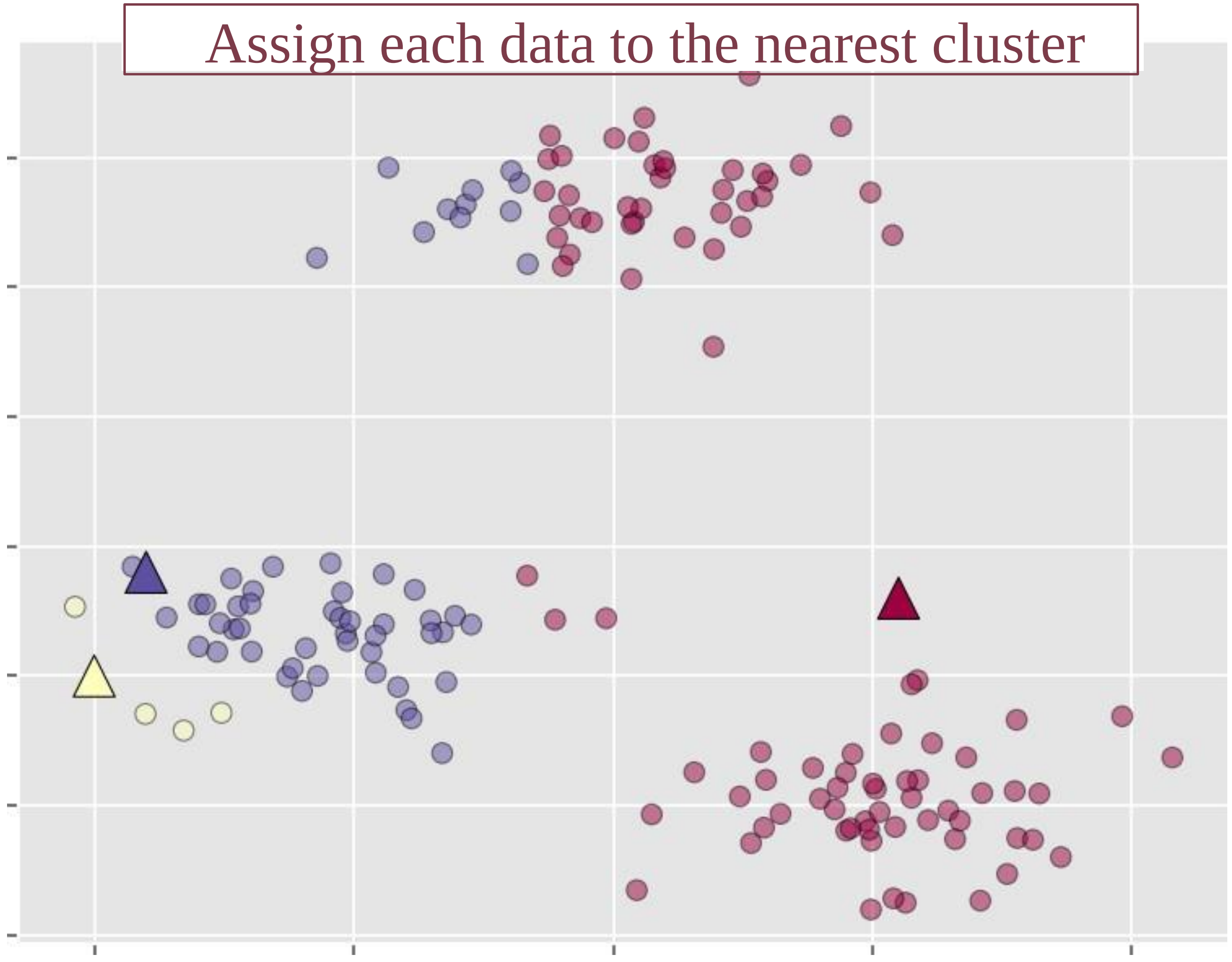


K-means Clustering Algorithm

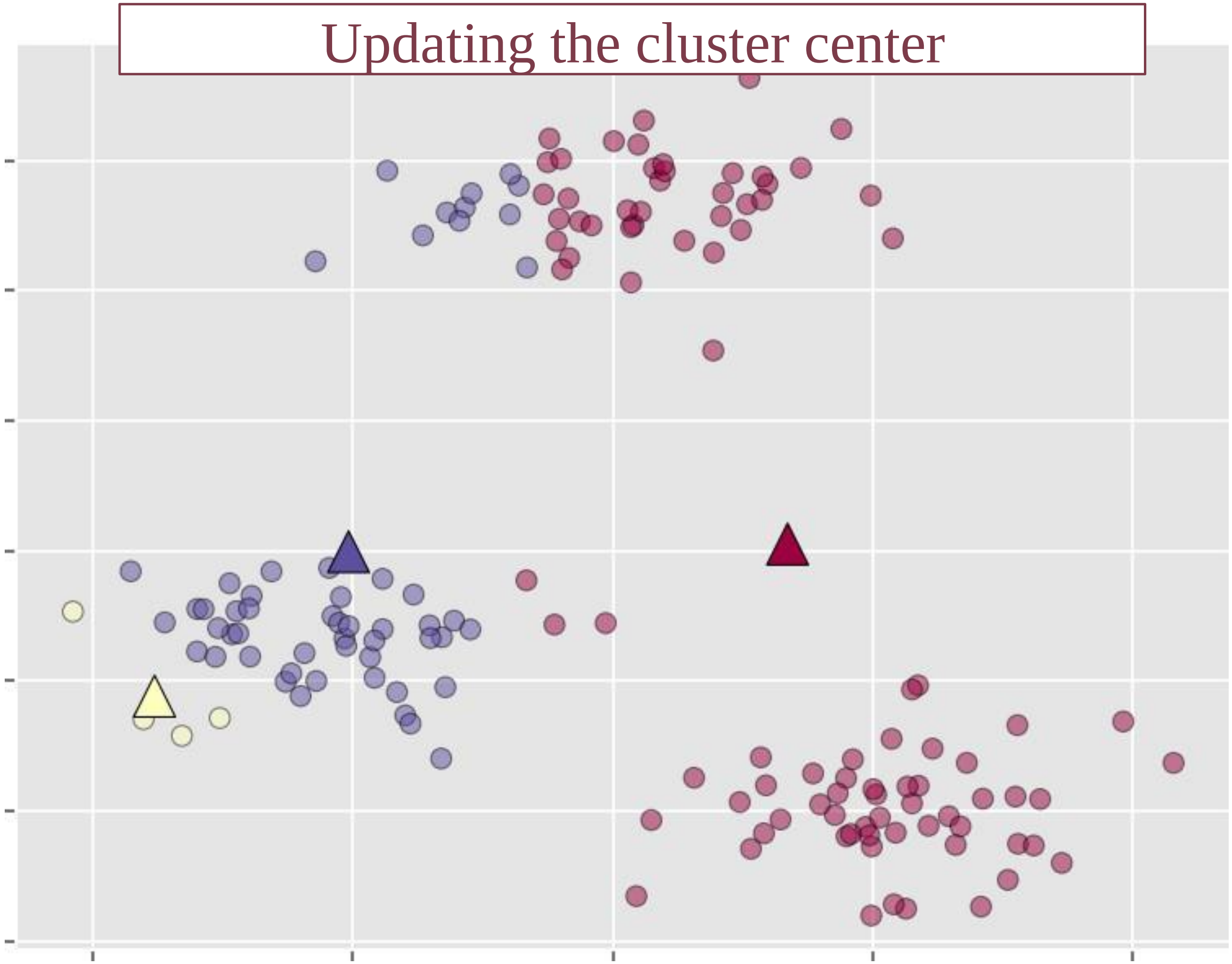
- Its an iterative clustering algorithm
  - It choosing K points randomly as the center of the clusters.
  - Repeat the following steps:
    - Assign each data to a cluster with the closest center.
    - Update the centroid of each cluster by averaging the data assigned to that cluster
  - Stop time : When no data changes its cluster in an iteration.



Illustrations of algorithm process

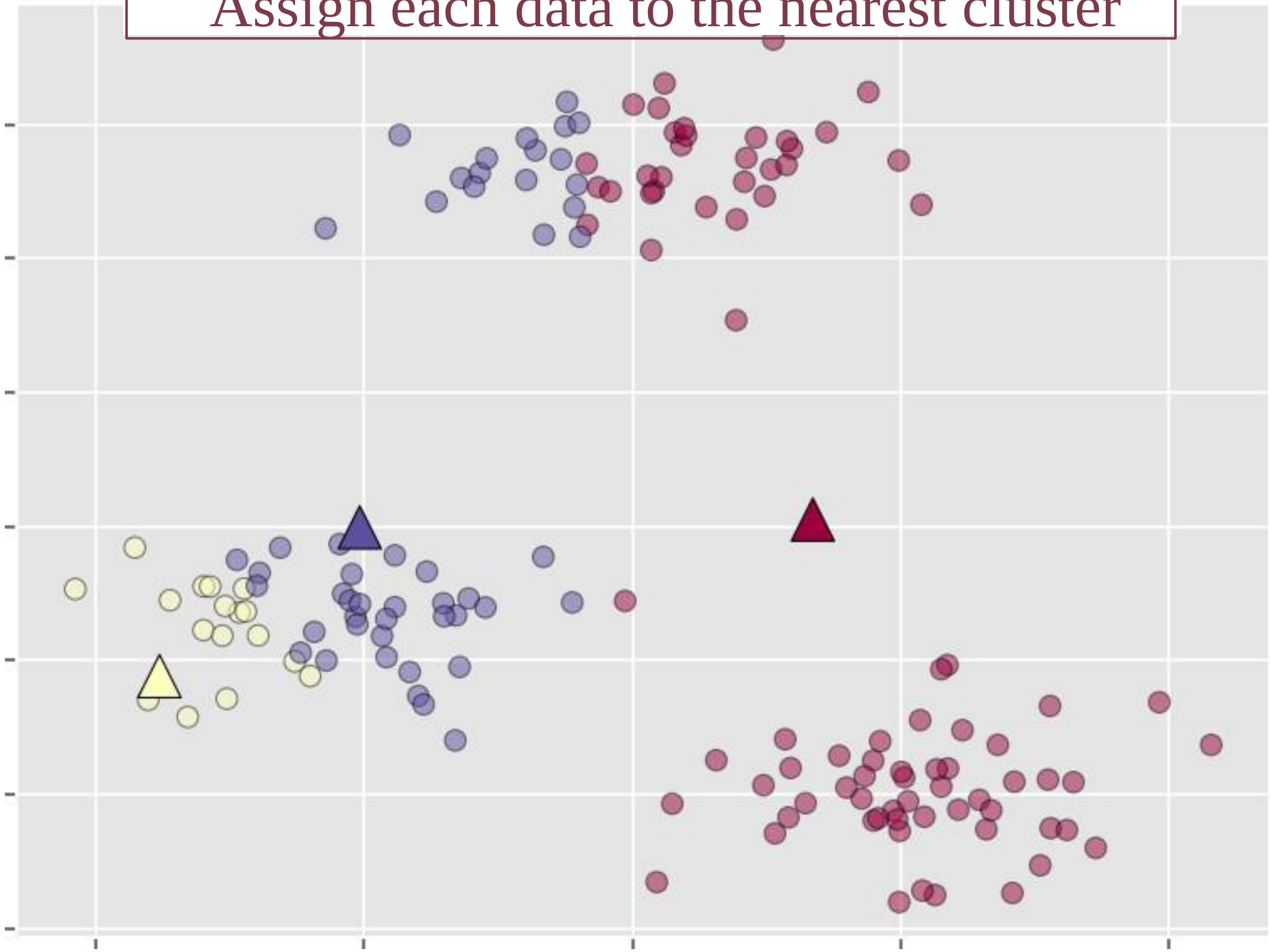


Illustrations of algorithm process



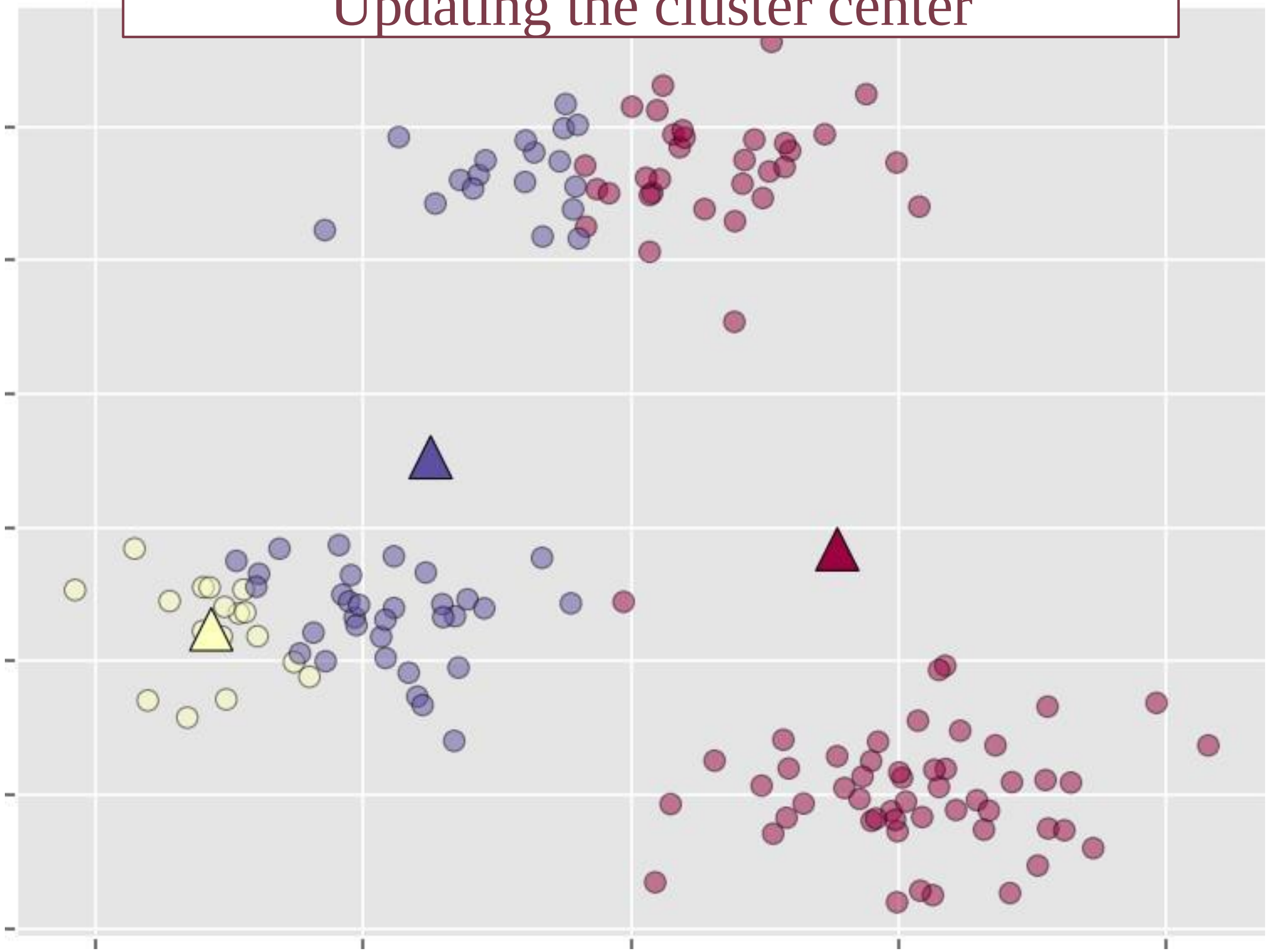
Illustrations of algorithm process

Assign each data to the nearest cluster

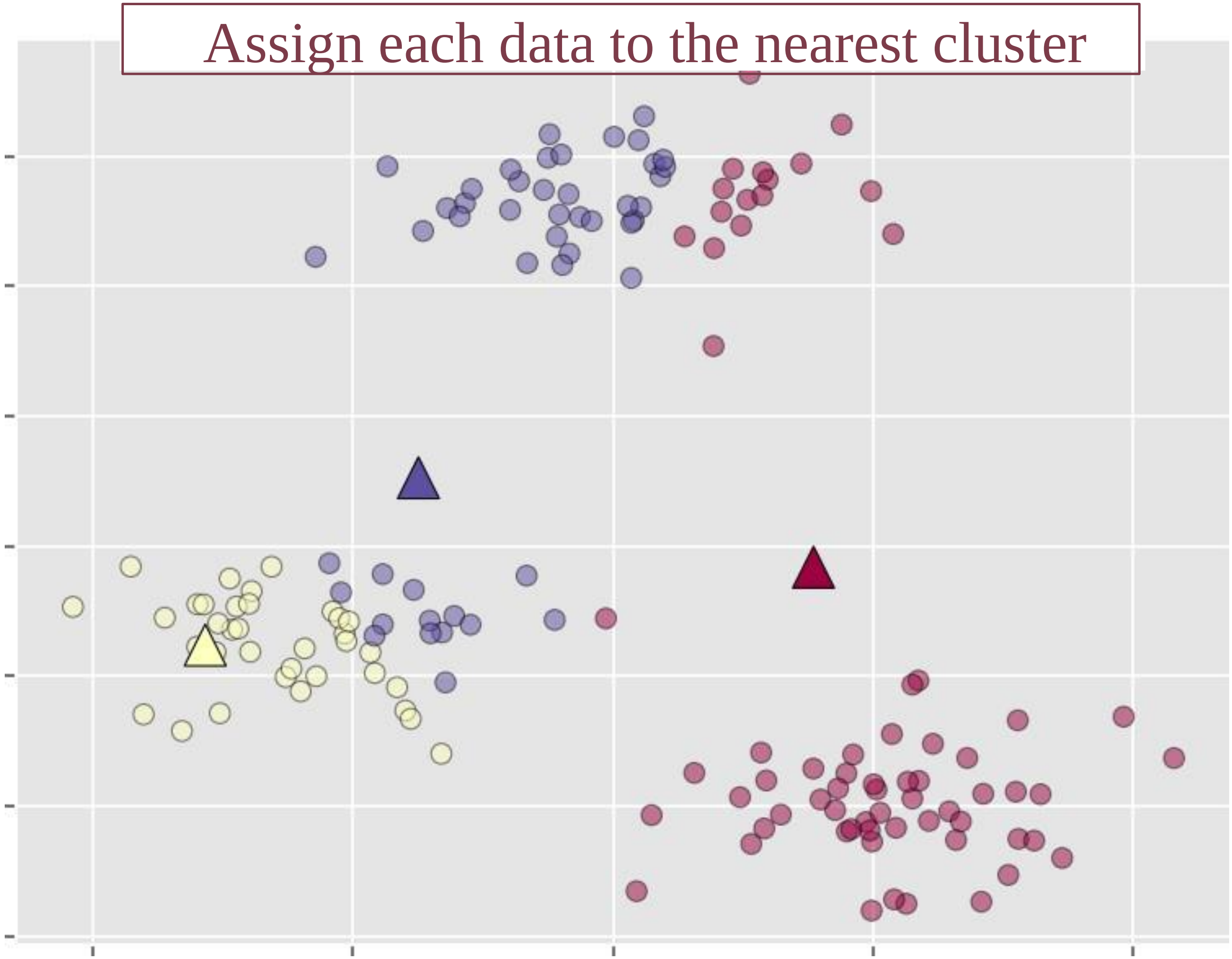


Illustrations of algorithm process

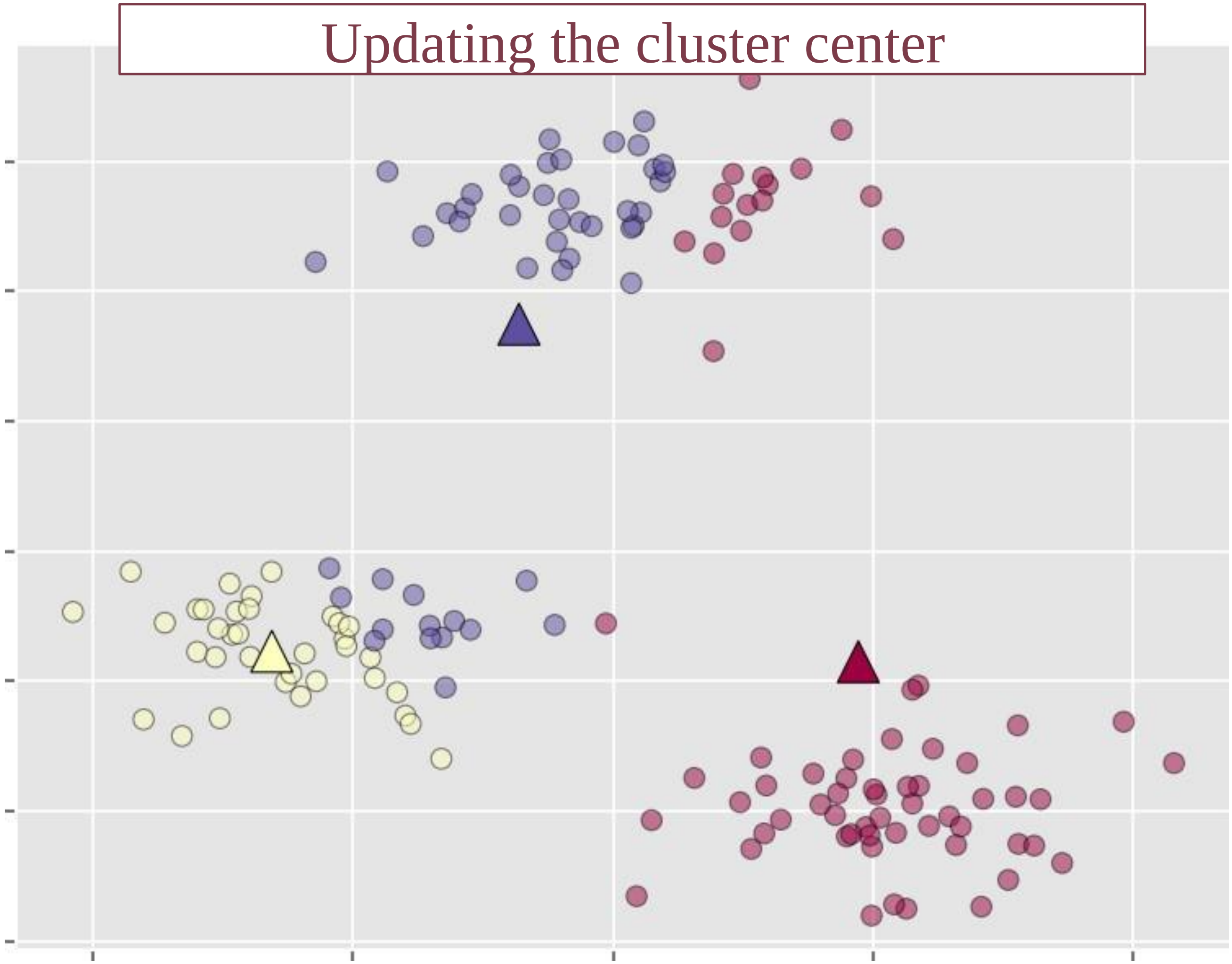
Updating the cluster center



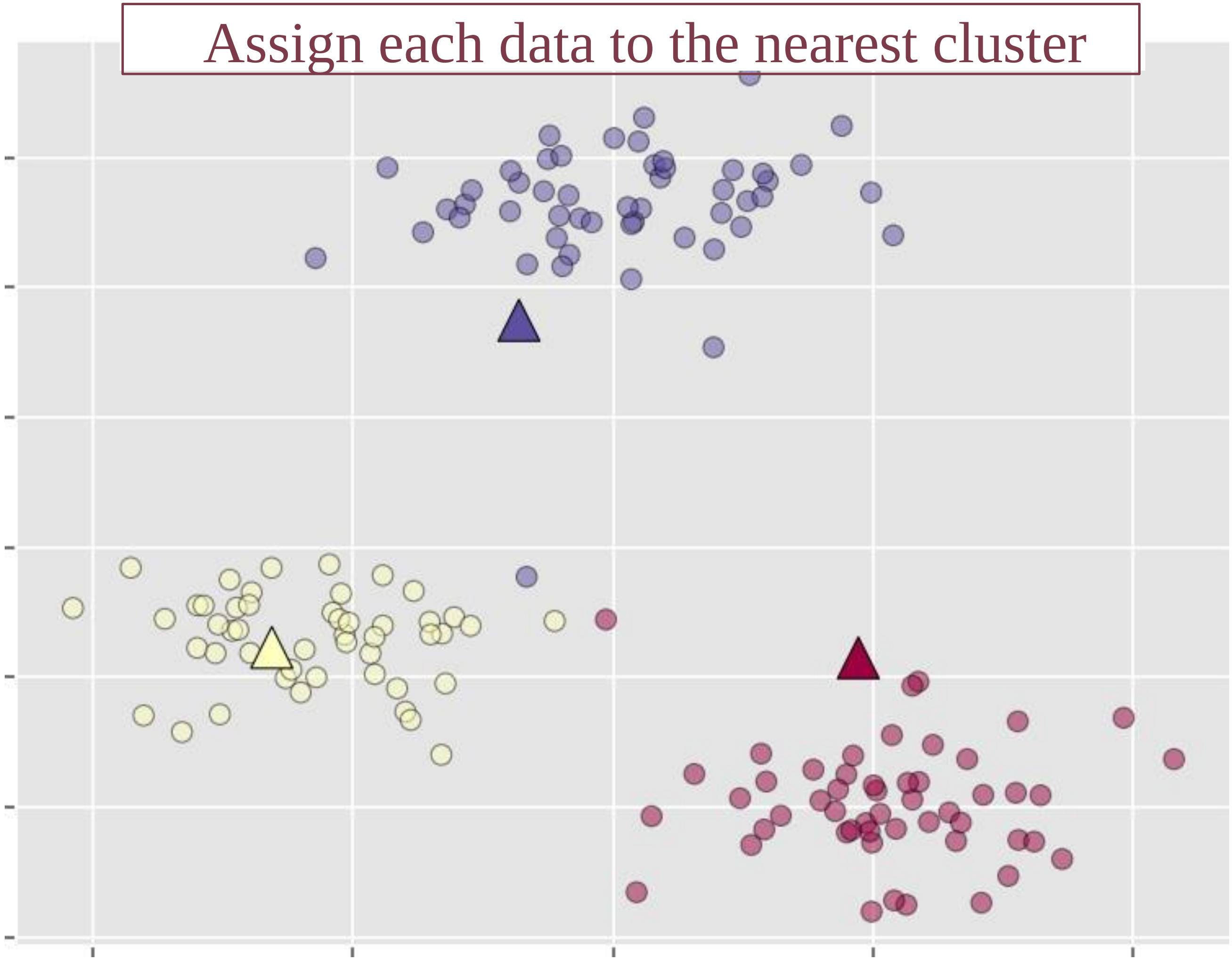
Illustrations of algorithm process



Illustrations of algorithm process

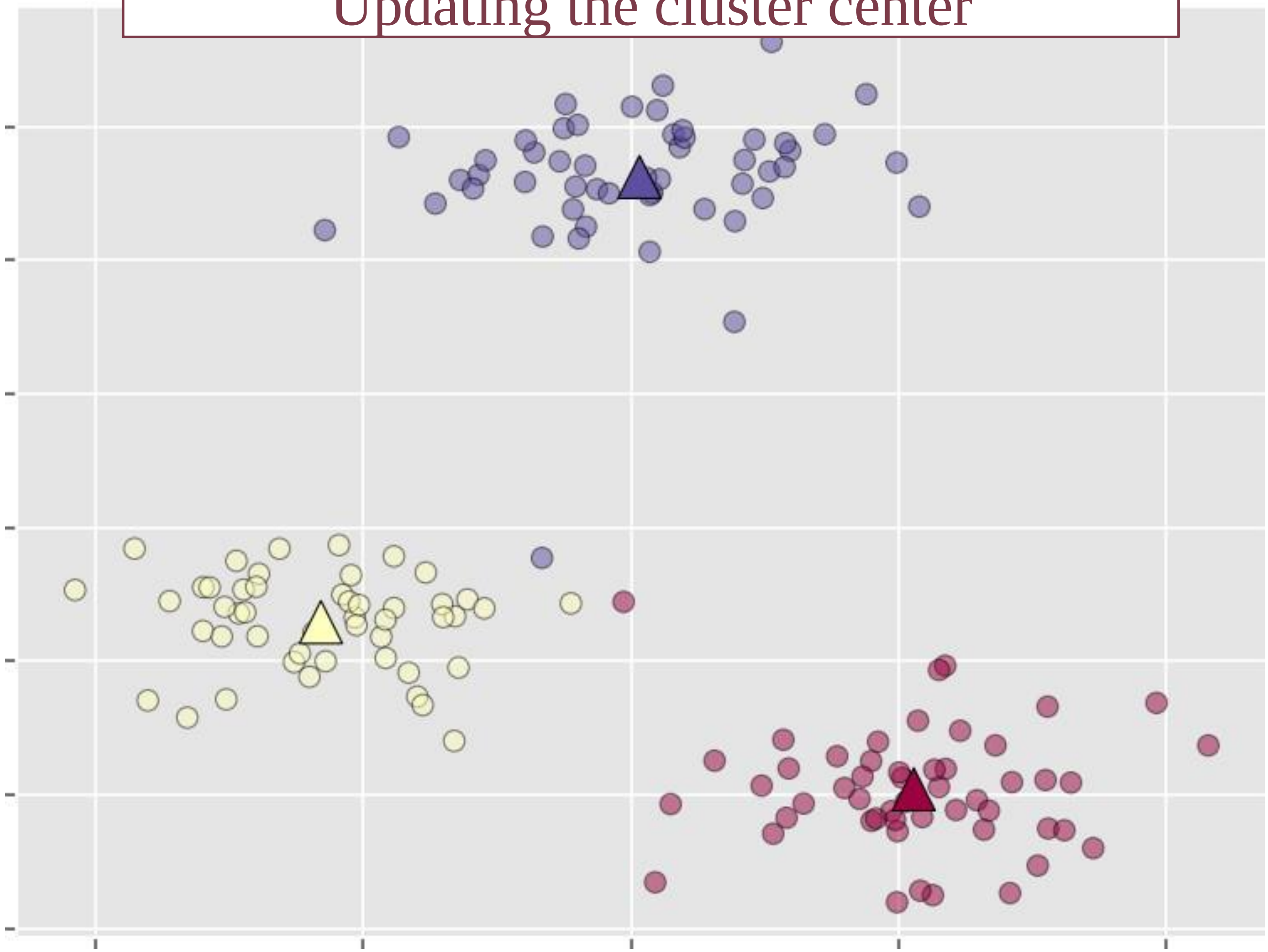


Illustrations of algorithm process



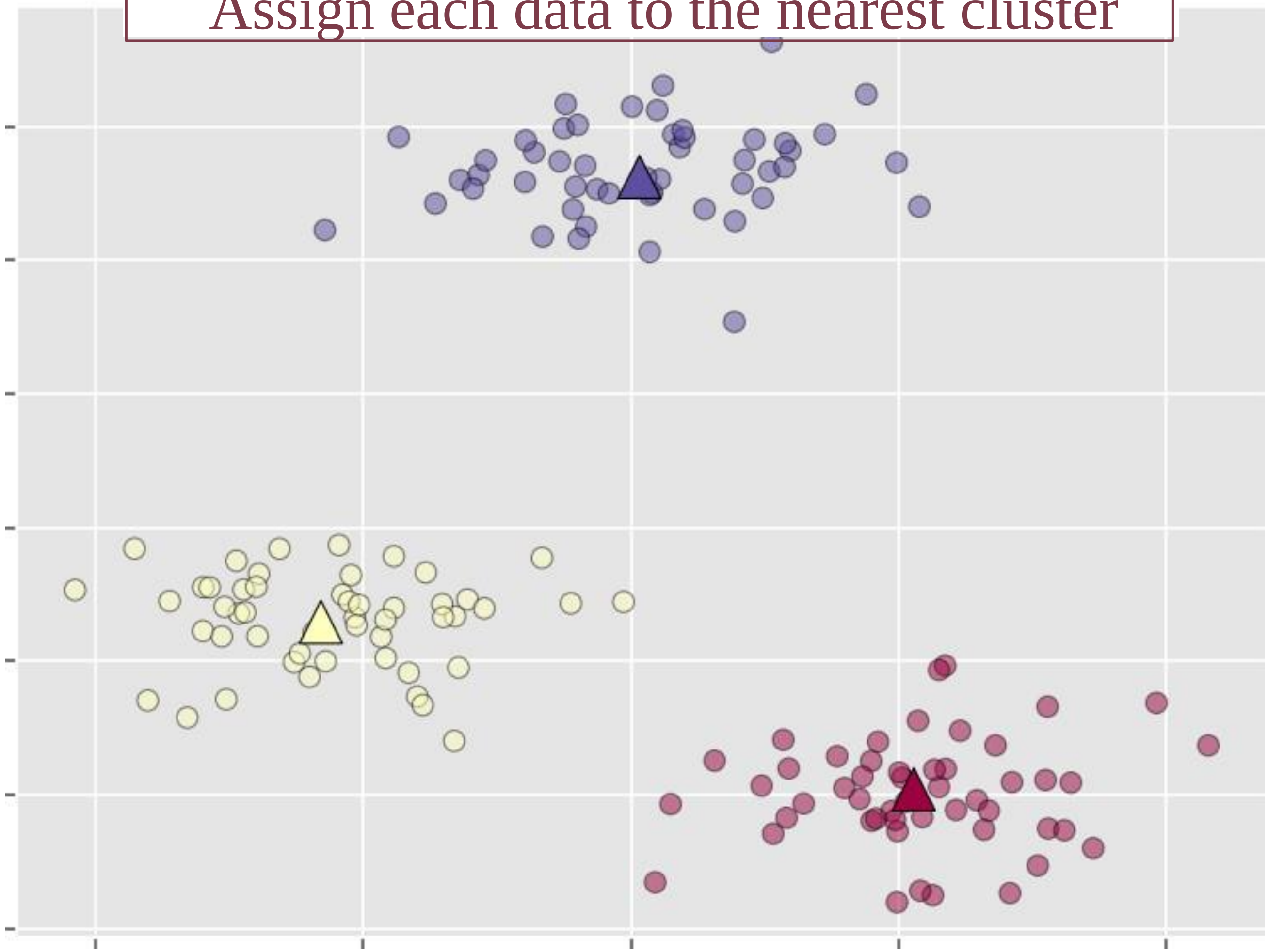
Illustrations of algorithm process

Updating the cluster center

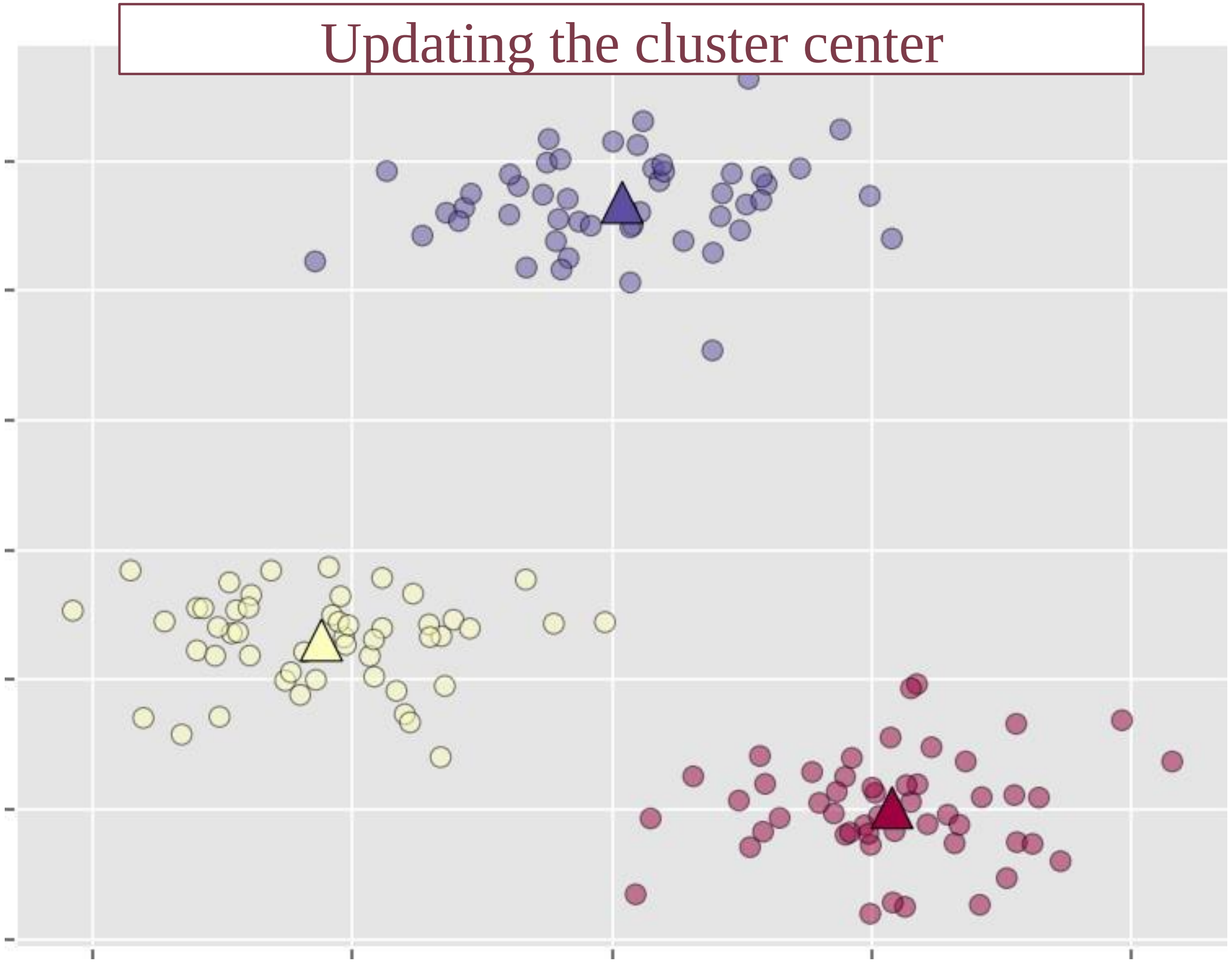


Illustrations of algorithm process

Assign each data to the nearest cluster

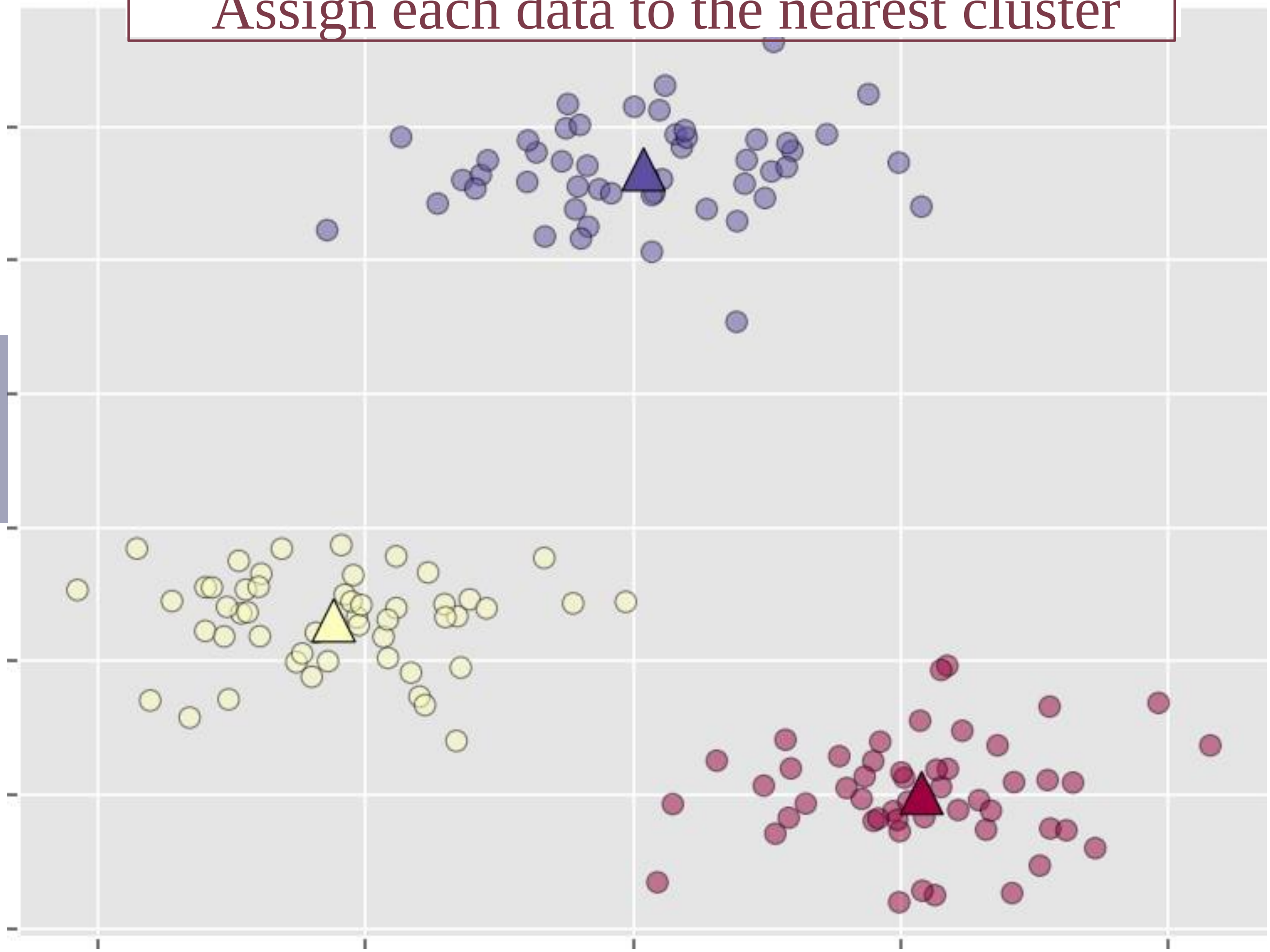


Illustrations of algorithm process



Illustrations of algorithm process

Assign each data to the nearest cluster



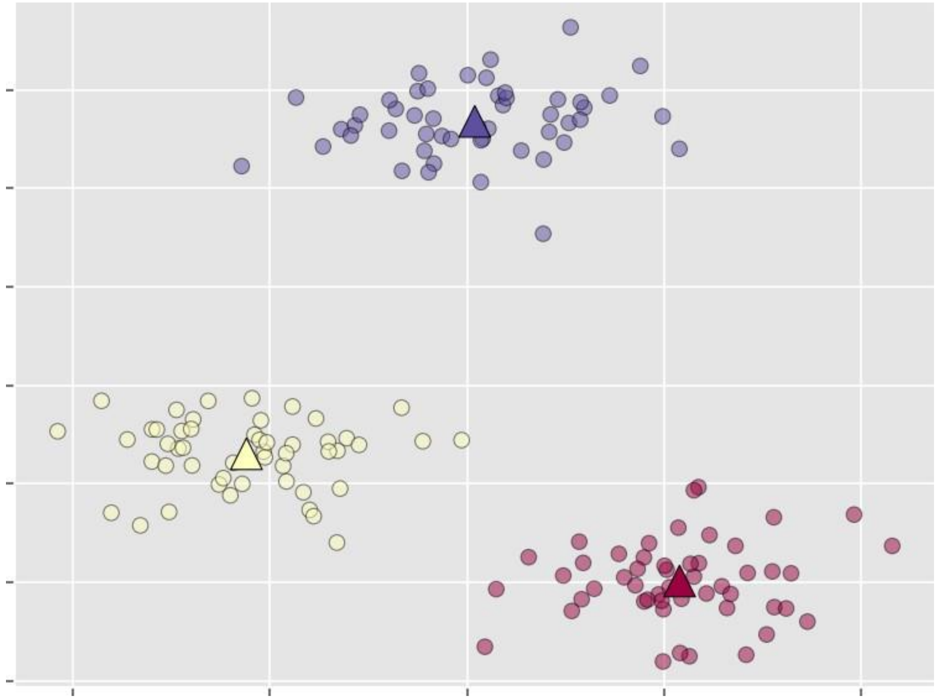
Convergence :

None of the data clusters were changed .

K-mean Algorithm

- Inputs.
  - Number Of Clusters : K
  - Training set

$$\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(m)}\}$$



- Note : In the training set, no label is assigned to the data.
- Note: There is no need to add the feature $x_0 = 1$ in the clustering.

K-mean Algorithm

randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$

repeat

{

for $i = 1$ **to** m

$$c^{(i)} = \arg\min_k \|x^{(i)} - \mu_k\|$$

Assigning data to clusters

for $k = 1$ **to** K

$\mu_k =$ average of points assigned to cluster k

Updating the cluster center

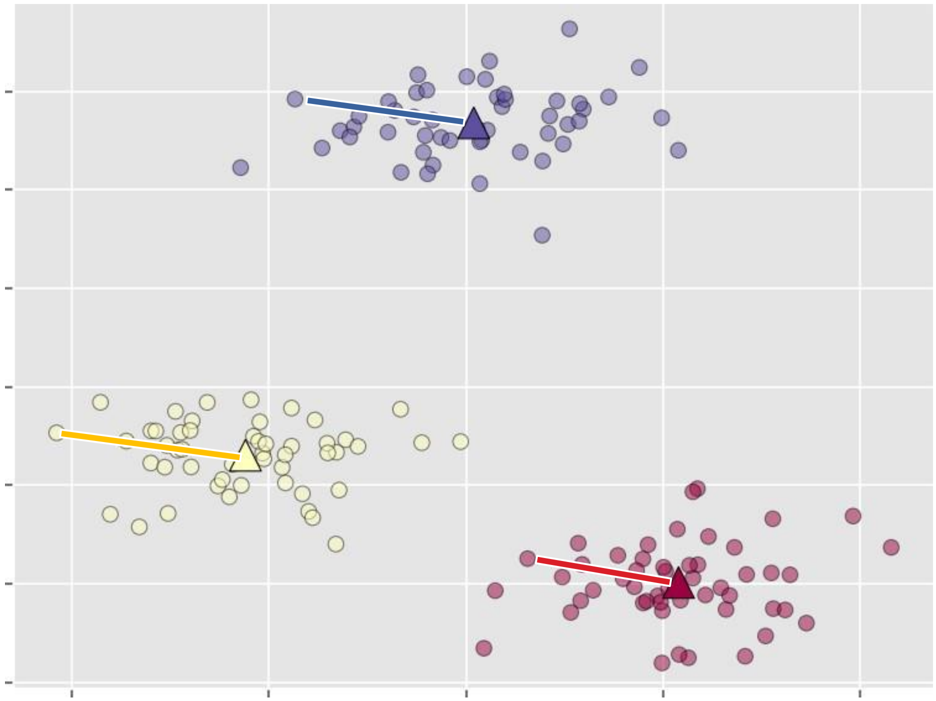
}

K-mean Algorithm


```
centroids = np.random.random((K, n))

while True:

    for i in range(m):
        c[i] = np.argmin(np.linalg.norm(X[i] - centroids, axis=1))

    for k in range(K):
        centroids[k] = np.mean(X[c == k], axis=0)
```

Clustering: Objective function

Objective function



- Symbols.
  - μ_k : center of the cluster k
  - $c^{(i)}$: Cluster number assigned to data $x^{(i)}$
  - $\mu_c(i)$: Cluster center assigned to data $x^{(i)}$

- Objective function

$$J(c^{(1)}, c^{(2)}, \ldots, c^{(m)}, \mu_1, \mu_2, \ldots, \mu_k) = \frac{1}{m}\sum_{i=1}^{m}\left\|x^{(i)} - \mu_{c^{(i)}}\right\|^2$$

K-mean Algorithm

randomly initialize K cluster centroids $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$

repeat

{

for $i = 1$ **to** m

$$c^{(i)} = \arg\min_k \left\| x^{(i)} - \mu_k \right\|$$

Minimization of the objective with respect to $c^{(i)}$ parameters

for $k = 1$ **to** K

μ_k= average of points assigned to cluster k

Minimization of the objective with respect to μ parameters

}

Setting initial value for cluster centers

K-mean Algorithm

randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$

repeat

{

for $i = 1$ **to** m

$$c^{(i)} = \arg\min_k \left\| x^{(i)} - \mu_k \right\|$$

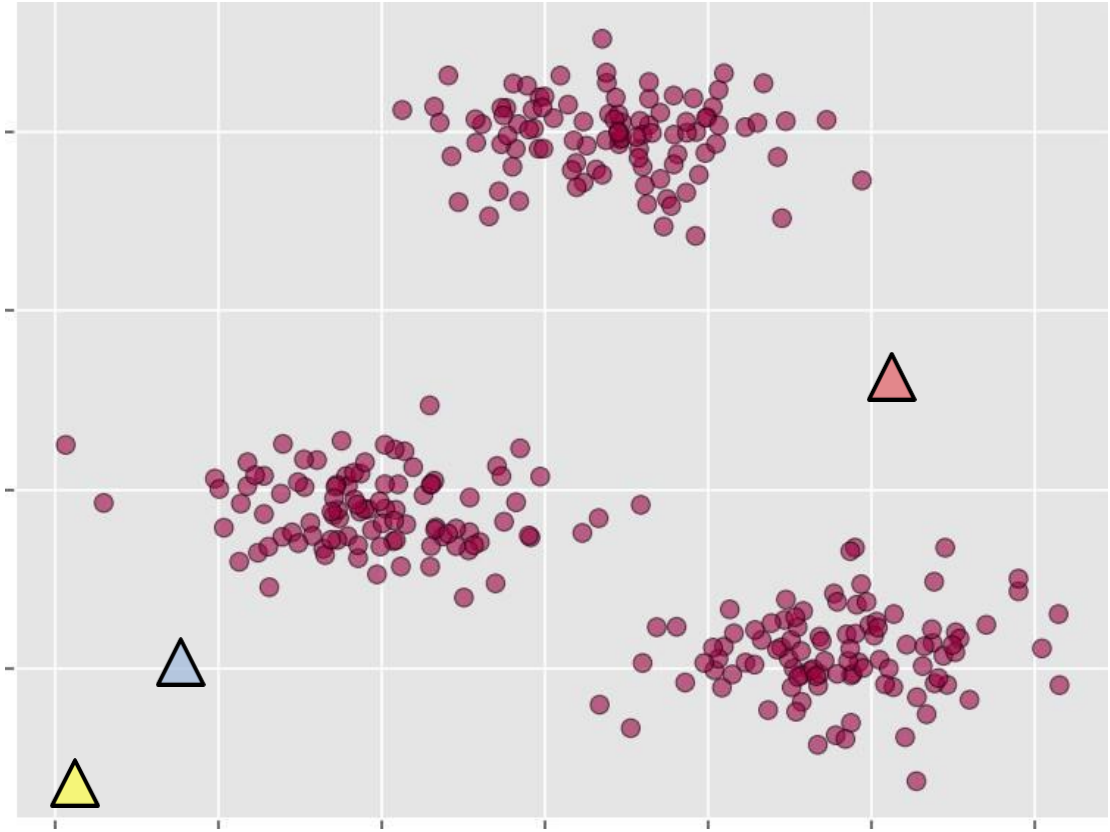
for $k = 1$ **to** K

μ_k= average of points assigned to cluster k

}

Setting initial value for cluster centers

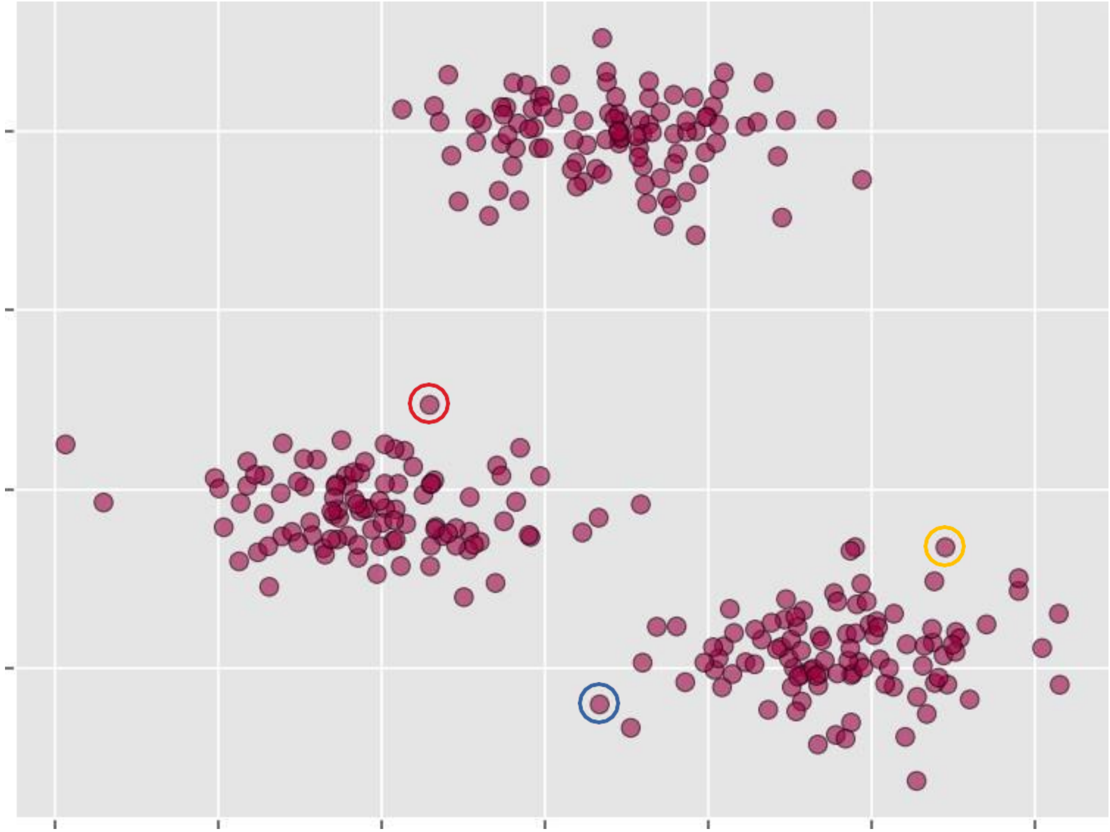
- Setting initial value. $[K \leq m]$
  - Selection of K training samples randomly
  - Assigning center of each cluster to K selected samples



A center may be selected in such a way that no data is assigned to it.

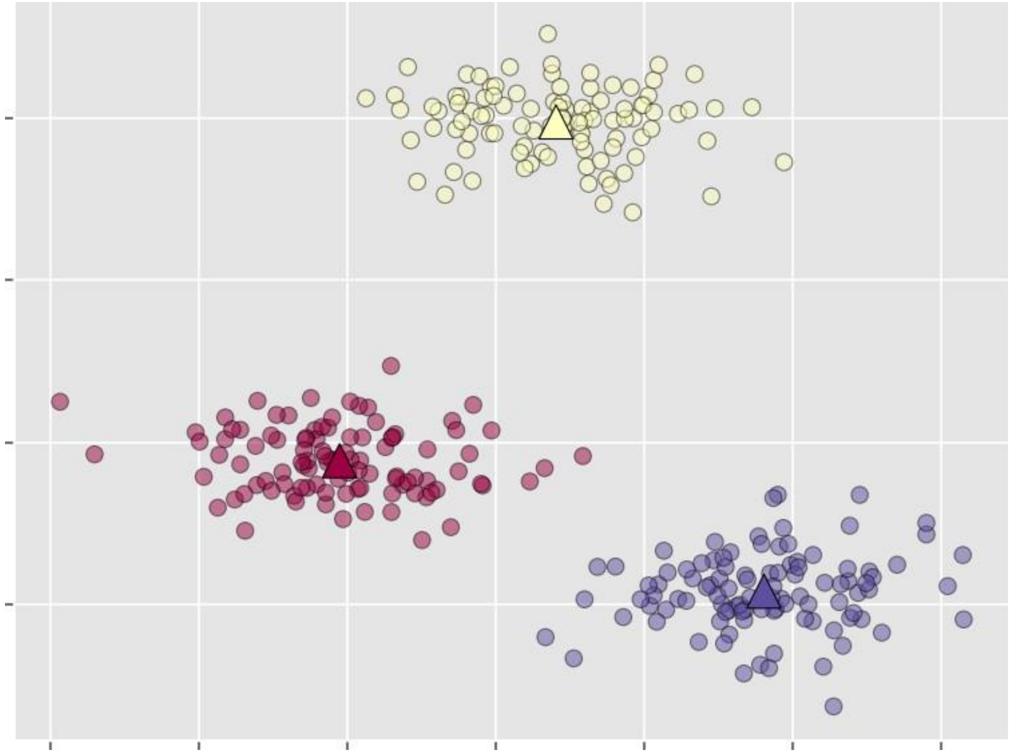
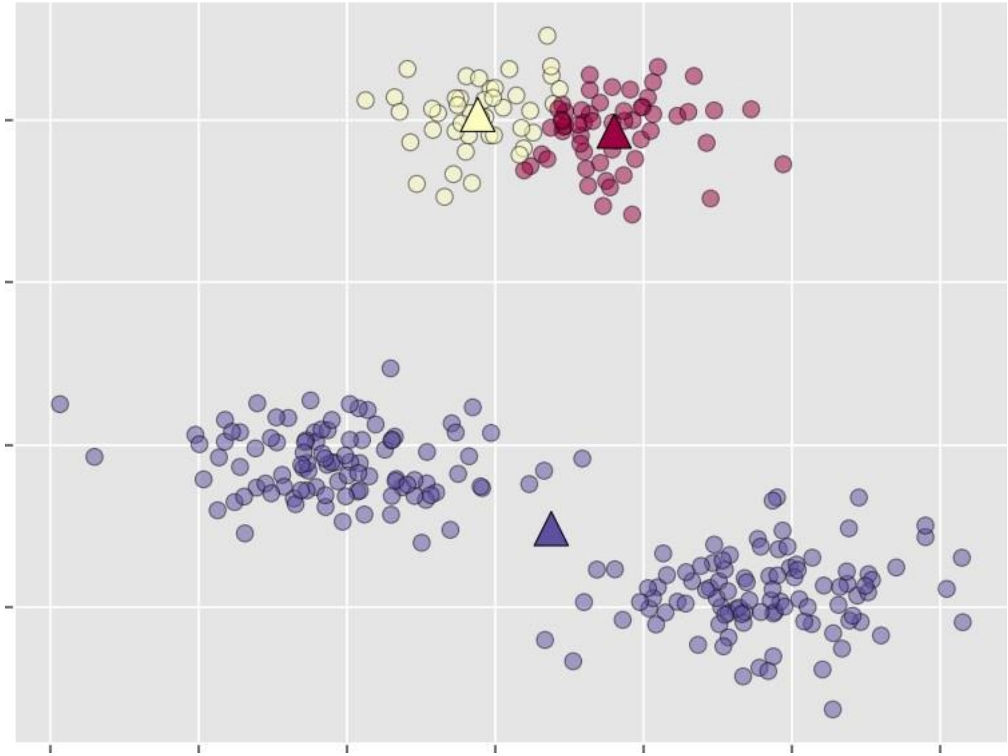
Setting initial value for cluster centers : Better way

- Setting initial value. $[K \leq m]$
  - Choosing k amount of training samples randomly
  - Assigning center of each cluster to K selected samples



```
C = np.random.permutation(X)[:K]
```

Global optimization and Local optimization

Local optimization	Global optimization
	

Avoid local optima

for $t = 1$ **to** MAX

{

randomly initialize cluster centroids $\mu_1, \mu_2, \dots, \mu_k$

run K-means to get $c^{(1)}, c^{(2)}, \dots c^{(m)}, \mu_1, \mu_2, \dots, \mu_k$

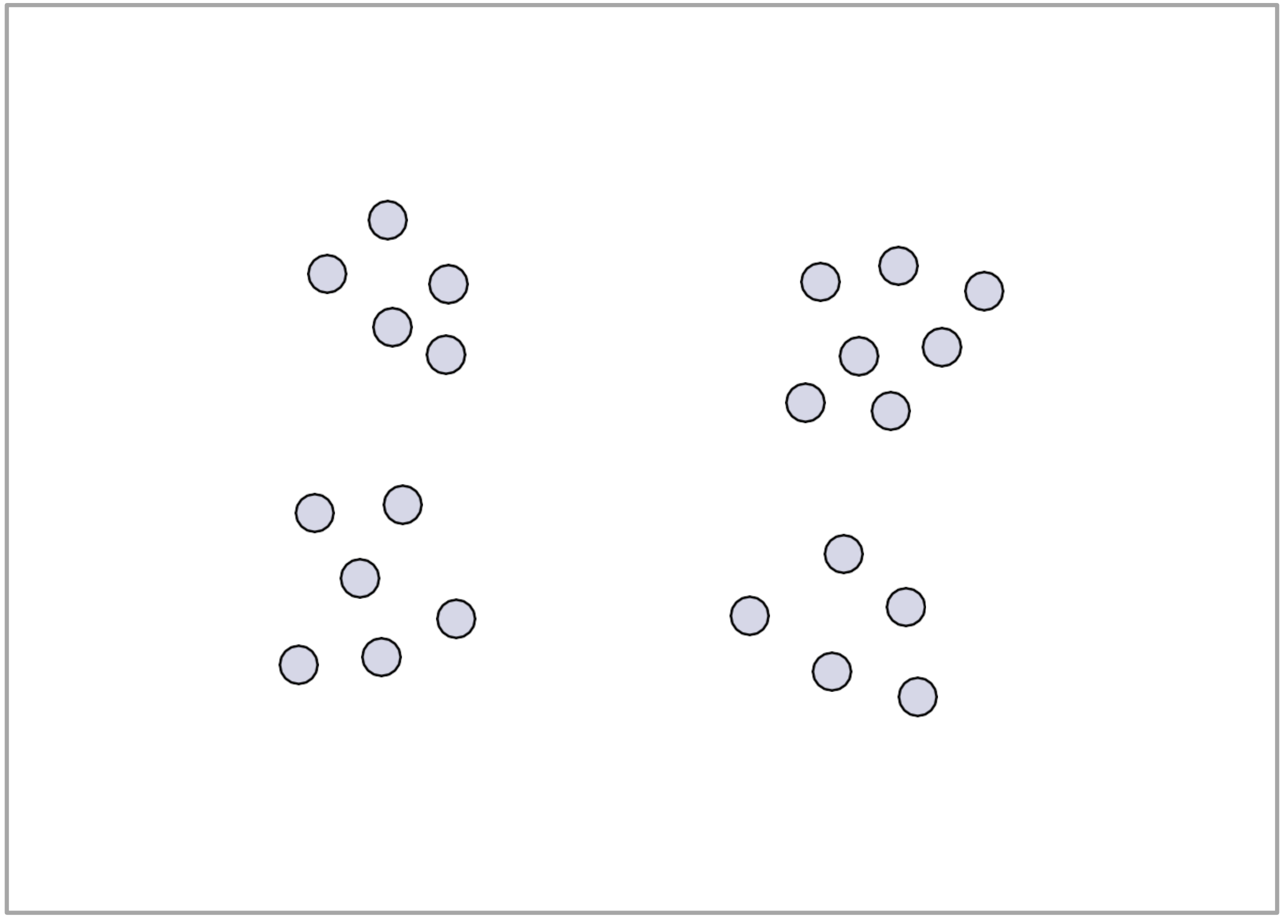
compute cost function $J(c^{(1)}, c^{(2)}, \dots c^{(m)}, \mu_1, \mu_2, \dots, \mu_k)$

}

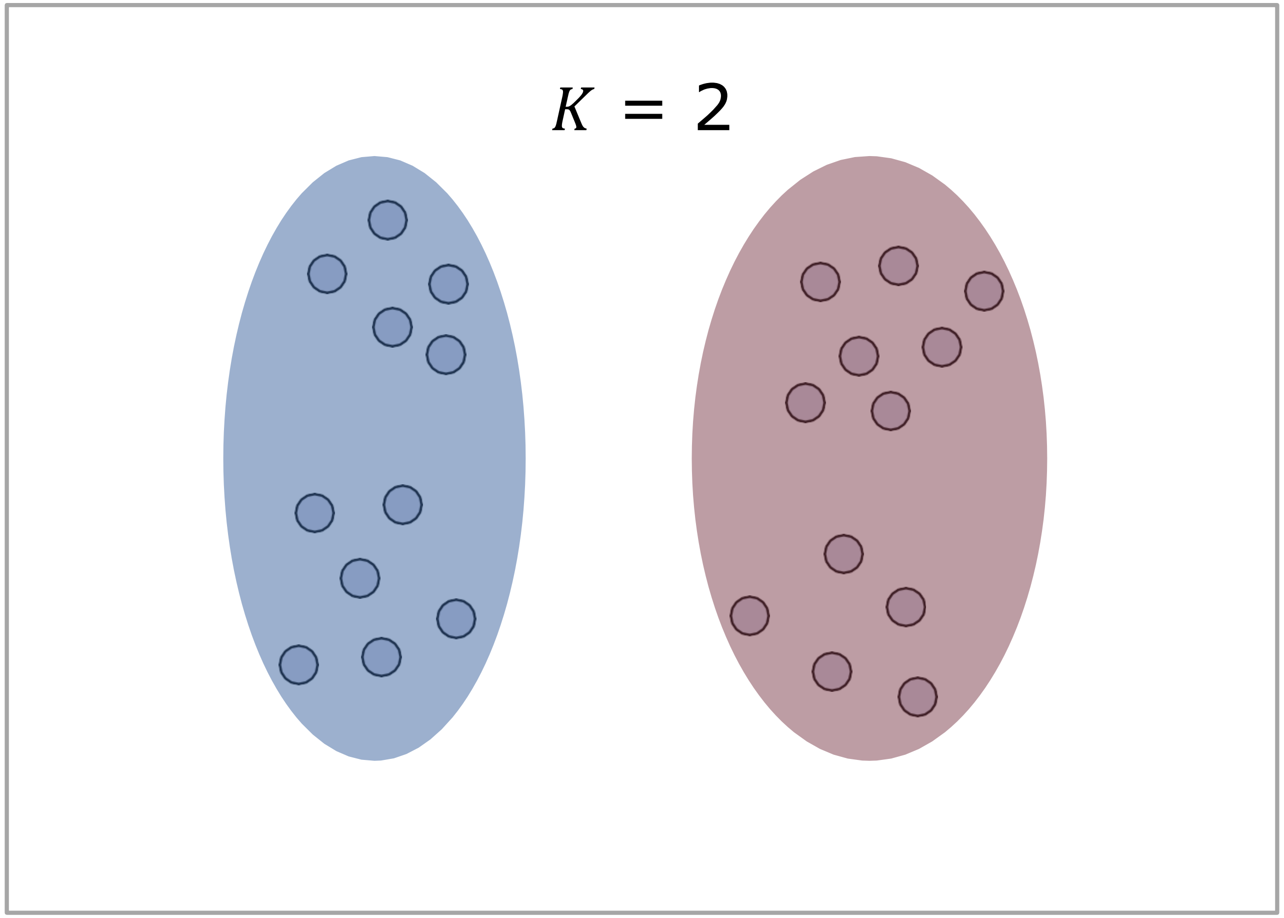
pick clustering with minimum cost

Determine the number of clusters

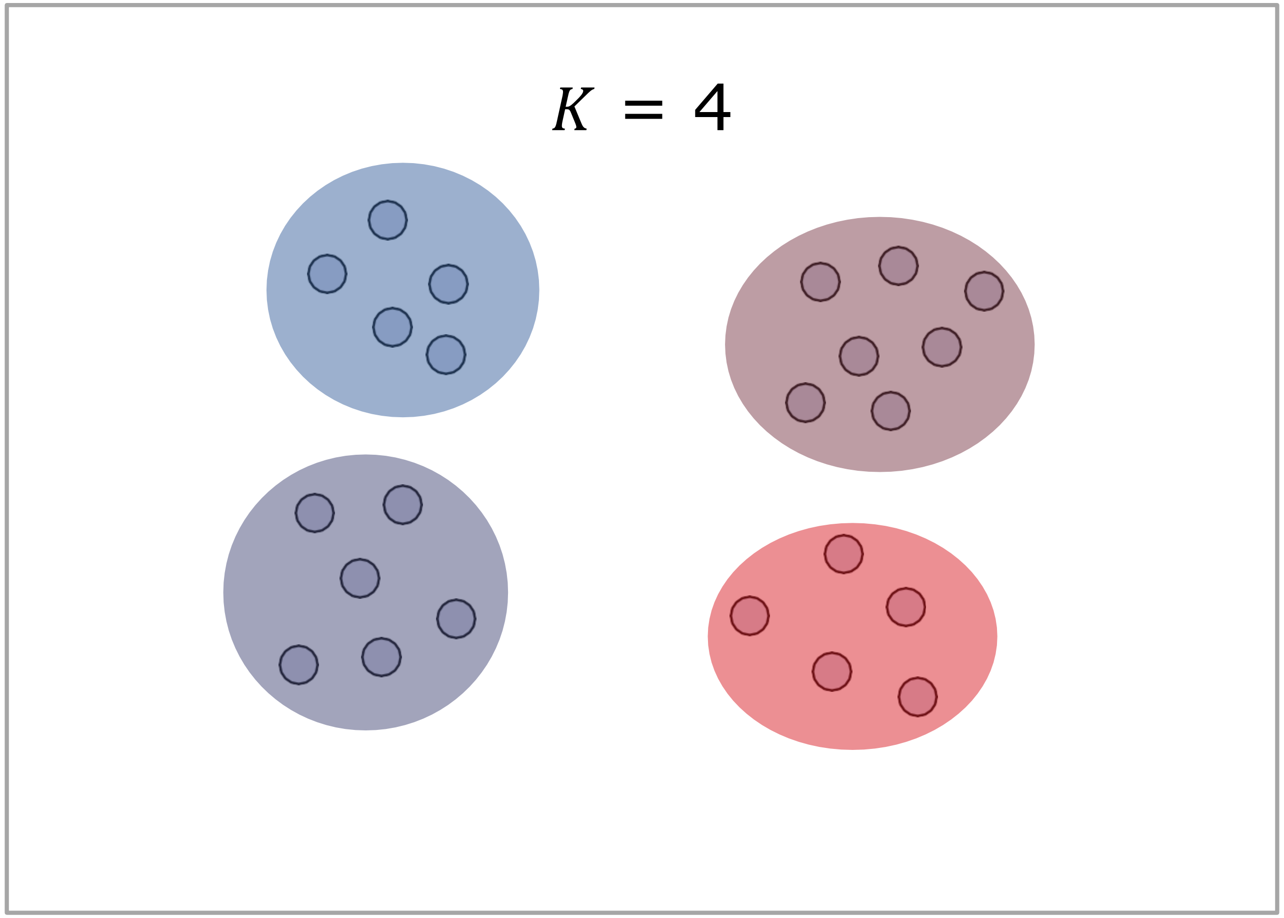
What is the right value for K?



What is the right value for K?



What is the right value for K?



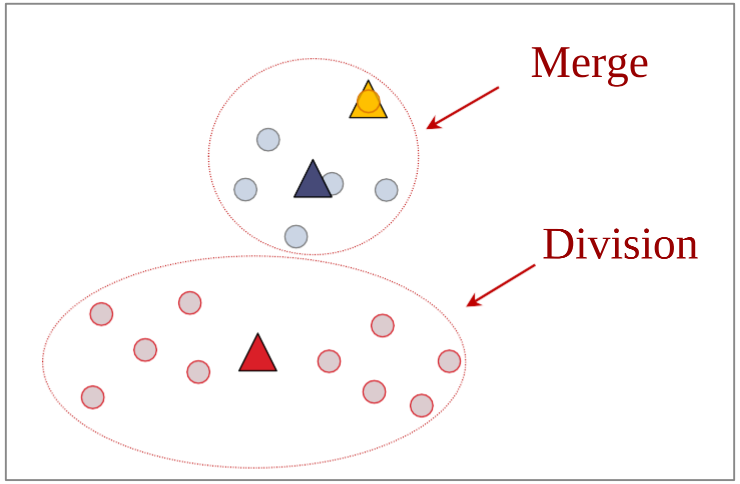
Determining the appropriate number of clusters: Elbow Method

Improving clustering

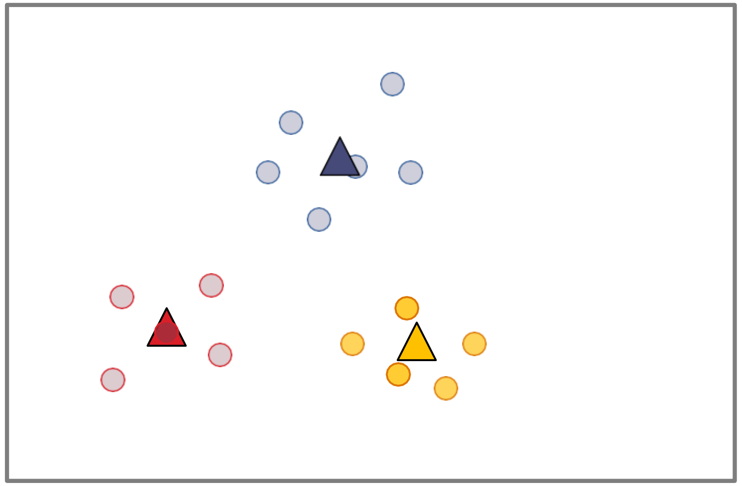
Improving clustering with post-processing of clusters

- **Division.**
  - Splitting a cluster with the highest error into two clusters.

By running the K-means on the data of this cluster with a value of K = 2



- **Merge.**
  - Merge the two closest clusters
  - Merging two clusters with minimal increase in total error.



Algorithm of bipartite generator K-Means

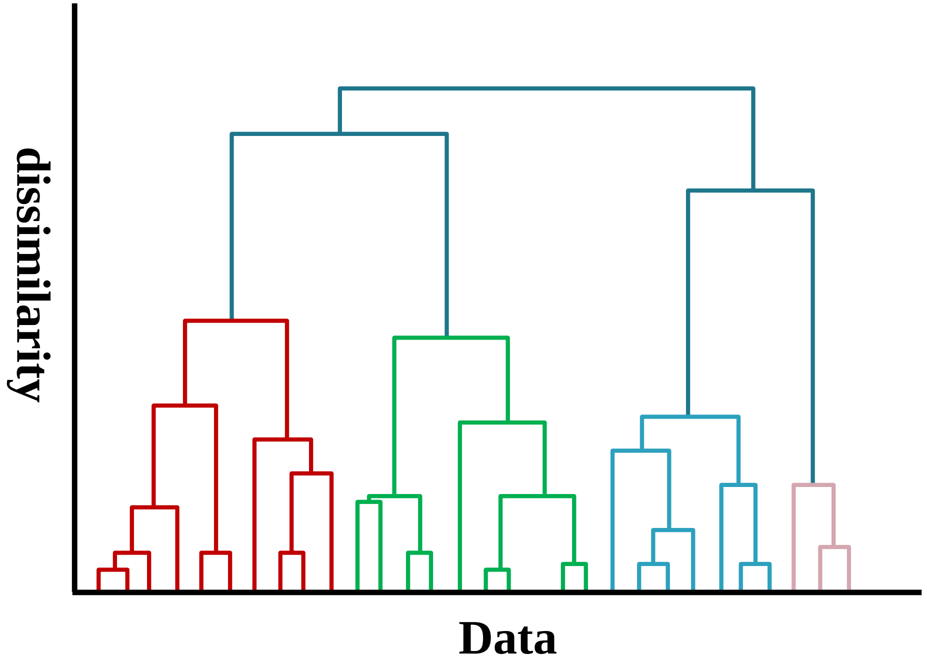
- Bipartite algorithm.
  - Start with a cluster containing all the data.
  - Choose one cluster at a time :
    - Divide the selected cluster into two clusters using K-means algorithm.
    - Calculate the total clustering error.
    - Choose the clustering with the least error.
  - Repeat the process above until you reach the desired number of clusters.

Algorithm of bipartite generator K-Means

Start with all the points in one cluster
while the number of clusters is less than K
    measure the total error
    for every cluster
        perform K-means clustering with $k=2$ on the given cluster
        measure the total error after splitting
    choose the cluster split that gives the lowest error

Hierarchical clustering

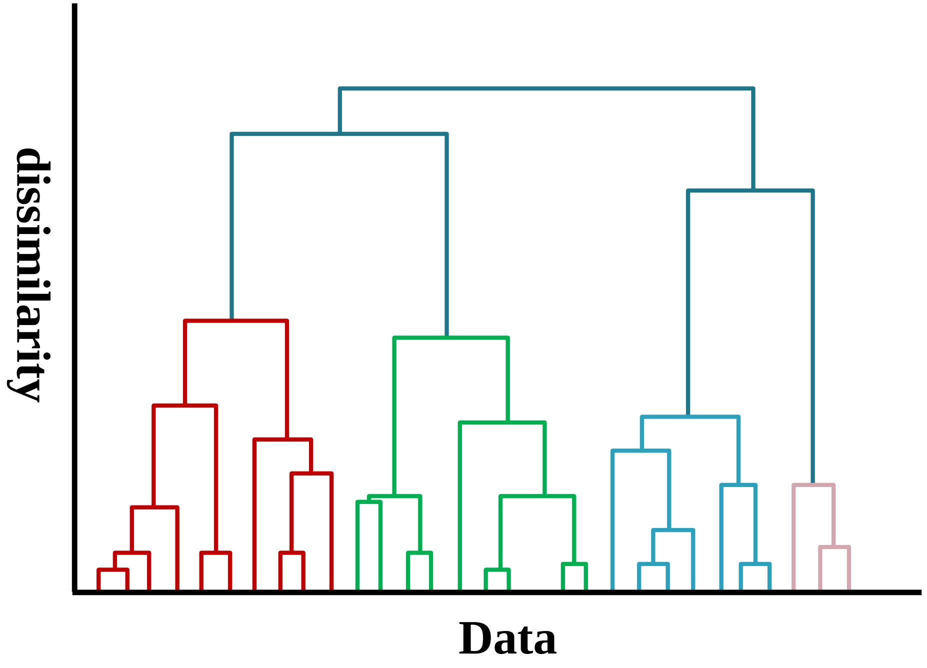
- Hierarchical clustering.
  - First, merge very similar data.
  - Gradually create larger clusters by merging smaller clusters.
- Algorithm.
  - At first, each data represents a cluster.
  - Repeat the following steps:
    - Choose the two closest clusters each time.
    - Merge those two clusters into a new cluster.
    - Stop: when only one cluster remains.



- Create a tree diagram containing a wide range of clusters.

Hierarchical clustering

- Hierarchical clustering.
  - First, merge very similar data.
  - Gradually create larger clusters by merging smaller clusters.

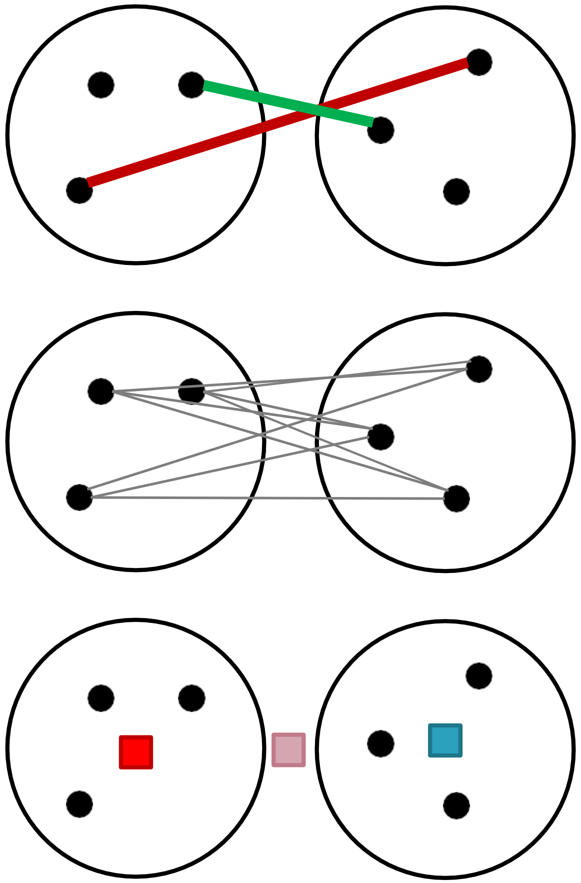


- Algorithm.
  - At first, each data represents a cluster.
  - Repeat the following steps:
    - Choose the two closest clusters each time.
    - Merge those two clusters into a new cluster.
    - Stop: when only one cluster is left.
- Create a tree diagram containing a wide range of clusters.

Hierarchical clustering

- How to define the closest two clusters.

- Criteria for determining clusters similarity.
  - Nearest pair (one-link clustering)
  - Farthest pair (all-link clustering)
  - Average distance of all pairs
  - «Enter» method (least variance, like k-means)



- Different criteria create different clusters.

Clustering of digits

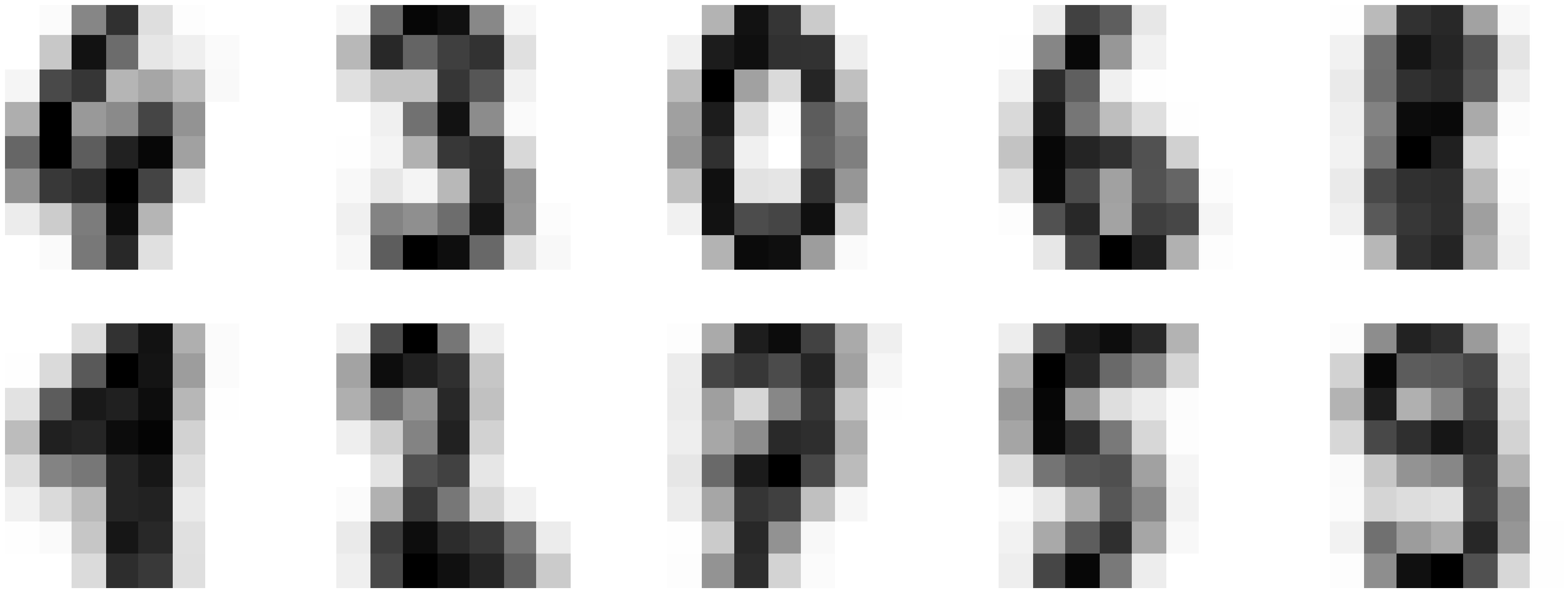


Image compression

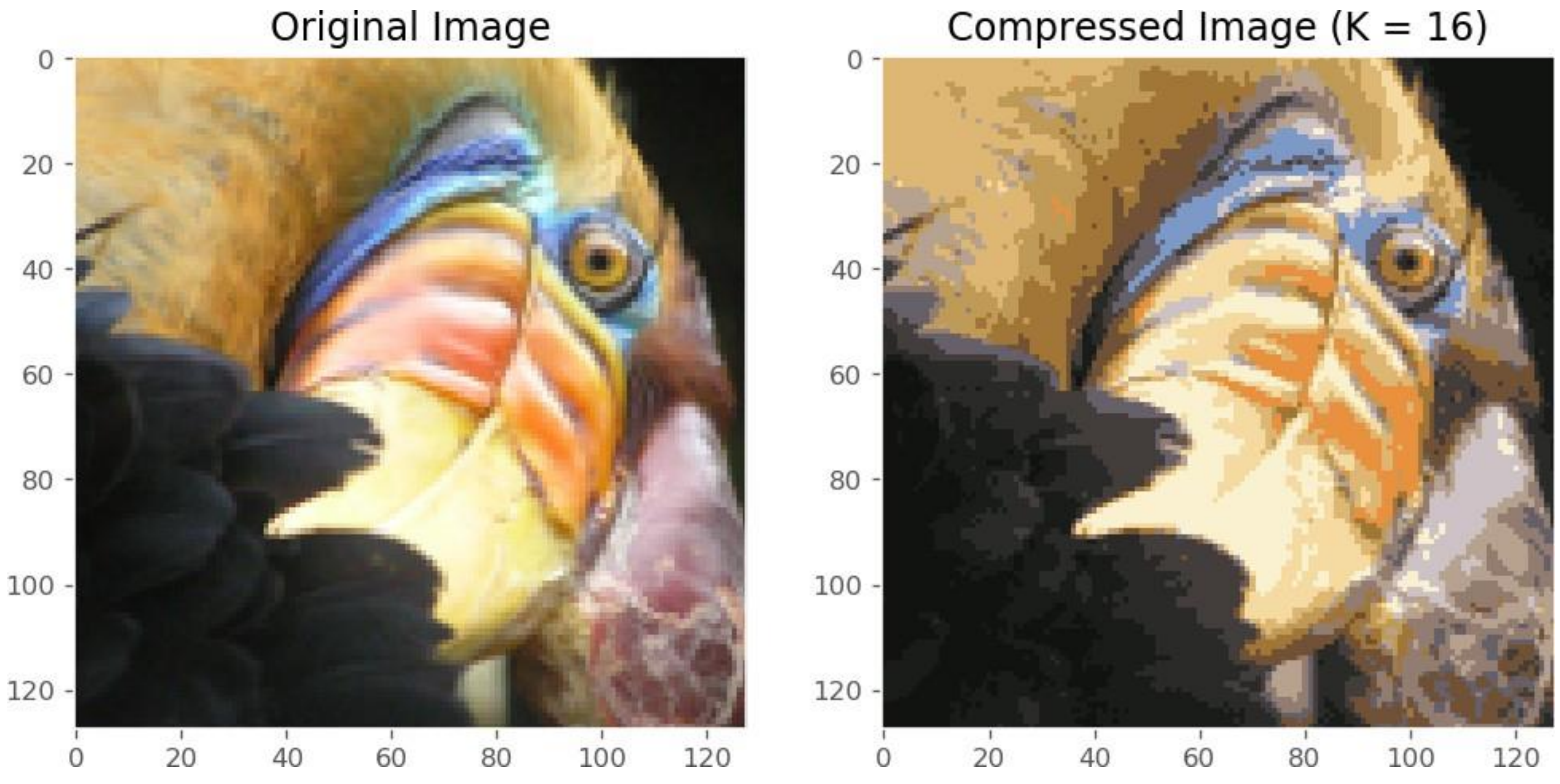
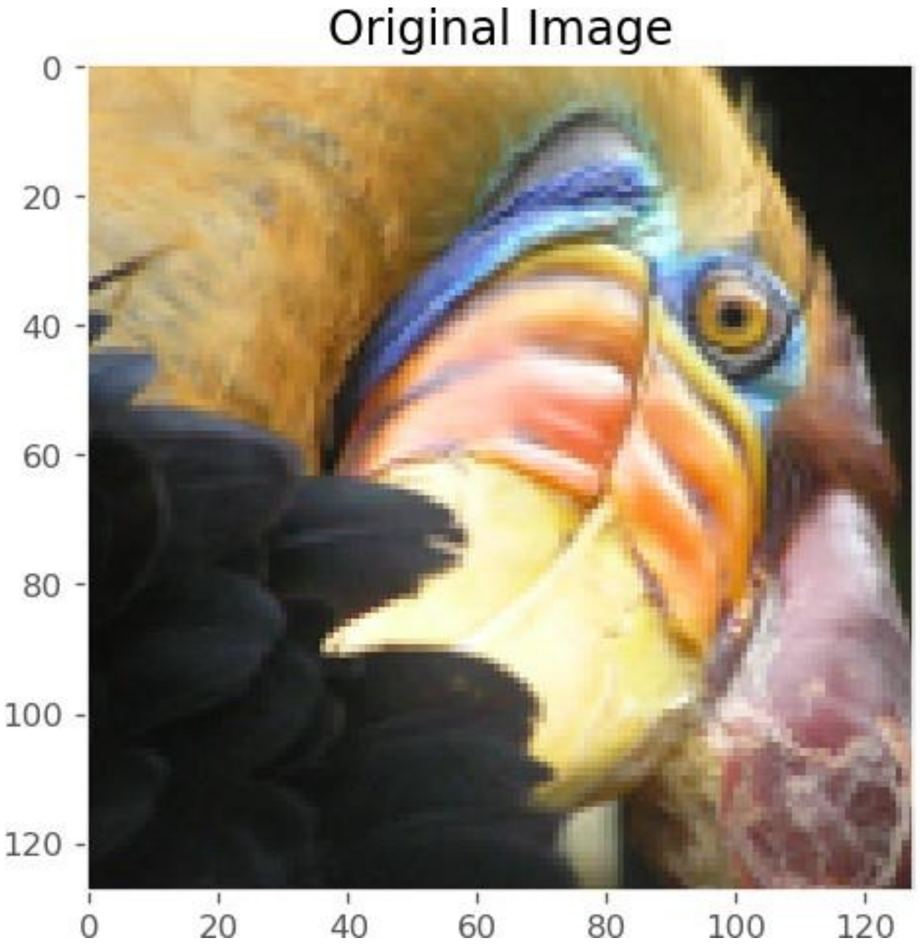
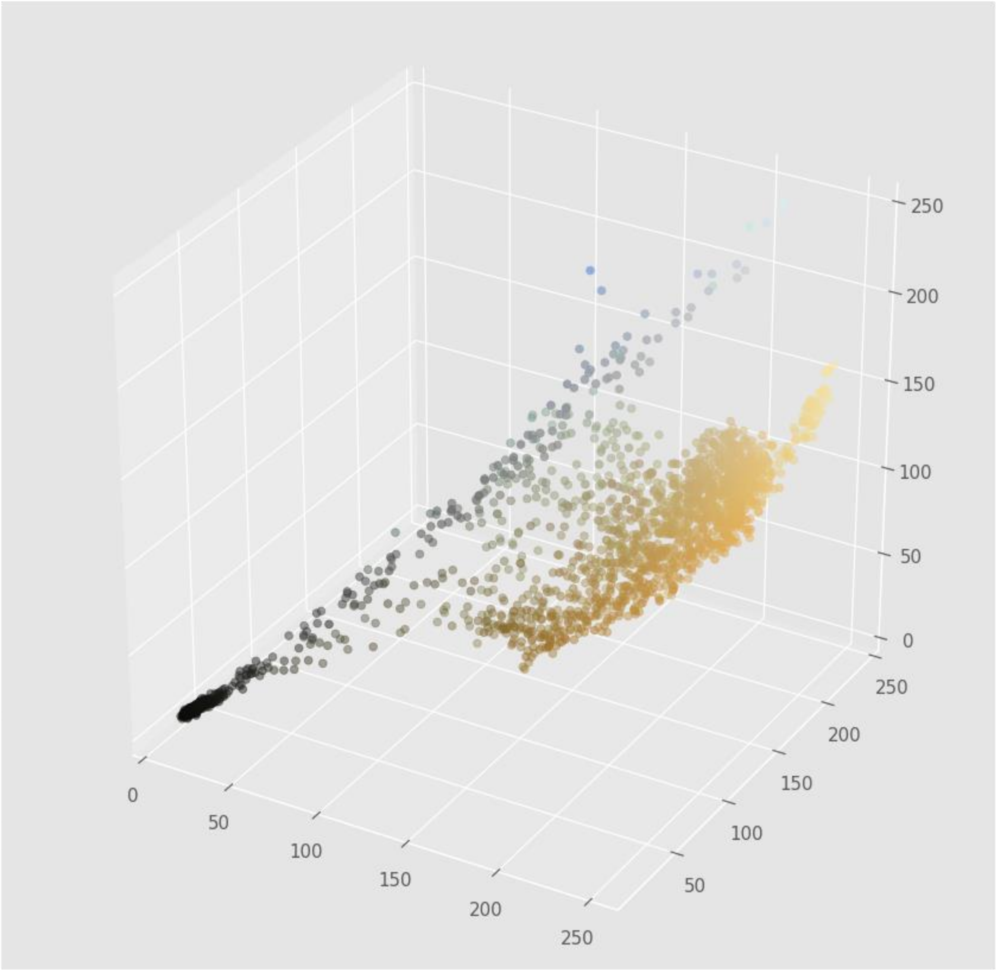


Image compression





Summary

- Unsupervised learning, Finding structure in data

- Clustering Grouping similar data
  - K-mean Algorithm
    - Easy implementation
    - Slow for very large data sets
    - Possibility of getting stuck in the local optimum
  - Post-processing of clusters : Splitting and merging clusters
  - Algorithm of bipartite generator K-Means
    - Better clustering than the K-Means algorithm
  - Hierarchical clustering algorithms