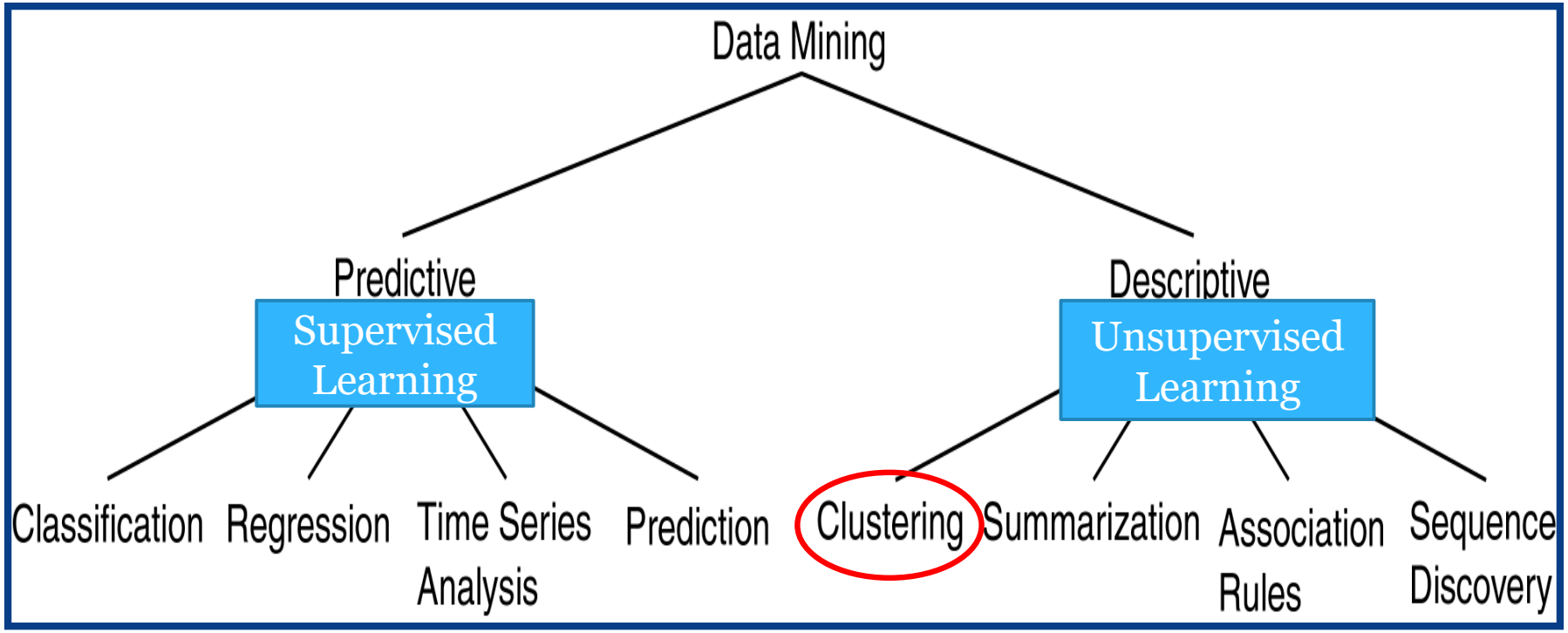
# Unsupervised Learning: Cluster Analysis with K-means

Dr. Yuzana Win (Nagasaki University, Japan)
Lecturer
Department of Computer Engineering and Information Technology

# Lecture Objectives

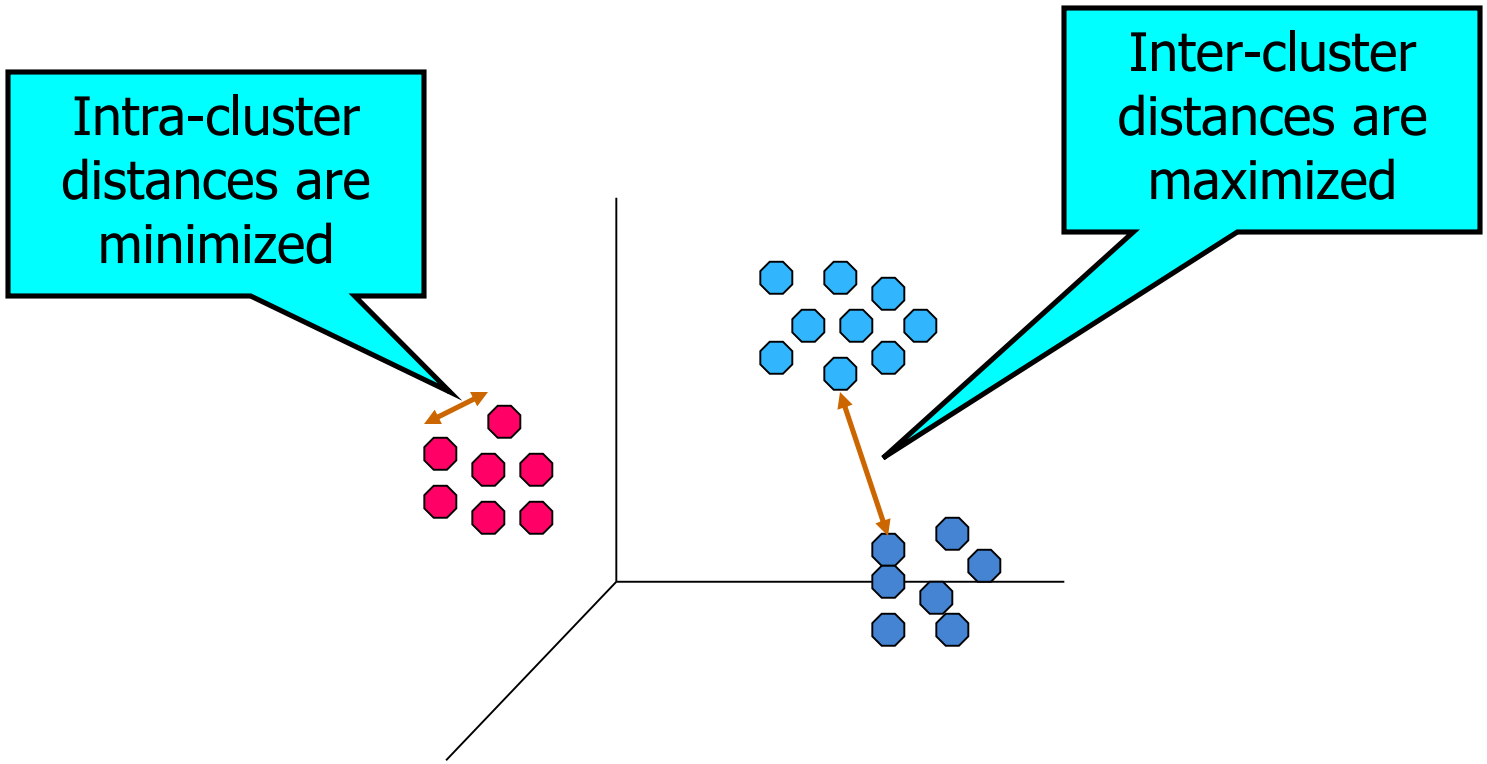
- To introduce
  - What is clustering?
  - What is K-means clustering?
  - How does K-means clustering works?
  - Application of K-means clustering

# Data Mining Methods and Models



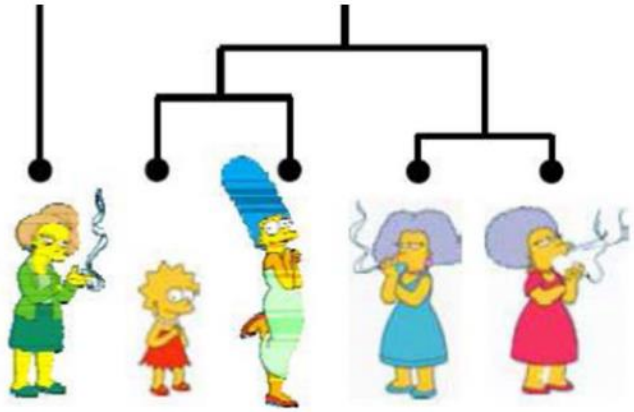
# Clustering

- Clustering groups similar data together into clusters.
- **Clustering** is the classification of objects into different groups according to some defined distance measure

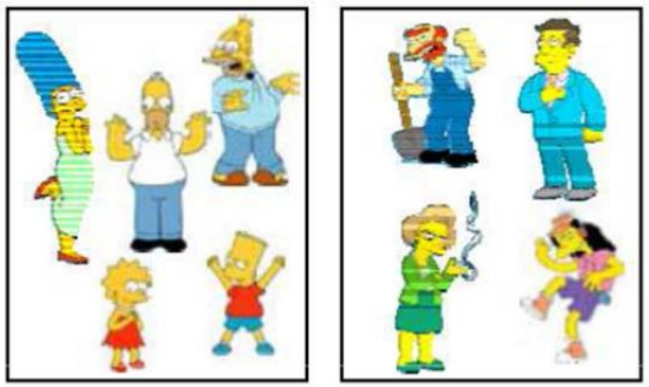


# Types of Clustering

- Hierarchical clustering
  - Agglomerative (bottom-up) algorithm
  - Divisive (top-down) algorithm



- Partitional clustering
  - K-means clustering
  - Fuzzy c-means clustering
  - QT clustering algorithm



# K-means Clustering

- Partitional clustering approach
- The k-means algorithm is an algorithm to cluster *n* objects or to group the objects based on attributes into *k* partitions, where k < n.
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, k, must be specified
- The grouping or clustering is done by minimizing the sum of squares of distances between the object and the corresponding cluster centroid.

# Distance Measures

- Distance measure will determine *how the similarity of two elements* is calculated and it will influence the shape of the clusters

1. Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

2. Manhattan Distance

$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p}|x_i - y_i|^2}$$

3. Hamming Distance

$$d^{HAD}(i,j) = \sum_{k=0}^{n-1}\left[y_{i,k} \neq y_{j,k}\right]$$

# Centroid

- The centroid (mean) of the cluster is defined as

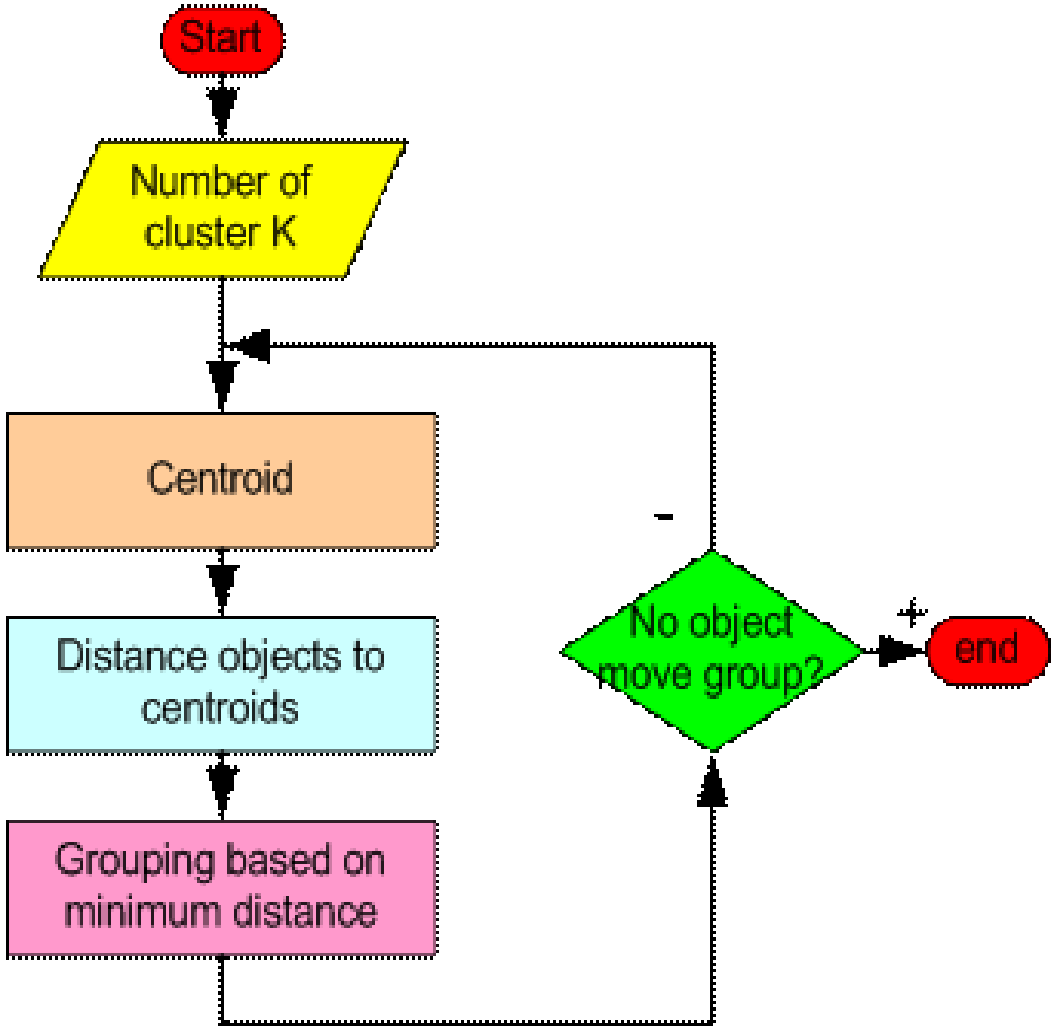
$$m(C) = \frac{1}{|C|} \sum_{x \in C} x$$

Where,

$$x + y = \sum_{i=1}^{n} x_i + y_i$$

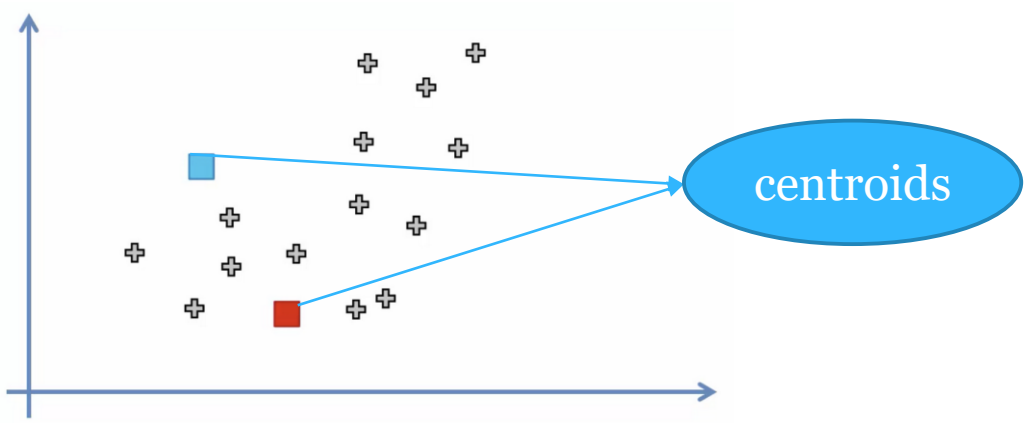
$$\frac{x}{|C|} = \sum_{i=1}^{n} \frac{x_i}{|C|}$$

# How K-means Clustering algorithm works?

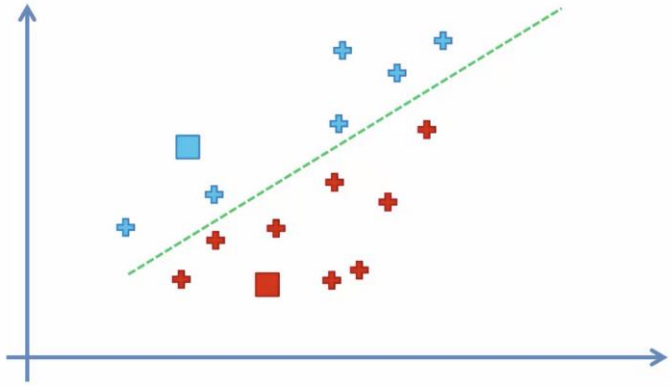


# K-means Clustering

1. Select K = 2 random points as cluster centers called centroids

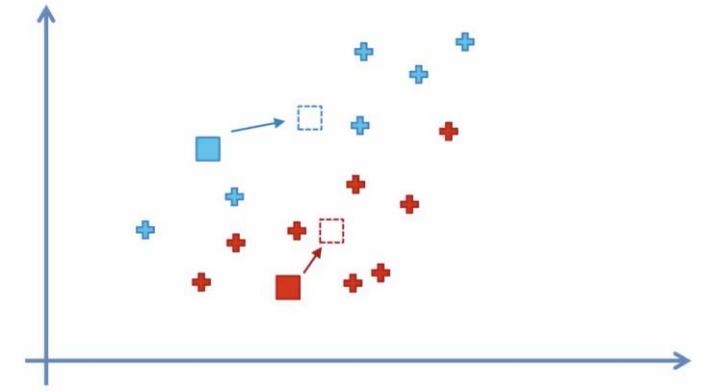



2. Assign each data point to the closest cluster by calculating its distance

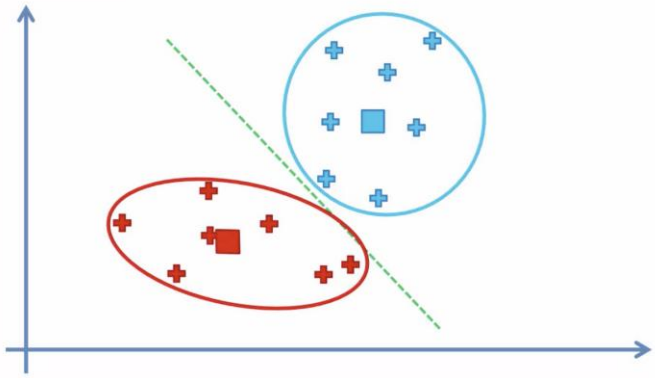


# K-means Clustering

3. Determine the new cluster center by computing the average of the assigned points



4. Repeat steps 2 and 3 until none of the cluster assignments change

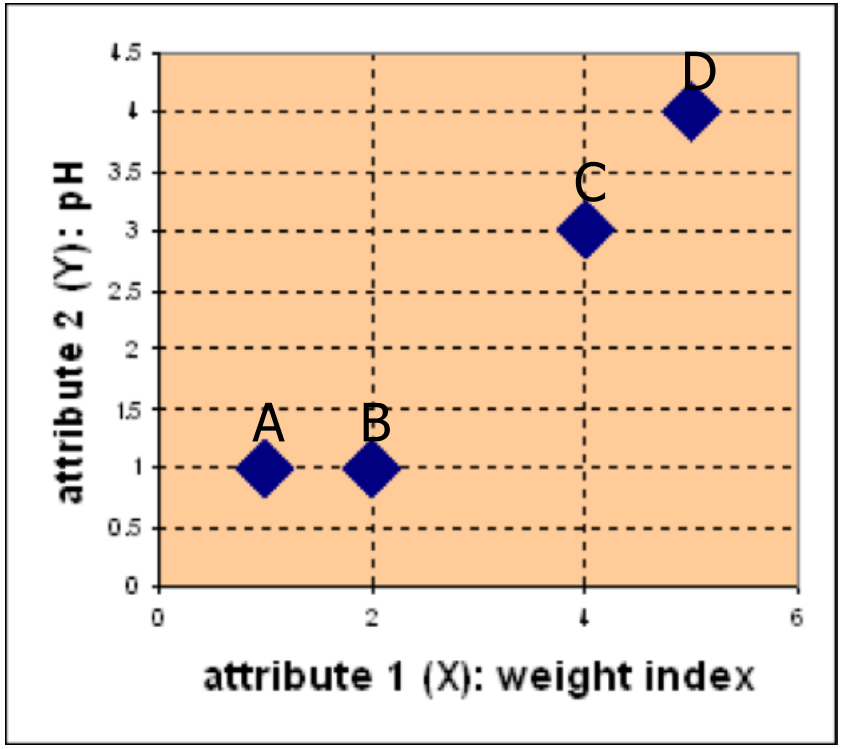


# Example: K-means Clustering

- We have 4 types of medicines and each has two attributes (pH and weight index)

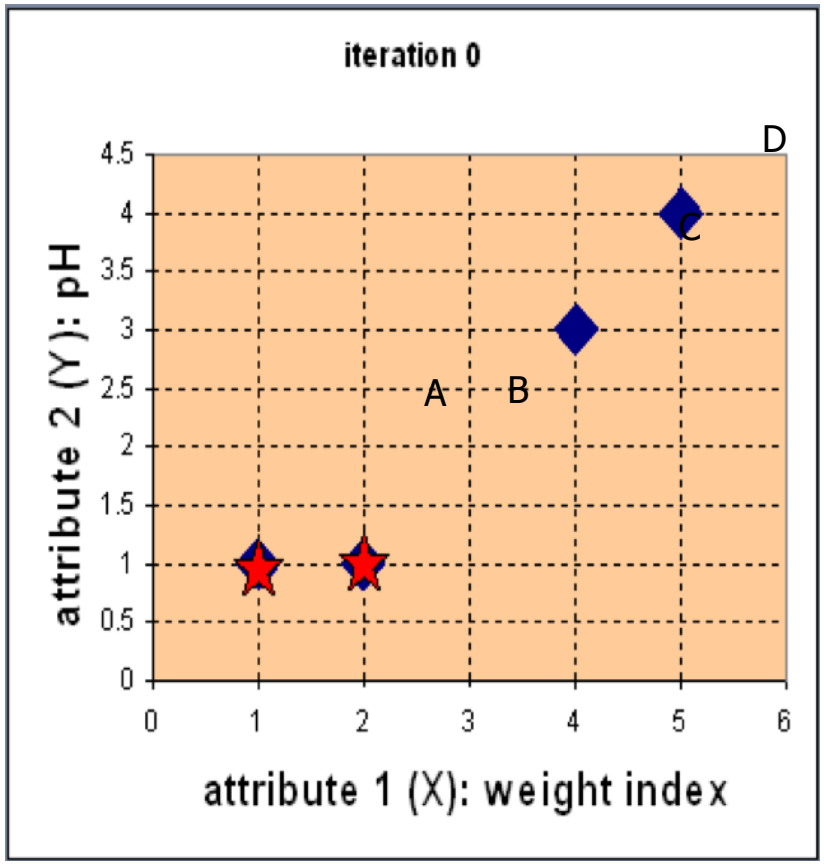
Our goal is to group these objects into *K=2* group of medicine

| Medicine | Weight | pH-Index |
|---|---|---|
| A | 1 | 1 |
| B | 2 | 1 |
| C | 4 | 3 |
| D | 5 | 4 |



# Example: K-means Clustering

- Step 1: Use initial seed points for partitioning




$$c_1 = A, c_2 = B$$

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (2,1) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

$$d(D,c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

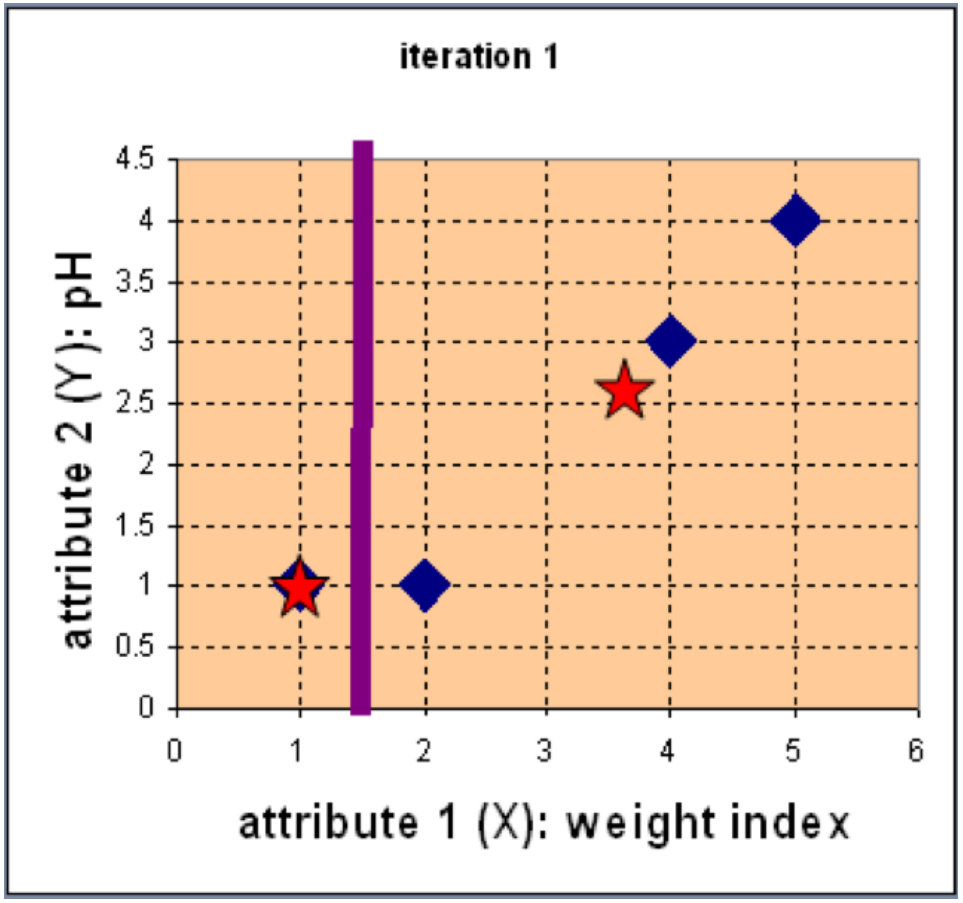
$$d(D,c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Euclidean distance

Assign each object to the cluster with the nearest seed point

# Example: K-means Clustering

- Step 2: Compute new centroids of the current partition




Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

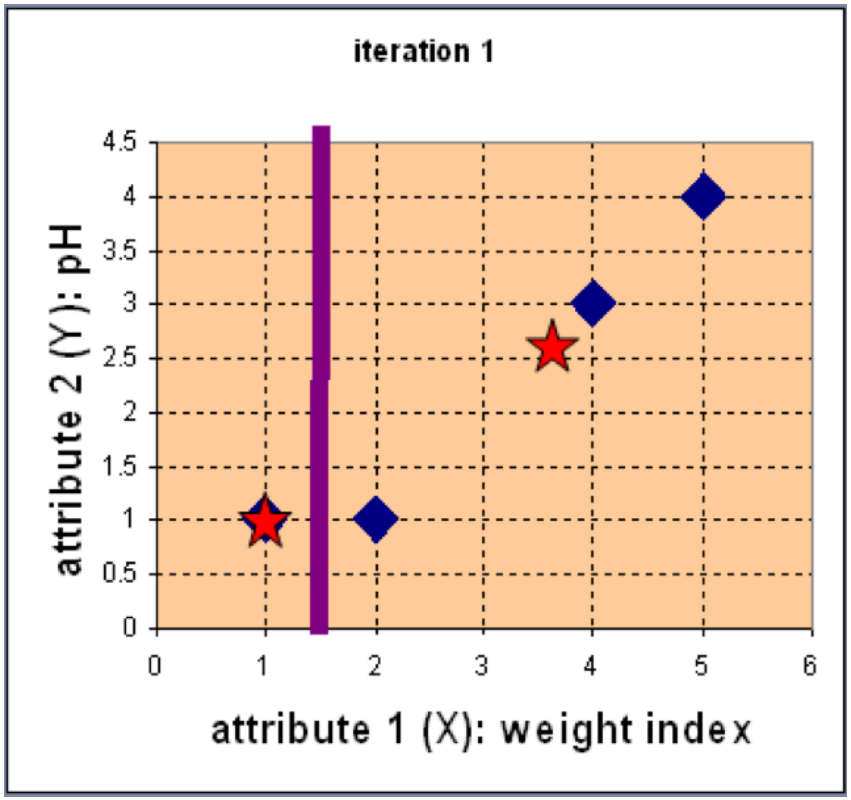
$$c_1 = (1,\ 1)$$

$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$

$$= \left(\frac{11}{3},\ \frac{8}{3}\right)$$

# Example: K-means Clustering

- Step 2: Renew membership based on new centroids




Compute the distance of all objects to the new centroids

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

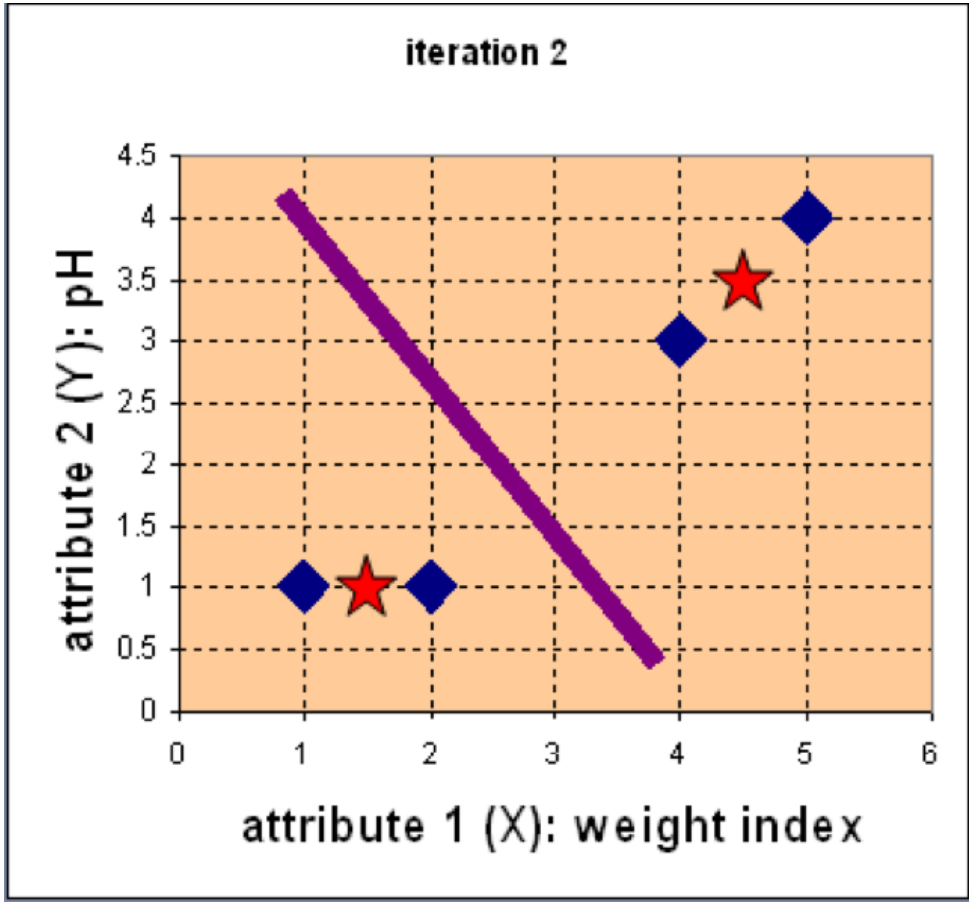
$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

Assign the membership to objects

# Example: K-means Clustering

- Step 3: Repeat the first two steps until its convergence



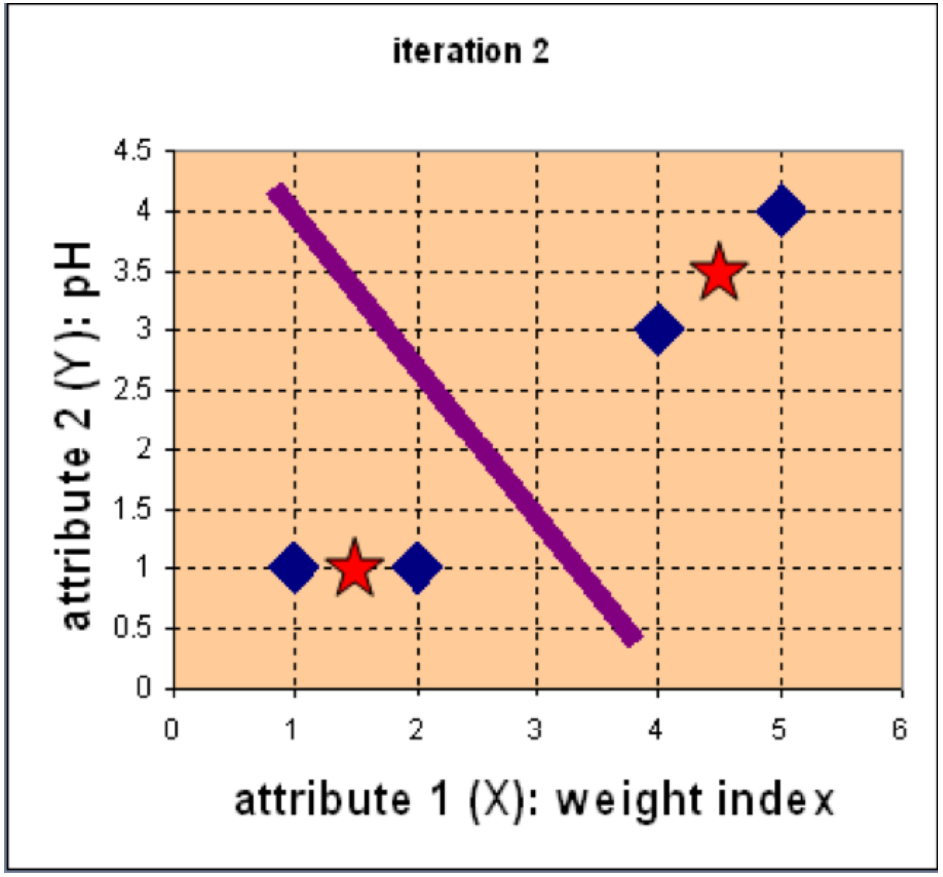

Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$$

# Example: K-means Clustering

- Step 3: Repeat the first two steps until its convergence




Compute the distance of all objects to the new centroids

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

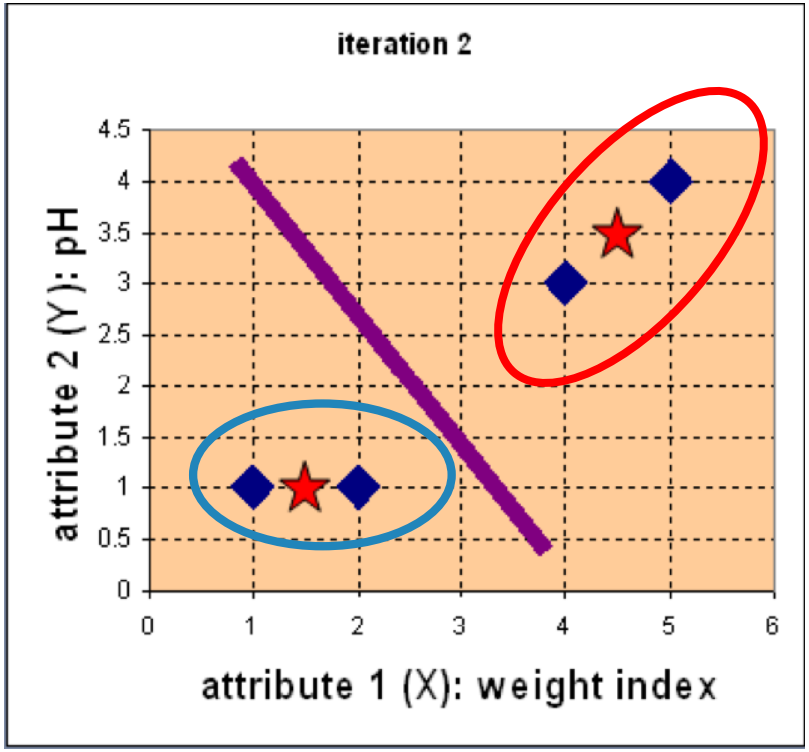
$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

Stop due to no new assignment
Membership in each cluster no longer change

- Step 4: The clusters obtained are

| Object | Feature1(X): weight index | Feature2 (Y): pH | Group (result) |
|---|---|---|---|
| Medicine A | 1 | 1 | 1 |
| Medicine B | 2 | 1 | 1 |
| Medicine C | 4 | 3 | 2 |
| Medicine D | 5 | 4 | 2 |



# Example: Implementation of k-means algorithm

| Individual | Variable 1 | Variable 2 |
|---|---|---|
| 1 | 1.0 | 1.0 |
| 2 | 1.5 | 2.0 |
| 3 | 3.0 | 4.0 |
| 4 | 5.0 | 7.0 |
| 5 | 3.5 | 5.0 |
| 6 | 4.5 | 5.0 |
| 7 | 3.5 | 4.5 |

**Step 1:**

Initialization: Randomly we choose following two centroids (k=2) for two clusters.

In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

| Individual | Variable 1 | Variable 2 |
|---|---|---|
| 1 | 1.0 | 1.0 |
| 2 | 1.5 | 2.0 |
| 3 | 3.0 | 4.0 |
| 4 | 5.0 | 7.0 |
| 5 | 3.5 | 5.0 |
| 6 | 4.5 | 5.0 |
| 7 | 3.5 | 4.5 |

| | Individual | Mean Vector |
|---|---|---|
| Group 1 | 1 | (1.0, 1.0) |
| Group 2 | 4 | (5.0, 7.0) |

CEIT
Think smart, be different.

## Step 2:

- Thus, we obtain two clusters containing:
  {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$m_1 = \left(\frac{1}{3}(1.0+1.5+3.0), \frac{1}{3}(1.0+2.0+4.0)\right) = (1.83, 2.33)$$

$$m_2 = \left(\frac{1}{4}(5.0+3.5+4.5+3.5), \frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$= (4.12, 5.38)$$

| Individual | Centroid 1 | Centroid 2 |
|---|---|---|
| 1 | 0 | 7.21 |
| 2 (1.5, 2.0) | 1.12 | 6.10 |
| 3 | 3.61 | 3.61 |
| 4 | 7.21 | 0 |
| 5 | 4.72 | 2.5 |
| 6 | 5.31 | 2.06 |
| 7 | 4.30 | 2.92 |

$$d(m_1, 2) = \sqrt{|1.0-1.5|^2 + |1.0-2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0-1.5|^2 + |7.0-2.0|^2} = 6.10$$

## Step 3:

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.

- Therefore, the new clusters are:
  {1,2} and {**3**,4,5,6,7}

- Next centroids are:
  m1=(1.25,1.5) and
  m2 = (3.9,5.1)

| Individual | Centroid 1 | Centroid 2 |
|---|---|---|
| 1 | 1.57 | 5.38 |
| 2 | 0.47 | 4.28 |
| 3 | 2.04 | 1.78 |
| 4 | 5.64 | 1.84 |
| 5 | 3.15 | 0.73 |
| 6 | 3.78 | 0.54 |
| 7 | 2.74 | 1.08 |

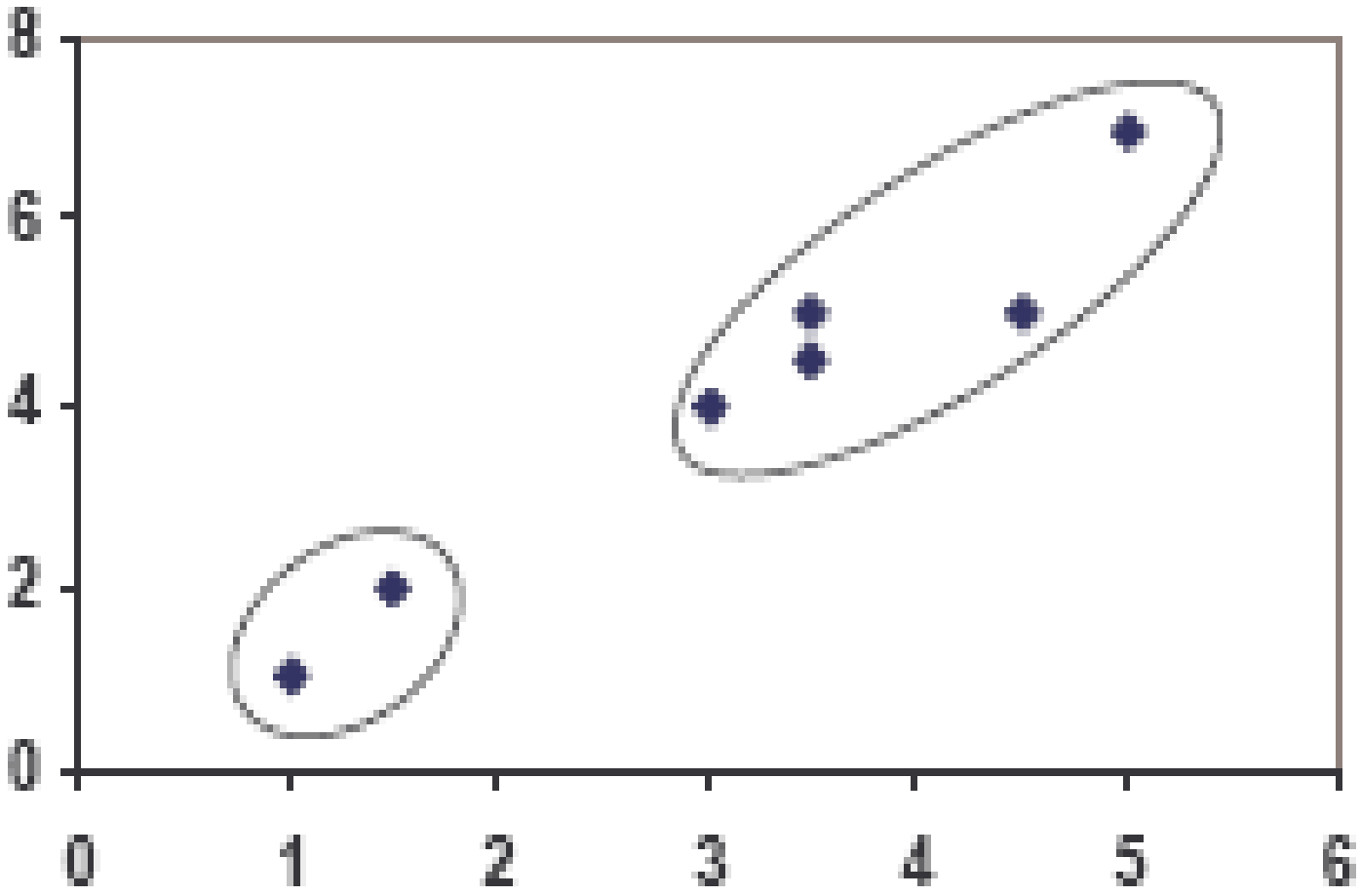
**Step 4 :**

The clusters obtained are: {1,2} and {3,4,5,6,7}

- Therefore, there is no change in the cluster.
- Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

| Individual | Centroid 1 | Centroid 2 |
|---|---|---|
| 1 | 0.56 | 5.02 |
| 2 | 0. 56 | 3.92 |
| 3 | 3.05 | 1.42 |
| 4 | 6.66 | 2.20 |
| 5 | 4.16 | 0.41 |
| 6 | 4.78 | 0.61 |
| 7 | 3.75 | 0.72 |

# PLOT



# Example: Implementation of k-means algorithm (k=3)

| Individual | $m_1 = 1$ | $m_2 = 2$ | $m_3 = 3$ | cluster |
|---|---|---|---|---|
| 1 | 0 | 1.11 | 3.61 | 1 |
| 2 | 1.12 | 0 | 2.5 | 2 |
| 3 | 3.61 | 2.5 | 0 | 3 |
| 4 | 7.21 | 6.10 | 3.61 | 3 |
| 5 | 4.72 | 3.61 | 1.12 | 3 |
| 6 | 5.31 | 4.24 | 1.80 | 3 |
| 7 | 4.30 | 3.20 | 0.71 | 3 |

$C_3$

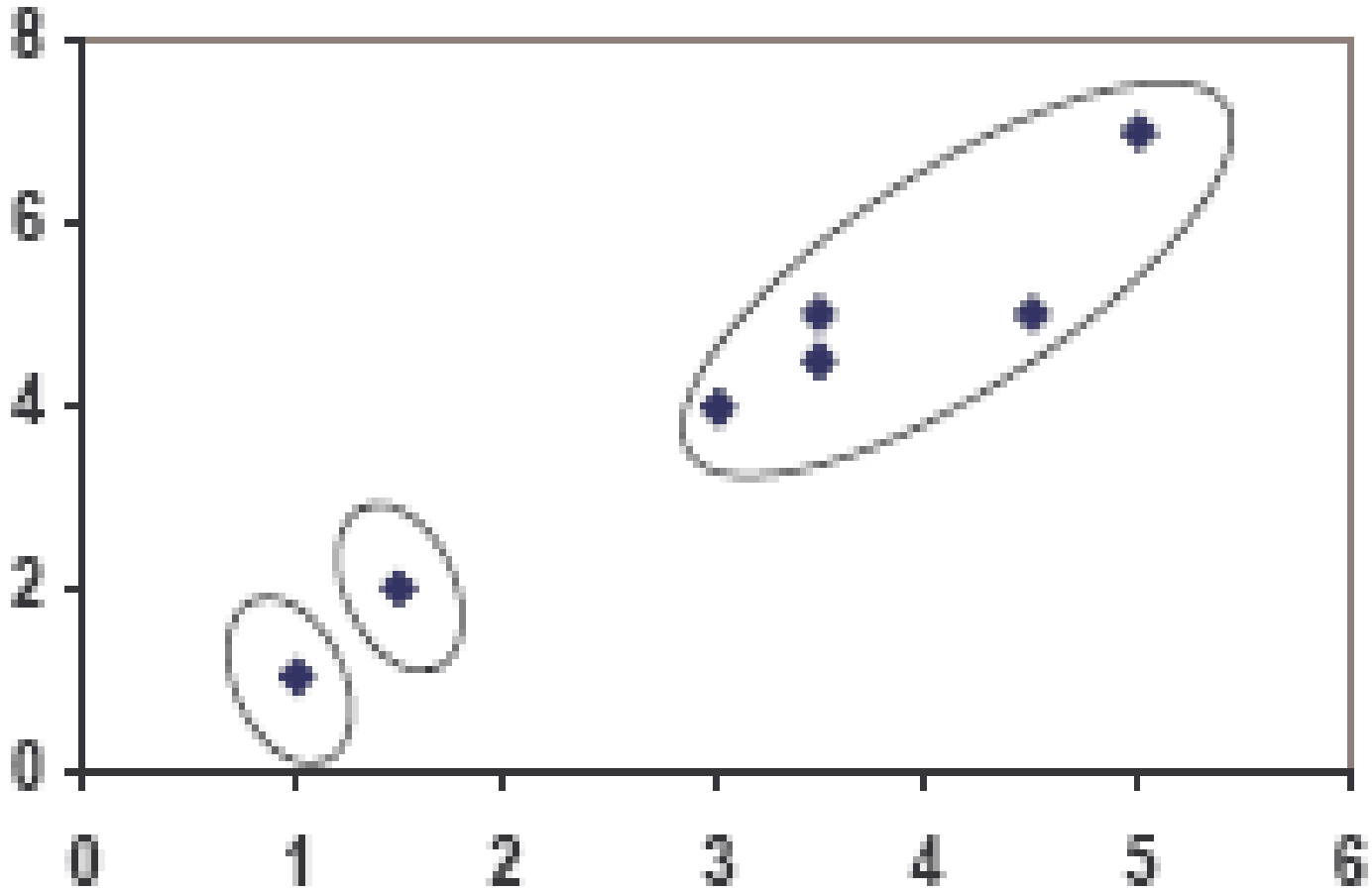
clustering with initial centroids (1, 2, 3)

**Step 1**

| Individual | $m_1$ (1.0, 1.0) | $m_2$ (1.5, 2.0) | $m_3$ (3.9,5.1) | cluster |
|---|---|---|---|---|
| 1 | 0 | 1.11 | 5.02 | 1 |
| 2 | 1.12 | 0 | 3.92 | 2 |
| 3 | 3.61 | 2.5 | 1.42 | 3 |
| 4 | 7.21 | 6.10 | 2.20 | 3 |
| 5 | 4.72 | 3.61 | 0.41 | 3 |
| 6 | 5.31 | 4.24 | 0.61 | 3 |
| 7 | 4.30 | 3.20 | 0.72 | 3 |

**Step 2**

# PLOT



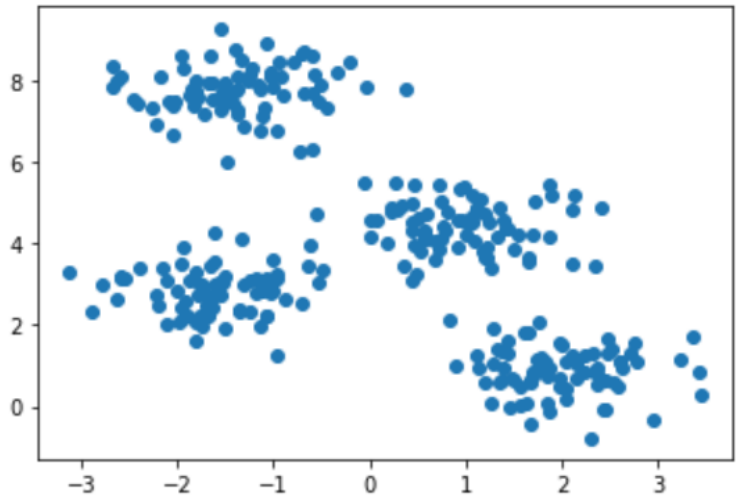
# Practical: Python Coding Environment

- Classification data using the k-means algorithm

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets.samples_generator import make_blobs
from sklearn.cluster import KMeans
```

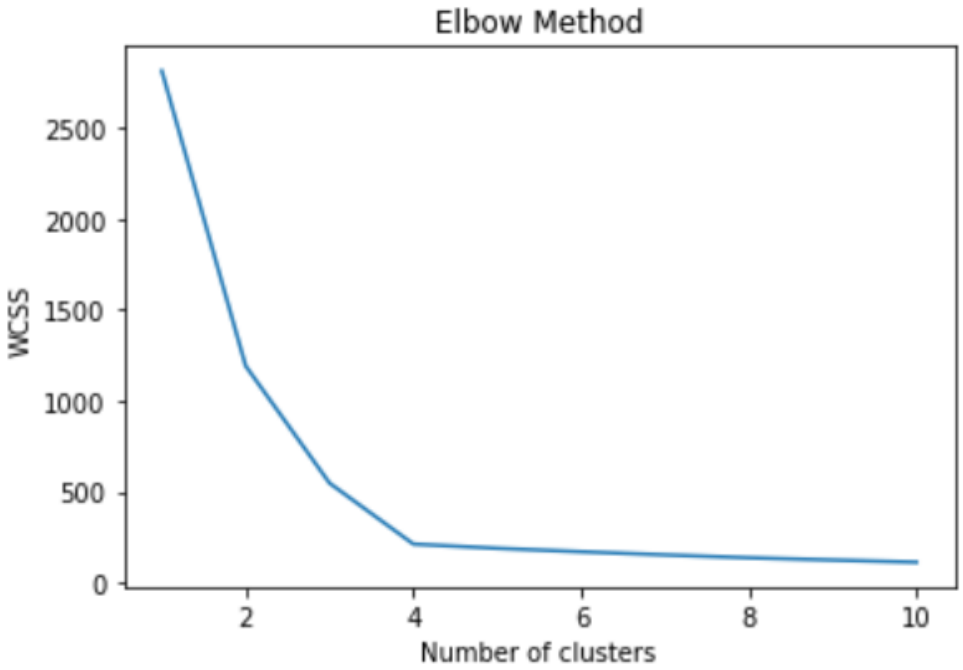
```
X, y = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)
plt.scatter(X[:,0], X[:,1])
```

```
<matplotlib.collections.PathCollection at 0x214766cdc88>
```



# Practical: Python Coding Environment

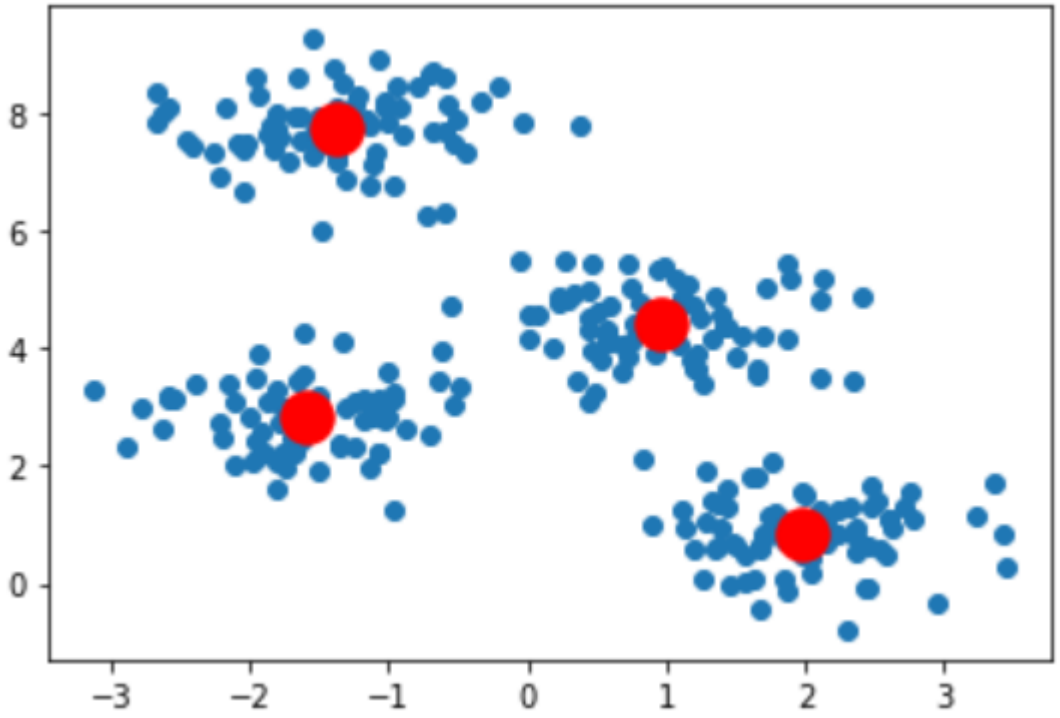
```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



# Practical: Python Coding Environment

```
kmeans = KMeans(n_clusters=4, init='k-means++', max_iter=300, n_init=10, random_state=0)
pred_y = kmeans.fit_predict(X)
plt.scatter(X[:,0], X[:,1])

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red')
plt.show()
```



# Applications of K-means clustering

- Image Segmentation
- Clustering Gene Segmentation Data
- News Articles Clustering
- Clustering Languages
- Species Clustering
- Anomaly Detection

# References


- Tutorial - Tutorial with introduction of Clustering Algorithms (k-means, fuzzy-c-means, hierarchical, mixture of gaussians)
- H. Zha, C. Ding, M. Gu, X. He and H.D. Simon. "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.
- J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.
- https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203
- https://mubaris.com/posts/kmeans-clustering/

# Next Week Lecture

- Dimensionality Reduction: Principal Component Analysis (PCA)

Thank You