# K-MEANS CLUSTERING

# INTRODUCTION-
# What is clustering?

- **Clustering** is the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often according to some defined distance measure.

# Types of clustering:

1. **Hierarchical algorithms**: these find successive clusters using previously established clusters.
   1. Agglomerative ("bottom-up"): Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters.
   2. Divisive ("top-down"): Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.
2. **Partitional clustering**: Partitional algorithms determine all clusters at once. They include:
   - ***K*-means and derivatives**
   - Fuzzy *c*-means clustering
   - QT clustering algorithm

# Common Distance measures:

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

  They include:

1. The Euclidean distance (also called 2-norm distance) is given by:

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

2. The Manhattan distance (also called taxicab norm or 1-norm) is given by:

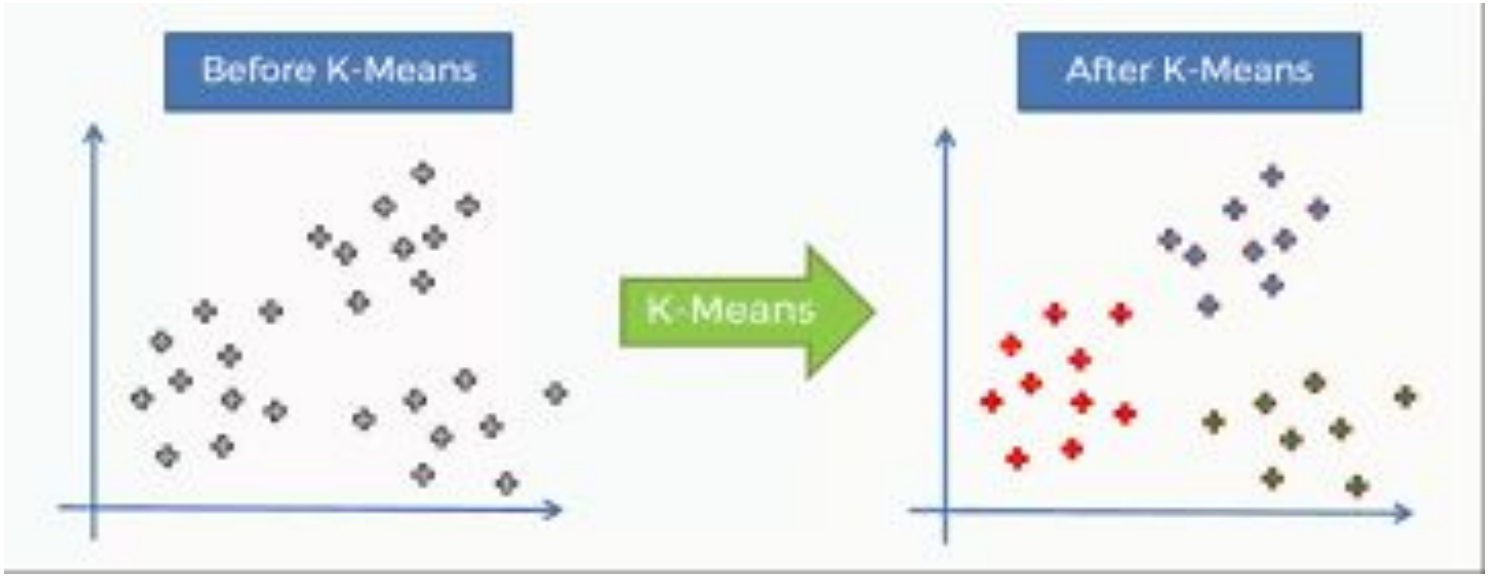
$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$

# K-MEANS CLUSTERING

- The **k-means algorithm** is an algorithm to cluster *n* objects based on attributes into *k* partitions, where $k < n$.

- It assumes that the object attributes form a vector space.

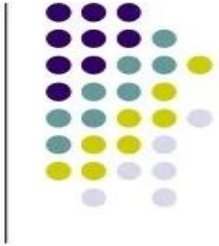
K-Means
-Simplest unsupervised clustering algorithm
-clusters unlabelled data into specified number of clusters
-Clustering groups the similar data points
-Used in targeted marketing eg:Amazon



- An algorithm for partitioning (or clustering) N data points into K disjoint subsets $S_j$ containing data points so as to minimize the sum-of-squares criterion

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \left\| x_n - \mu_j \right\|^2,$$

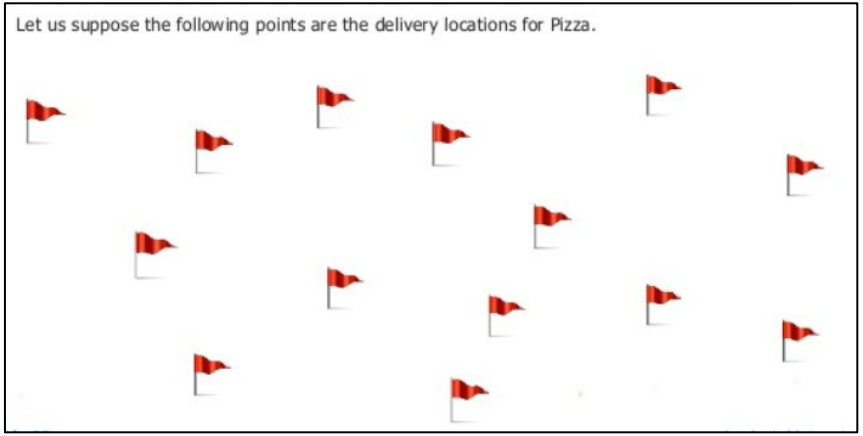
where $x_n$ is a vector representing the the $n^{th}$ data point and $u_j$ is the geometric centroid of the data points in $S_j$.

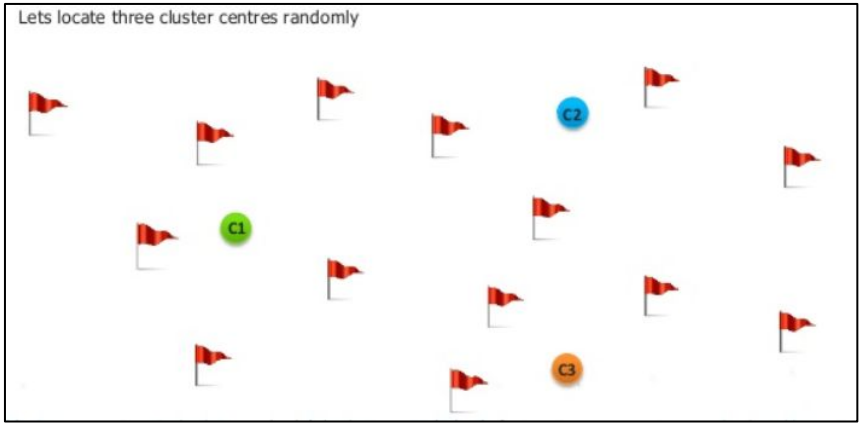


- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

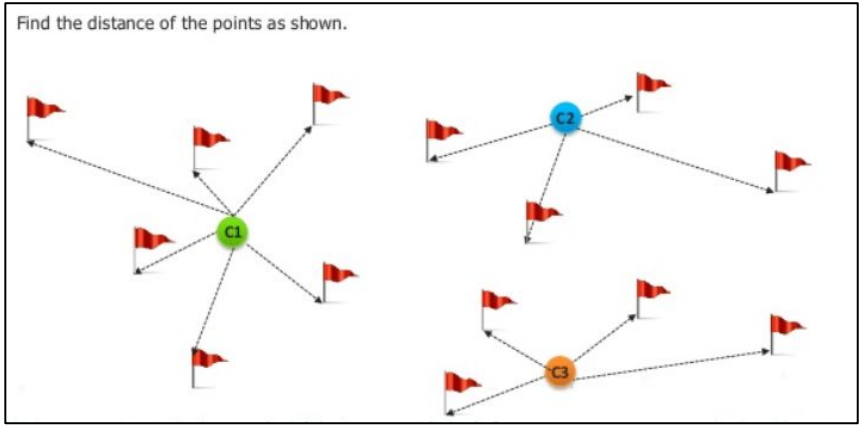
# Case study
# Establish Pizza delivery centers


Reference:
https://www.slideshare.net/EdurekaIN/applications-of-clustering-in-real-life

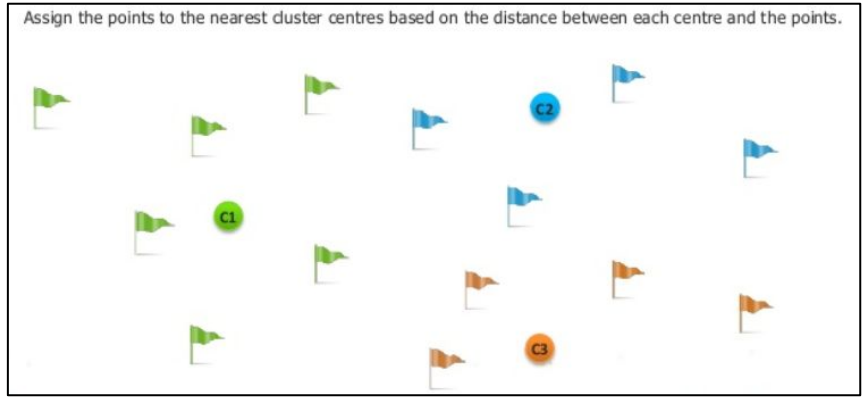


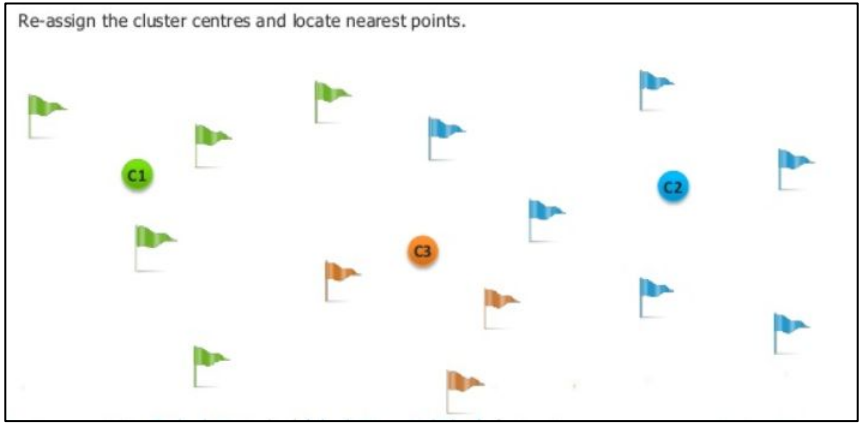

**Step 1:** Decide the number of clusters ‘K’



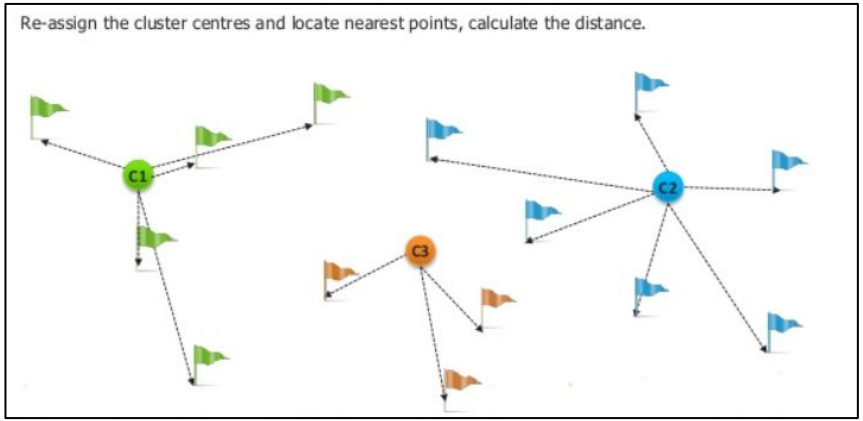

**Step 2:** Calculate distance between each data point and cluster centers

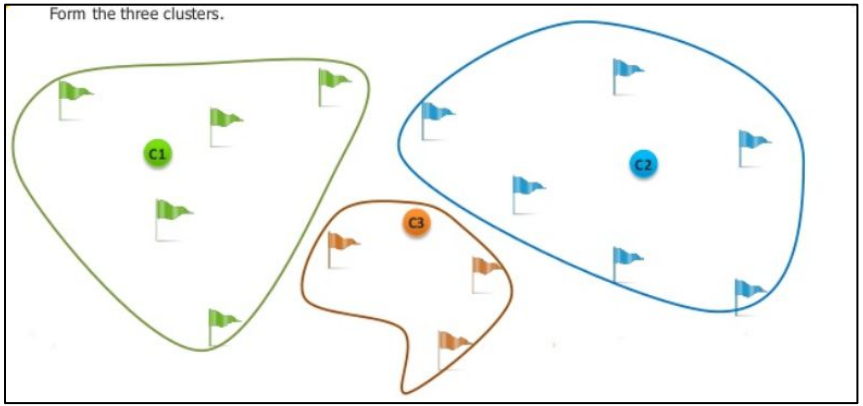



**Step 3:** Assign the data points to the nearest cluster cente



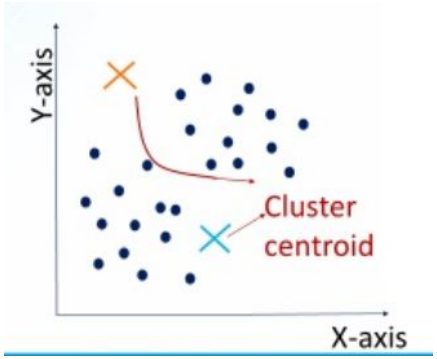

**Step 4:** Calculate the new center for each cluster

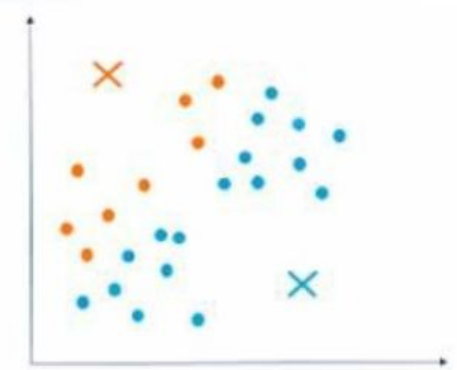


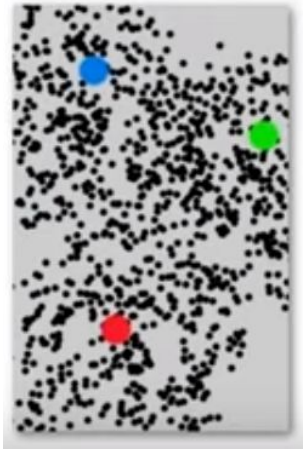
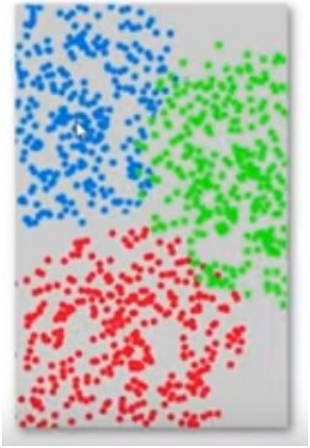
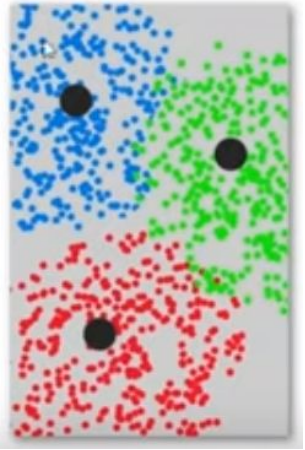
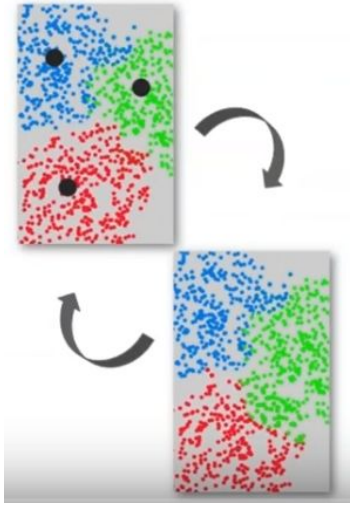




**Step 5:** Repeat step 3 and 4

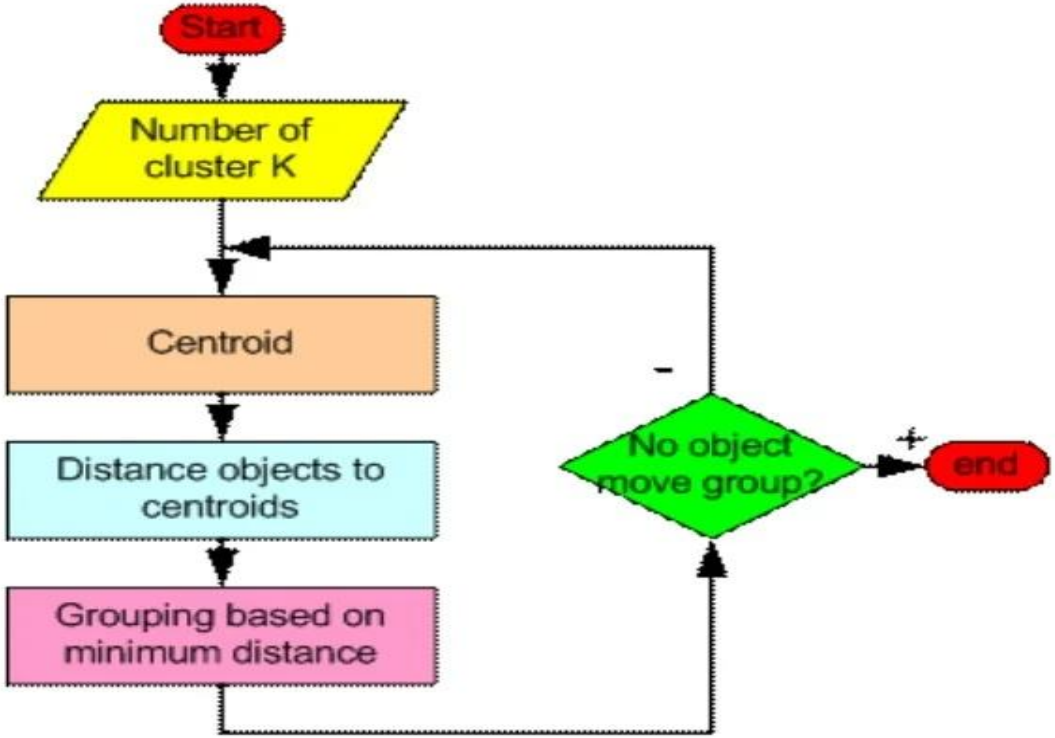
- K-means, K denotes the number of clusters



# How the K-Mean Clustering algorithm works?



**K-means can be summarized as below:**

Let x = $[x_1, x_2, x_3 \ldots x_n]$ be the data points and C = $[c_1, c_2, c_3 \ldots c_K]$ be the set of cluster centers

The procedure of K-means clustering algorithm is detailed as follows:

- **Step 1:** Decide the number of clusters 'K' and randomly select 'K' cluster centers
- **Step 2:** Calculate distance between each data point and cluster centers
- **Step 3:** Assign the data points to the nearest cluster centers based on its minimum Euclidean distance to the cluster center
- **Step 4:** Calculate the new center for each cluster by computing the average of all the data points belonging to that cluster. The center $(c_i)$ of the $i^{th}$ cluster in the K-means algorithm is represented as the arithmetic mean of the data points present in that cluster.

$$c_i = \frac{1}{T_i} \sum_{x_i \epsilon T_i} x_i$$

  - $T_i$ is the total number of data points present in the $i^{th}$ cluster.
- **Step 5:** Repeat step 3 and 4 until the global cluster centers are reached.

**K-means can be summarized as below:**

Let x = $[x_1, x_2, x_3 \dots x_n]$ be the data points and C = $[c_1, c_2, c_3 \dots c_K]$ be the set of cluster centers

The procedure of K-means clustering algorithm is detailed as follows:

- **Step 1:** Decide the number of clusters 'K' and randomly select 'K' cluster centers
- **Step 2:** Calculate distance between each data point and cluster centers
- **Step 3:** Assign the data points to the nearest cluster centers based on its minimum Euclidean distance to the cluster center
- **Step 4:** Calculate the new center for each cluster by computing the average of all the data points belonging to that cluster. The center $(c_i)$ of the $i^{th}$ cluster in the K-means algorithm is represented as the arithmetic mean of the data points present in that cluster.

$$c_i = \frac{1}{T_i} \sum_{x_i \epsilon T_i} x_i$$

  - $T_i$ is the total number of data points present in the $i^{th}$ cluster.

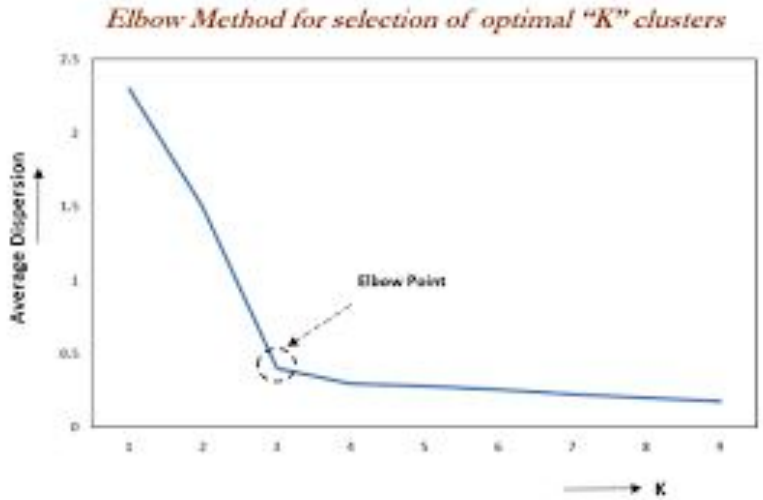
- Step 5: Repeat step 3 and 4 until convergence is reached

- **Step 1:** Begin with a decision on the value of k = number of clusters .
- **Step 2**: Put any initial partition that classifies the data into k clusters. You may assign the training samples randomly,or systematically as the following:
  1. Take the first k training sample as single-element clusters
  2. Assign each of the remaining (N-k) training sample to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

- **Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
- **Step 4 .** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

How to decide on the value of K

-Using Elbow method
-if we increase the value of k from the elbow point ,the distortion remains constant
-This is the ideal value for cluster creation



# A Simple example showing the implementation of k-means algorithm (using K=2)

| 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) |

| Subject | A | B |
|---|---|---|
| 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 (centroid) |
|---|---|---|
| Group 1 | 1 | (1.0, 1.0) |
| Group 2 | 4 | (5.0, 7.0) |

**Step 1:** Take two random cluster centers

| Subject | dist(x,c1) [c1 = 1.0,1.0] | dist(x,c2) [c2 = 5.0,7.0] |
|---|---|---|
| 1 (1,1) | $\sqrt{(1-1)^2+(1-1)^2} = 0$ | $\sqrt{(1-5)^2+(1-7)^2} = 7.2$ |
| 2 (1.5,2) | $\sqrt{(1.5-1)^2+(2-1)^2} = 1.12$ | $\sqrt{(1.5-5)^2+(2-7)^2} = 6.1$ |
| 3 (3,4) | $\sqrt{(3-1)^2+(4-1)^2} = 3.6$ | $\sqrt{(3-5)^2+(4-7)^2} = 3.6$ |
| 4 (5,7) | $\sqrt{(5-1)^2+(7-1)^2} = 7.21$ | $\sqrt{(5-5)^2+(7-7)^2} = 0$ |
| 5 (3.5,5) | $\sqrt{(3.5-1)^2+(5-1)^2} = 4.72$ | $\sqrt{(3.5-5)^2+(5-7)^2} = 2.5$ |
| 6 (4.5,5) | $\sqrt{(4.5-1)^2+(5-1)^2} = 5.32$ | $\sqrt{(4.5-5)^2+(5-7)^2} = 2.06$ |
| 7 (3.5,4.5) | $\sqrt{(3.5-1)^2+(4.5-1)^2} = 4.3$ | $\sqrt{(3.5-5)^2+(4.5-7)^2} = 2.91$ |

**Step 2: C**alculate the distance between data points to the cluster center

| Subject | dist(x,c1) [c1 = 1.0,1.0] | dist(x,c2) [c2 = 5.0,7.0] | Class |
|---|---|---|---|
| 1 (1,1) | $\sqrt{(1-1)^2+(1-1)^2}$ = **0** | $\sqrt{(1-5)^2+(1-7)^2}$ = 7.2 | C1 |
| 2 (1.5,2) | $\sqrt{(1.5-1)^2+(2-1)^2}$ = **1.12** | $\sqrt{(1.5-5)^2+(2-7)^2}$ = 6.1 | C1 |
| 3 (3,4) | $\sqrt{(3-1)^2+(4-1)^2}$ = **3.6** | $\sqrt{(3-5)^2+(4-7)^2}$ = 3.6 | C1 |
| 4 (5,7) | $\sqrt{(5-1)^2+(7-1)^2}$ = 7.21 | $\sqrt{(5-5)^2+(7-7)^2}$ = **0** | C2 |
| 5 (3.5,5) | $\sqrt{(3.5-1)^2+(5-1)^2}$ = 4.72 | $\sqrt{(3.5-5)^2+(5-7)^2}$ = **2.5** | C2 |
| 6 (4.5,5) | $\sqrt{(4.5-1)^2+(5-1)^2}$ = 5.32 | $\sqrt{(4.5-5)^2+(5-7)^2}$ = **2.06** | C2 |
| 7 (3.5,4.5) | $\sqrt{(3.5-1)^2+(4.5-1)^2}$ = 4.3 | $\sqrt{(3.5-5)^2+(4.5-7)^2}$ = **2.91** | C2 |

| | Individual |
|---|---|
| Cluster 1 | 1, 2, 3 |
| Cluster 2 | 4, 5, 6, 7 |

**Step 3:** Assign the data points to the nearest cluster centers

| | Data points | Average |
|---|---|---|
| Cluster 1 | (1,1)<br>(1.5,2)<br>(3,4) | (1+1.5+3)/3<br>(1+2+4)/3 |
| Cluster 2 | (5,7)<br>(3.5,5)<br>(4.5,5)<br>(3.5,4.5) | (5+3.5+4.5+3.5)/4<br>(7+5+5+4.5)/4 |

| | Individual | Mean Vector (centroid) |
|---|---|---|
| Cluster 1 | 1, 2, 3 | (1.8, 2.3) |
| Cluster 2 | 4, 5, 6, 7 | (4.1, 5.4) |

**Step 4:** Calculate the new center for each cluster

Step 5:Repeat step 3 and 4 until global cluster centers are reached

## Step 2:

- Thus, we obtain two clusters containing:

  {1,2,3} and {4,5,6,7}.
- Their new centroids are:

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

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

$$= (4.12, 5.38)$$

| 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.

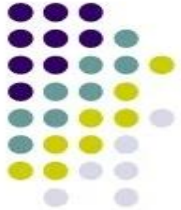
- Therefore, the new clusters are:
  {1,2} and {**3**,4,5,6,7}

- 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 |

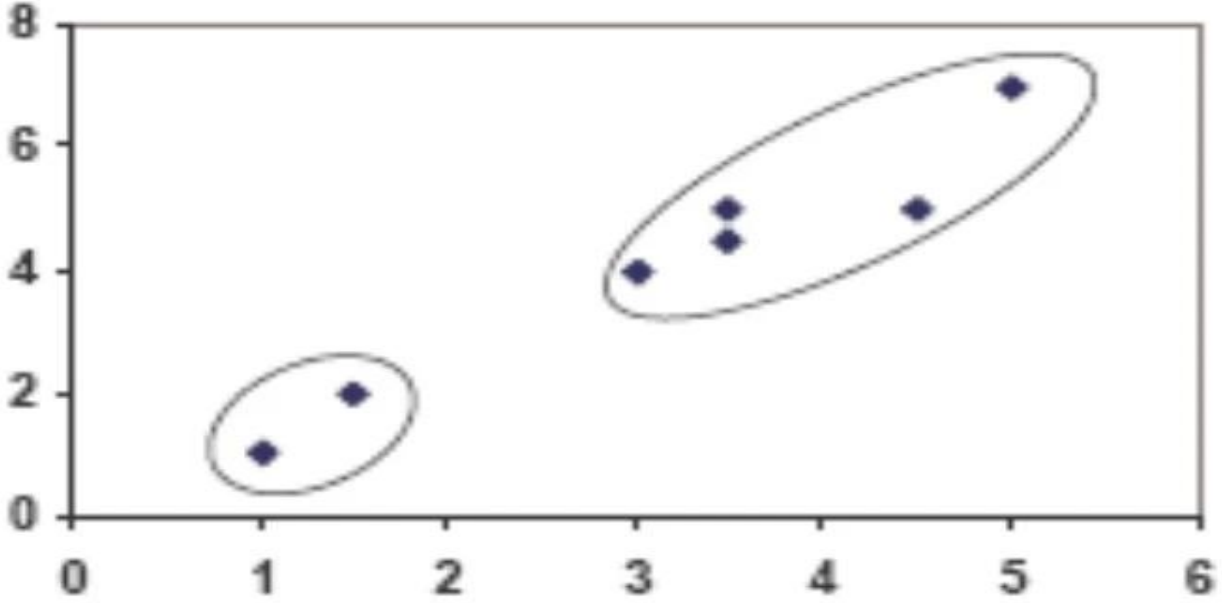
- Step 4 :
  The clusters obtained are:
  {1,2} and {3,4,5,6,7}

- 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



# (with 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 |

$C_3$

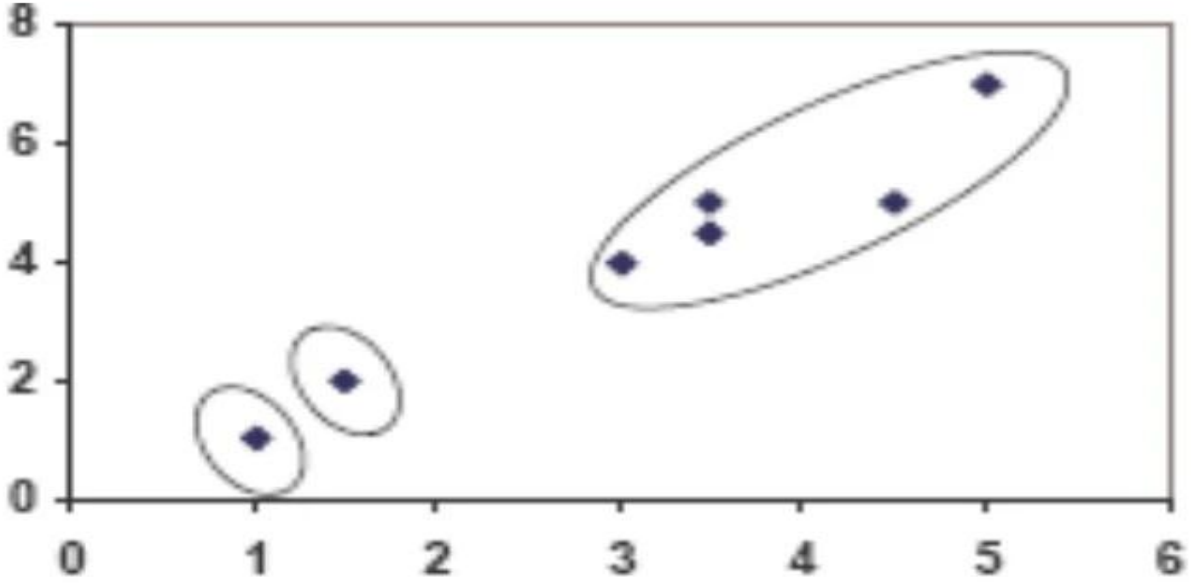
clustering with initial centroids (1, 2, 3)

**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



# Real-Life Numerical Example of K-Means Clustering

We have 4 medicines as our training data points object and each medicine has 2 attributes. Each attribute represents coordinate of the object. We have to determine which medicines belong to cluster 1 and which medicines belong to the other cluster.

| Object | Attribute1 (X): weight index | Attribute 2 (Y): pH |
|---|---|---|
| Medicine A | 1 | 1 |
| Medicine B | 2 | 1 |
| Medicine C | 4 | 3 |
| Medicine D | 5 | 4 |

# Weaknesses of K-Mean Clustering

1. When the numbers of data are not so many, initial grouping will determine the cluster significantly.
2. The number of cluster, K, must be determined before hand. Its disadvantage is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments.
3. We never know the real cluster, using the same data, because if it is inputted in a different order it may produce different cluster if the number of data is few.
4. It is sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the *local optimum*.

# Applications of K-Mean Clustering

- It is relatively *efficient and fast.* It computes result at **O(tkn),** where n is number of objects or points, k is number of clusters and t is number of iterations.
- k-means clustering can be applied to *machine learning or data mining*
- *Used on acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation).*
- *Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization.*

# CONCLUSION

- *K-means algorithm is* useful for undirected knowledge discovery and is relatively simple. K-means has found wide spread usage in lot of fields, ranging from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligence, image processing, machine vision, and many others.

# Thank You