# Clustering Algorithms

# Machine learning:

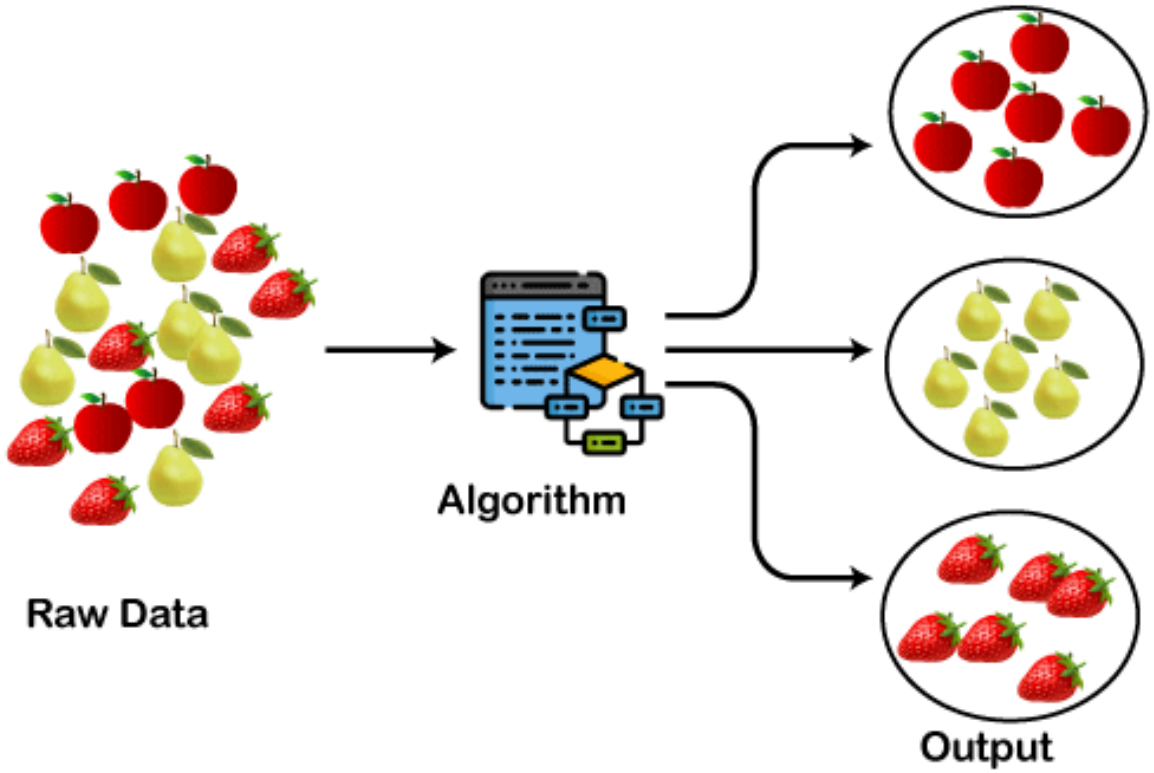
- **Supervised vs Unsupervised.**
  - ***Supervised learning*** - the presence of the outcome variable is available to guide the learning process.
    - there **must** be a training data set in which the solution is already known.
  - ***Unsupervised learning*** - the outcomes are unknown.
    - cluster the data to reveal meaningful partitions and hierarchies

# Clustering:

- Clustering is the task of gathering samples into groups of similar samples according to some predefined similarity or dissimilarity measure.
- Clustering is a kind of Unsupervised learning: no predefined classes (i.e., *learning by observations*)

sample

Cluster/group


Raw Data
Algorithm
Output

Clustering is an unsupervised machine learning technique that divides the population or data points into several groups or clusters such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

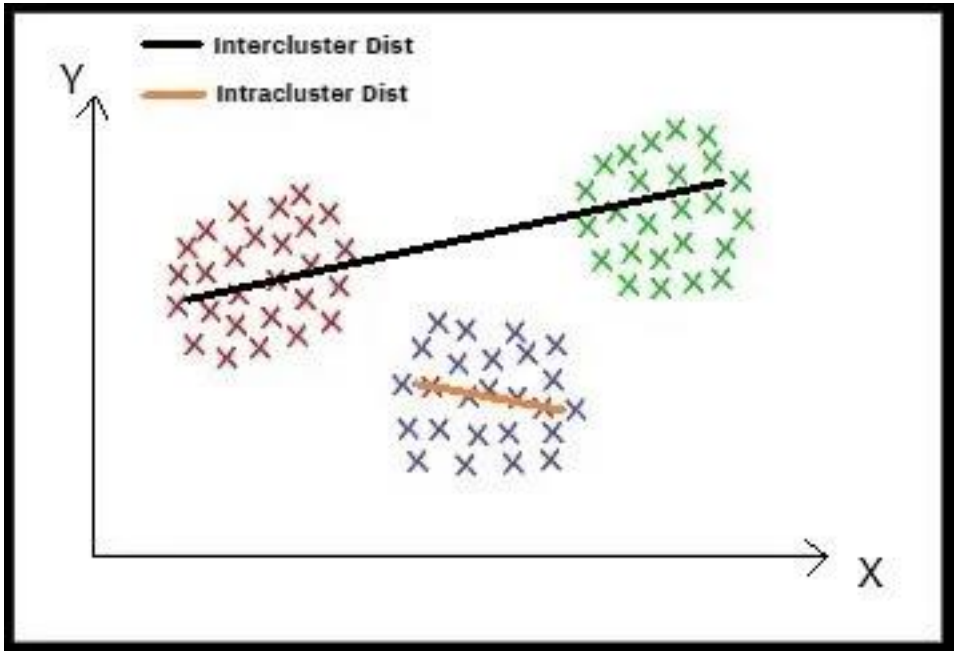
- Points in the same cluster are closer to each other.
- Points in the different clusters are far apart.

The primary aim is to maximize the similarity within clusters while minimizing the similarity between clusters.

Clustering is considered to be best if it has maximum intercluster distance and minimum intracluster distance.

**Intracluster distance:** Distance between two point in the same cluster.

**Intercluster distance:** Distance between two points in the different clusters.



## Why Clustering:

Clustering is an essential task which determines the intrinsic grouping among the unlabeled data present.

There are no criteria for good clustering. It depends on the user, what is the criteria they may use which satisfy their need

The clustering algorithm must make some assumptions that constitute the similarity of points and each assumption make different and equally valid clusters.

# What is Cluster Analysis?

Cluster: A collection of data objects

similar (or related) to one another within the same group

dissimilar (or unrelated) to the objects in other groups

Cluster analysis (or *clustering*, *data segmentation, …*)

Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

# Clustering Applications

**Biology:** taxonomy of living things: kingdom, phylum, class, order, family, genus and species

**Information retrieval**: document clustering

**Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

**City-planning**: Identifying groups of houses according to their house type, value, and geographical location

**Social Network Analysis**:- Various Community

**Image Segmentation**

# Quality: What Is Good Clustering?

A good clustering method will produce high quality clusters

high intra-class similarity: cohesive within clusters

low inter-class similarity: distinctive between clusters

The quality of a clustering method depends on the similarity measure used by the method its implementation, and Its ability to discover some or all of the hidden patterns.

# Measure the Quality of Clustering

Dissimilarity/Similarity metric

- Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables
- Weights should be associated with different variables based on applications and data semantics

Quality of clustering:

- There is usually a separate "quality" function that measures the "goodness" of a cluster.
- It is hard to define "similar enough" or "good enough"
  - The answer is typically highly subjective

# Major Clustering Approaches

## Partitioning approach:

Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors

Typical methods: k-means, k-medoids, CLARANS

## Hierarchical approach:

Create a hierarchical decomposition of the set of data (or objects) using some criterion

Typical methods: Diana, Agnes, BIRCH, CAMELEON

## Density-based approach:

Based on connectivity and density functions

Typical methods: DBSACN

# K-MEANS CLUSTERING

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

- 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.

- K-Means algorithm is a centroid based clustering technique.

- Each cluster in k-means clustering algorithm is represented by a centroid point.

# Centroid point

The centroid point is the point that represents its cluster. Centroid point is the average of all the points in the set and will change in each step and will be computed by:

$$C_i = \frac{1}{||S_i||} \sum_{x_j \epsilon S_i} x_j$$

For the above equation,

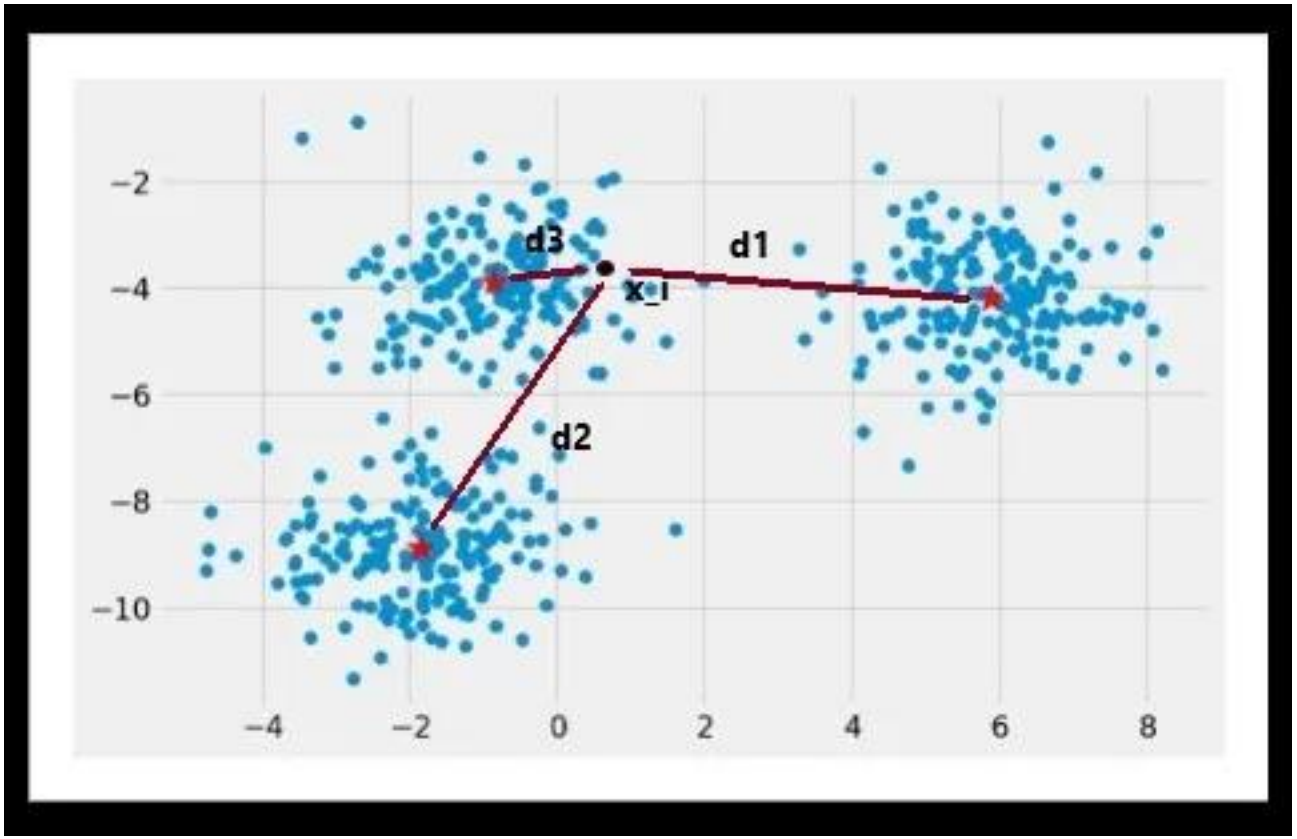
C_i: i'th Centroid

S_i: All points belonging to set_i with centroid as C_i

x_j: j'th point from the set

||S_i||: number of points in set_i

The idea of the K-Means algorithm is to find k-centroid points and every point in the dataset will belong either of k-sets having minimum Euclidean distance.



# The *K-Means* Clustering Method

Given *k*, the *k-means* algorithm is implemented in four steps:

**Steps #1: Initialization:**

The initial k-centroids are randomly picked from the dataset of points

**Steps #2: Assignment:**

For each point in the dataset, find the Euclidean distance between the point and all centroids. The point will be assigned to the cluster with the nearest centroid. Assign each object to the cluster with the nearest seed point

# The *K-Means* Clustering Method

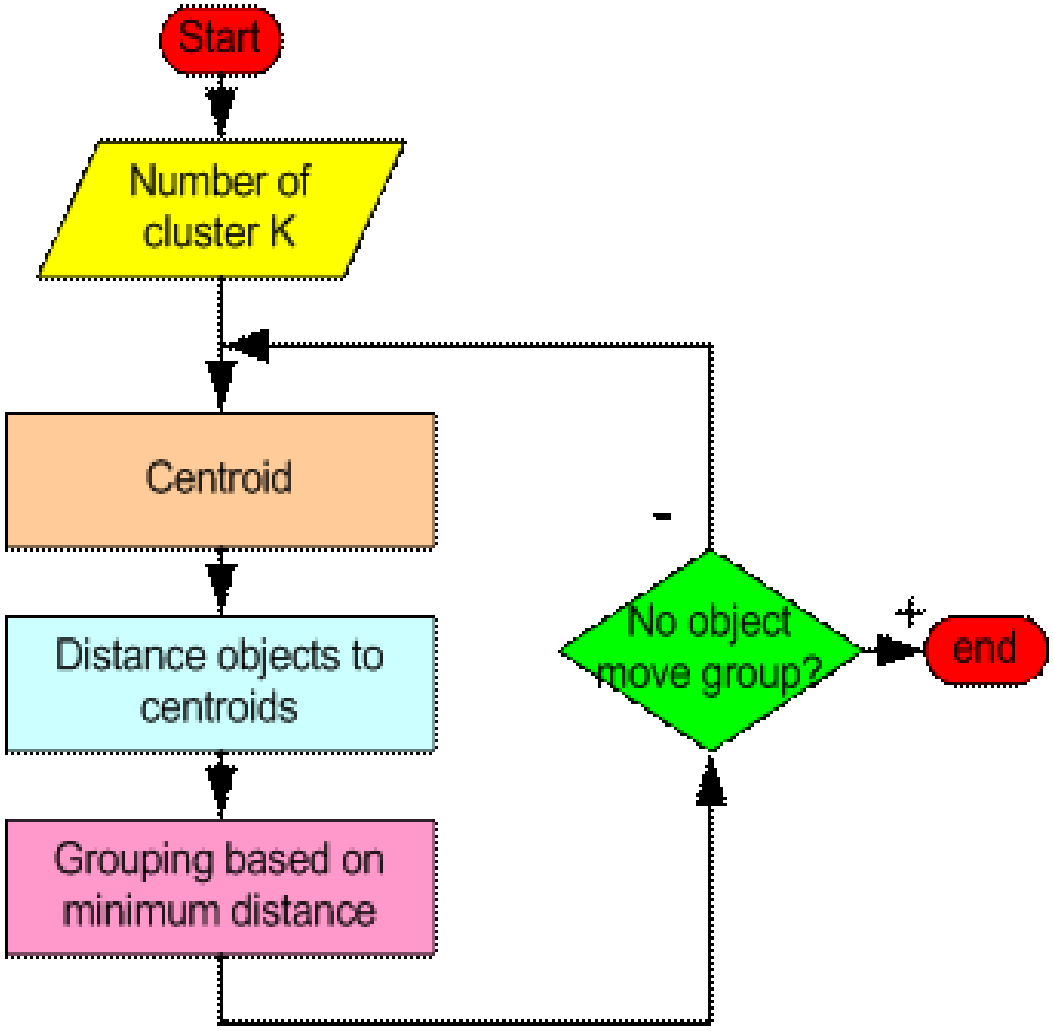
**Steps #3: Updation of Centroid:**

Update the value of the centroid with the new mean value

**Steps #4: Repeat:**

Repeat steps 2 and 3 unless convergence is achieved. If convergence is achieved then stop. Convergence refers to the condition where the previous value of centroids is equal to the updated value.

How K-means clustering works

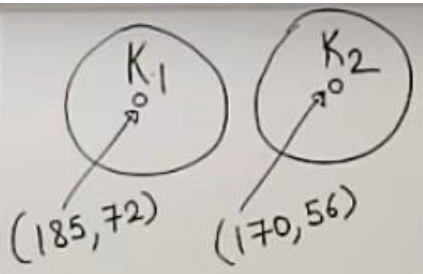


# K-means Algorithm

| | Height | weight |
|---|---|---|
| ① | 185 | 72 |
| ② | 170 | 56 |
| ③ | 168 | 60 |
| ④ | 179 | 68 |
| ⑤ | 182 | 72 |
| ⑥ | 188 | 77 |
| ⑦ | 180 | 71 |
| ⑧ | 180 | 70 |
| ⑨ | 183 | 84 |
| ⑩ | 180 | 88 |
| ⑪ | 180 | 67 |
| ⑫ | 177 | 76 |

## Euclidean Distance

$$\sqrt{(X_0 - X_c)^2 + (Y_0 - Y_c)^2}$$




ED for ③ → $K_1$ → $\sqrt{(168-185)^2+(60-72)^2}$

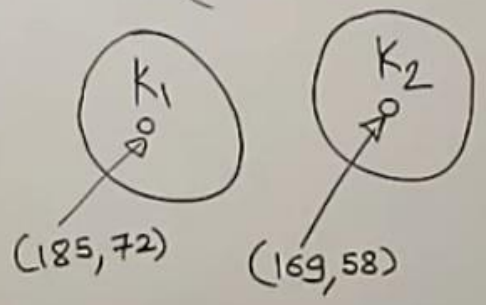
20.80

→ $K_2$ → $\sqrt{(168-170)^2+(60-56)^2}$

= 4.48

## New Centroid Calculation :-

for $K_2 = \left(\frac{170+168}{2}, \frac{60+56}{2}\right) = (169, 58)$




E.D for ④ → $K_1 = \sqrt{(179-185)^2+(68-72)^2}$

= (6.32)

→ $K_2 = \sqrt{(179-169)^2+(68-58)^2}$

= 14.14

$K_1 \rightarrow \{1,4,5,6,7,8,9,10,11,12\}$

$K_2 \rightarrow \{2,3\}$

# K-Means++ Clustering:

In the case of finding initial centroids for K-Means clustering, we were using randomization. The initial k-centroids were picked randomly from the data points.

This randomization of picking k-centroids points results in the problem of initialization sensitivity.

This problem tends to affect the final formed clusters.

The final formed clusters depend on how initial centroids were picked.

There are two approaches to avoid this problem of initialization sensitivity:

**1.Repeat K-means:** Repeating the algorithm and initialization of centroids several times and pick the clustering approach that has small intracluster distance and large intercluster distance.

**2.K-Means++:** K-Means++ is a smart centroid initialization technique.

# Steps of k-mean++

**Step 1** :Randomly select the first centroid from the data points.

**Step 2**: Compute distance of all points in the dataset from the selected centroid.

**Step 3:** For the subsequent centroids, it uses a weighted probability distribution to choose the next centroid.

The probability of selecting a point as the next centroid is higher if it is far away from the already chosen centroids.

# Steps of k-mean++

This ensures that the initial centroids are well spread out across the dataset, leading to better convergence and potentially avoiding suboptimal solutions.

**Step 4:** The remaining steps of the K-means algorithm, including point assignment and centroid update, are the same as in the regular K-means algorithm.

# What Is the Problem of the K-means/K-means++ Method?

A problem with the K-means and K-means++ clustering is that the final centroids are not interpretable or in other words, centroids are not the actual point but the mean of points present in that cluster.

The k-means and k-means++ algorithm is sensitive to outliers !

Since an object with an extremely large value may substantially distort the distribution of the data

## K-Medoids:

Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.

**Python library:-** **sklearn_extra.cluster.KMedoids**

**Medoids** are representative objects of a data set or a cluster within a data set whose sum of dissimilarities to all the objects in the cluster is minimal.

Medoids are similar in concept to means or centroids, but medoids are always restricted to be members of the data set.

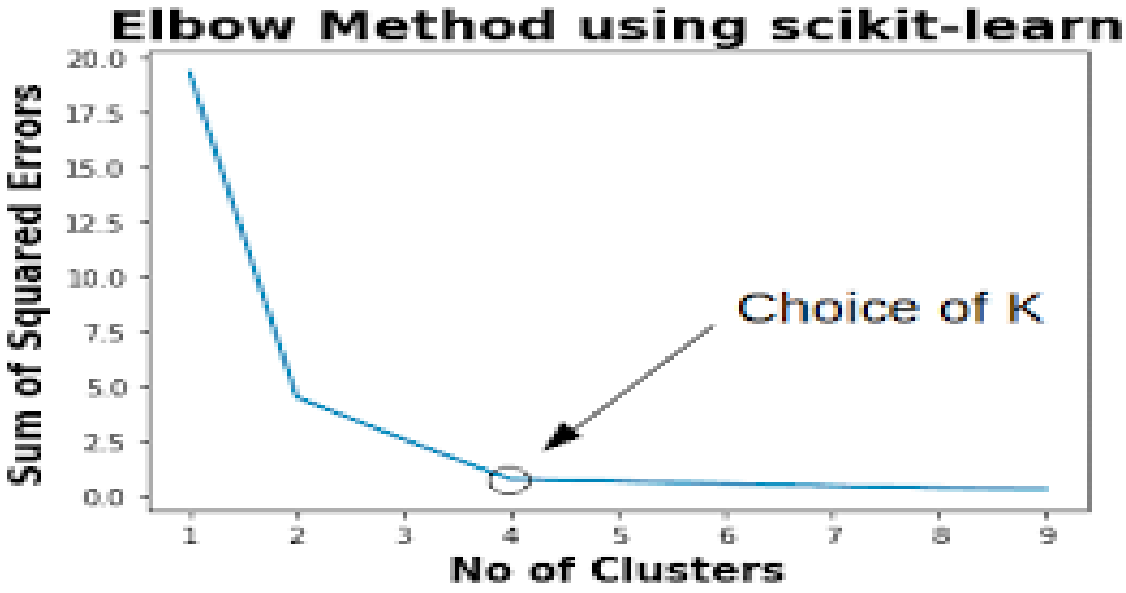
## How to pick the best value of K in case of K-means/K-means++/K-medoids- using Elbow method

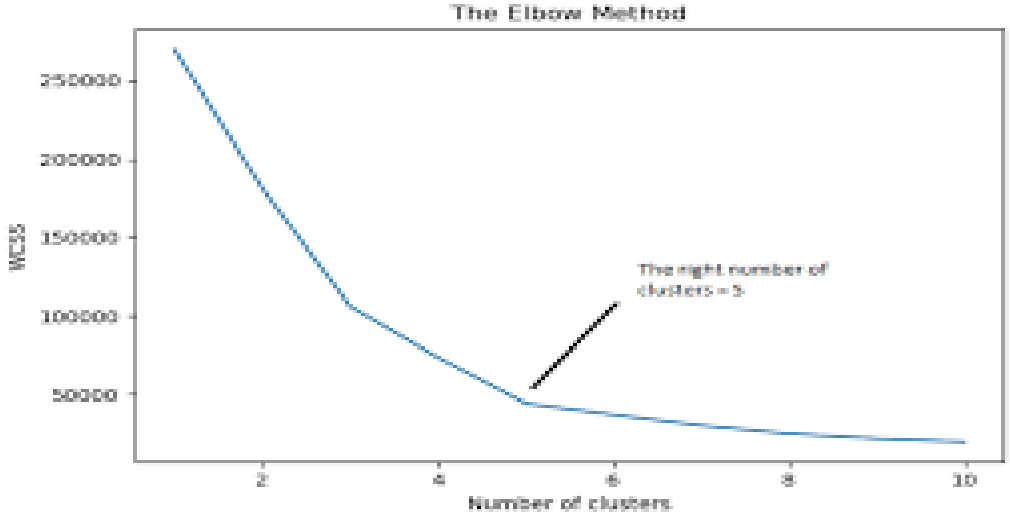
The best value of K can be computed using the *Elbow method.* The cost function of K-Means, K-Means++, and K-Medoids techniques is to minimize intracluster distance and maximize intercluster distance.

The Cost function is known as WCSS (within cluster sum of square distance)

$$\text{WCSS}(k) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \text{cluster } j} \|\mathbf{x}_i - \bar{\mathbf{x}}_j\|^2,$$

where $\bar{\mathbf{x}}_j$ is the sample mean in cluster $j$


Elbow Method using scikit-learn
Sum of Squared Errors
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Choice of K
1
2
3
4
5
6
7
8
9
No of Clusters


The Elbow Method
WCSS
250000
200000
150000
100000
50000
The right number of clusters = 5
2
4
6
8
10
Number of clusters

# Hierarchical Clustering Algorithms :

- It is a statistical method used to build a cluster by arranging elements at various levels.

- The clusters formed in this method form a tree-type structure based on the hierarchy.

- New clusters are formed using the previously formed one. It is divided into two category

- **Agglomerative** (bottom-up *approach*)
- **Divisive** (top-down *approach*)

## Agglomerative Clustering

We start from each data point and keep on making cluster until all the data points are included.

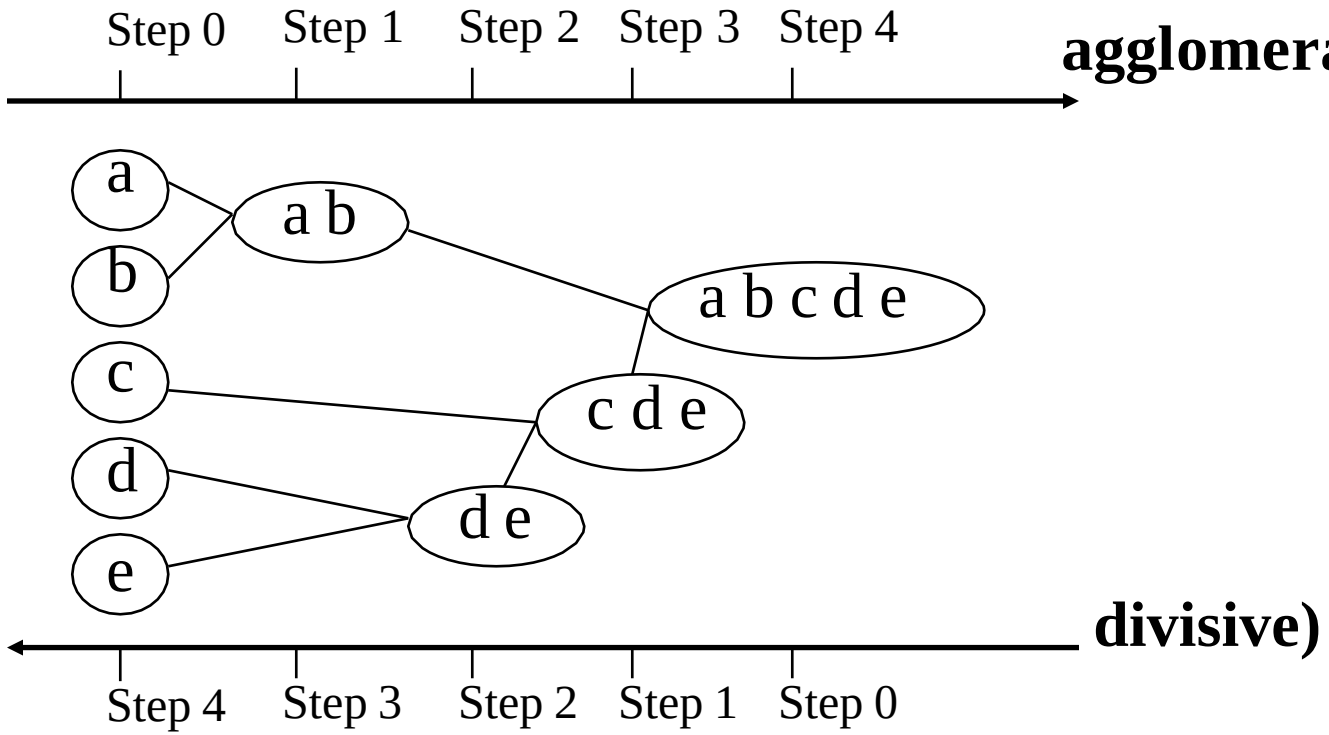
Finally, we find one single big cluster comprises of all data points that we considered initially.

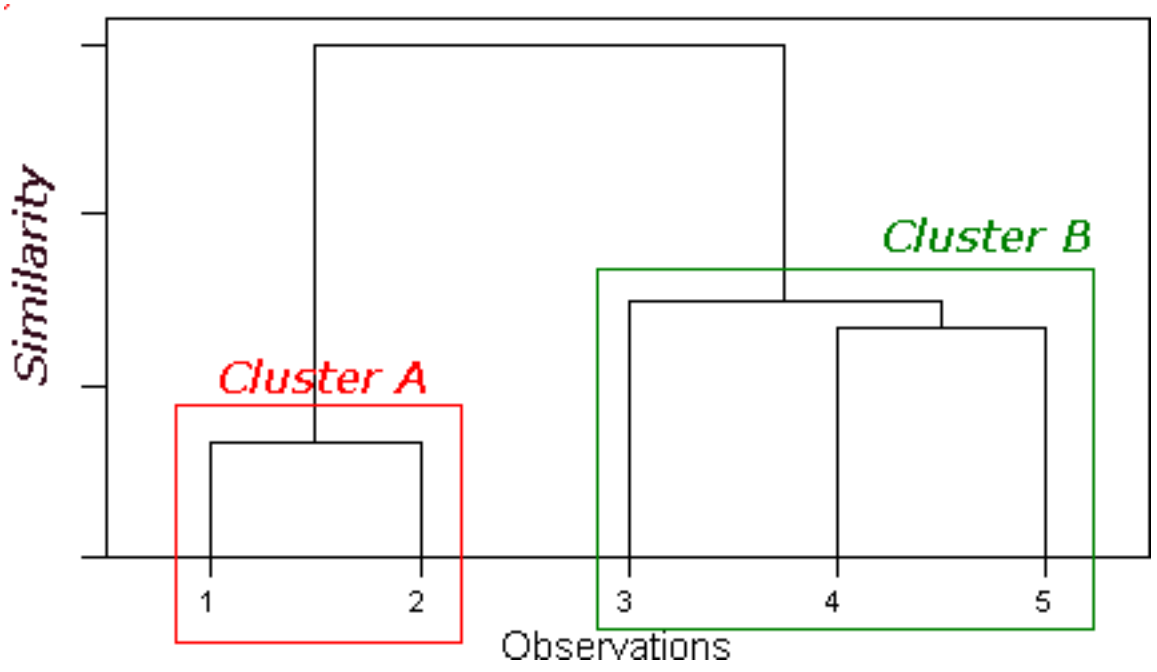
# Dendogram:

- Each level will then represent a possible cluster.
- The height of the dendrogram shows the level of similarity that any two clusters are joined
- The closer to the bottom they are the more similar the clusters are

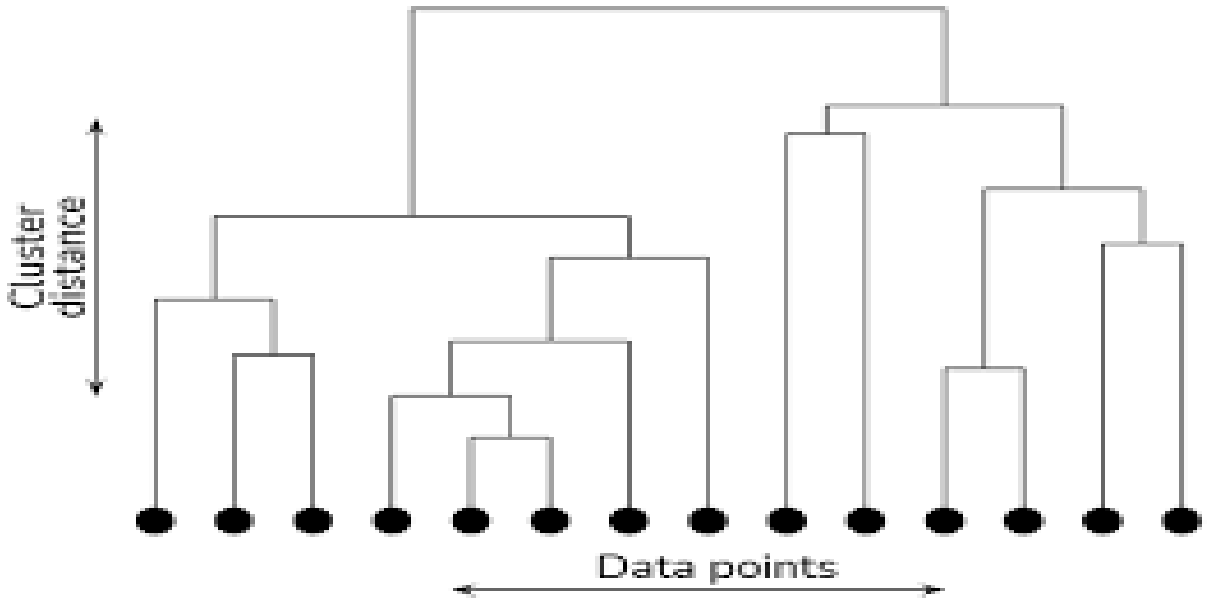
# Hierarchical Clustering

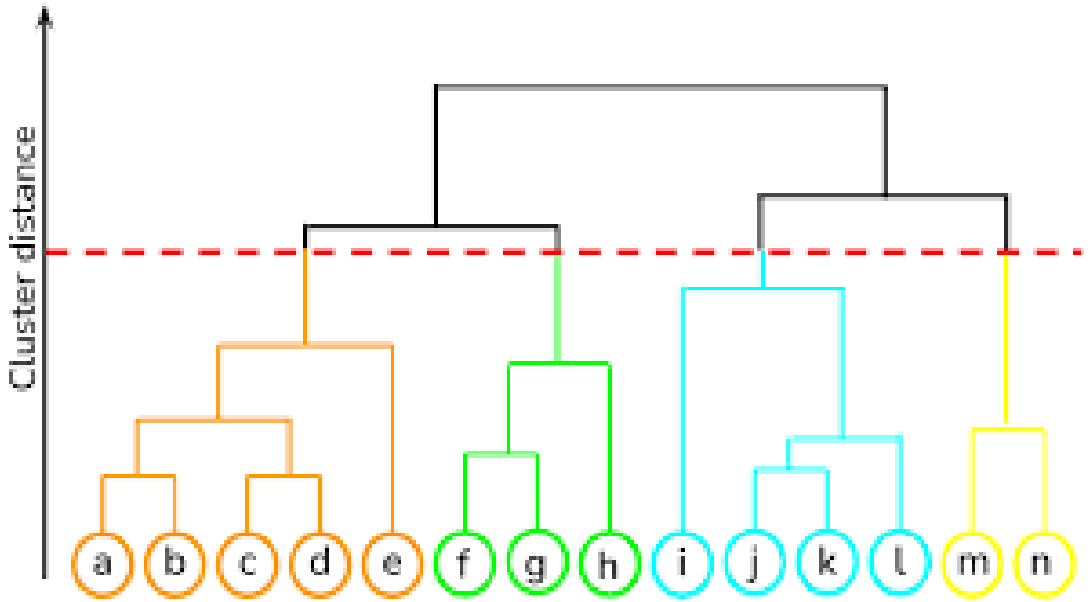
Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition




Similarity
Cluster A
Cluster B
1
2
3
4
5
Observations

# *Dendrogram:* Shows How Clusters are Merged






**There will be a total of 4 clusters**

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|---|
| $P_1$ | 0 | | | | |
| $P_2$ | 9 | 0 | | | |
| $P_3$ | 3 | 7 | 0 | | |
| $P_4$ | 6 | 5 | 9 | 0 | |
| $P_5$ | 11 | 10 | (2) | 8 | 0 |

$\Rightarrow$

| | $P_1$ | $P_2$ | $[P_3, P_5]$ | $P_4$ |
|---|---|---|---|---|
| $P_1$ | 0 | | | |
| $P_2$ | 9 | 0 | | |
| $[P_3, P_5]$ | 3 | 7 | 0 | |
| $P_4$ | 6 | 5 | 8 | 0 |

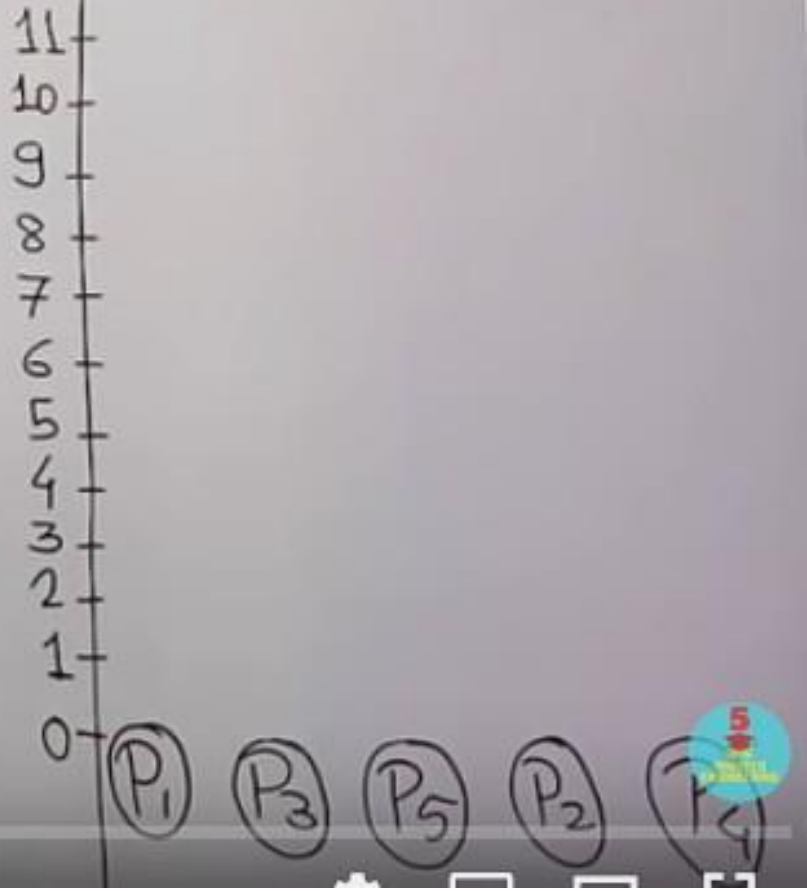
$\Rightarrow d(P_1, [P_3, P_5])$
$\Rightarrow \min(d(P_1, P_3), d(P_1, P_5))$
$\Rightarrow \min(3, 11) \Rightarrow 3$

---

$\Rightarrow d(P_2, [P_3, P_5])$
$\Rightarrow \min(d(P_2, P_3), d(P_2, P_5))$
$\Rightarrow \min(7, 10) \Rightarrow 7$

---

$\Rightarrow d(P_4, [P_3, P_5])$
$\Rightarrow \min(d(P_4, P_3), d(P_4, P_5))$
$\Rightarrow \min(9, 8) \Rightarrow 8$



# Affinity Propagation

Affinity Propagation is a clustering algorithm that does not require specifying the number of clusters in advance. It discovers the number of clusters automatically based on the data's similarities.

It operates by iteratively exchanging messages between data points to determine the most representative exemplars, which then define the clusters.

Affinity Propagation is effective in finding clusters with varying sizes and shapes.

However, it can be computationally expensive, especially for large datasets, as it requires pairwise similarity calculations and matrix operations

1. Similarity Matrix:

•Affinity Propagation begins by constructing a similarity matrix that measures the pairwise similarities between data points.

•The similarity metric can be based on various distance measures such as Euclidean distance or similarity measures like cosine similarity.

## 2. Availability and Responsibility Matrices:

The algorithm initializes two matrices: the availability matrix (A) and the responsibility matrix (R), both of size N x N, where N is the number of data points.

The availability matrix reflects the accumulated evidence of how well-suited a data point is to be an exemplar, while the responsibility matrix quantifies the suitability of one data point being chosen as an exemplar relative to others.

## 3. Message Passing:

- The algorithm iteratively updates the availability and responsibility matrices until convergence.

- In each iteration, the algorithm passes messages between data points to update their availability and responsibility values.

- The responsibility update equation is based on the current similarities and the maximum responsibility of other data points for each exemplar.

- The availability update equation considers the current responsibilities and the accumulated availabilities of other data points for each exemplar.

4. Exemplar Selection:

•After convergence, the algorithm determines the exemplars, which are the most representative data points for each cluster.

•Exemplars are selected based on high availability values, indicating their ability to represent other data points well.

## 5. Convergence:

- The algorithm continues updating the matrices and exchanging messages until convergence criteria are met.

- Convergence is typically determined by a maximum number of iterations or when the change in the matrices' values falls below a threshold.

Python code

#Create an instance of the AffinityPropagation class and fit it to your data:

```
from sklearn.cluster import AffinityPropagation
clustering = AffinityPropagation().fit(X)
```

#Obtain the cluster centers and labels from the trained model:

```
cluster_centers = clustering.cluster_centers_
labels = clustering.labels_
num_clusters = len(cluster_centers)
num_clusters
cluster_centers
```

```
#Plot the data points and cluster centers using different colors for each cluster
plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis’)

plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1], marker='x', color='red',
s=100)

plt.title('Affinity Propagation Clustering')
plt.show()
```

The Colormap(cmap) instance used to map scalar data to colors.