Introduction To Unsupervised Learning

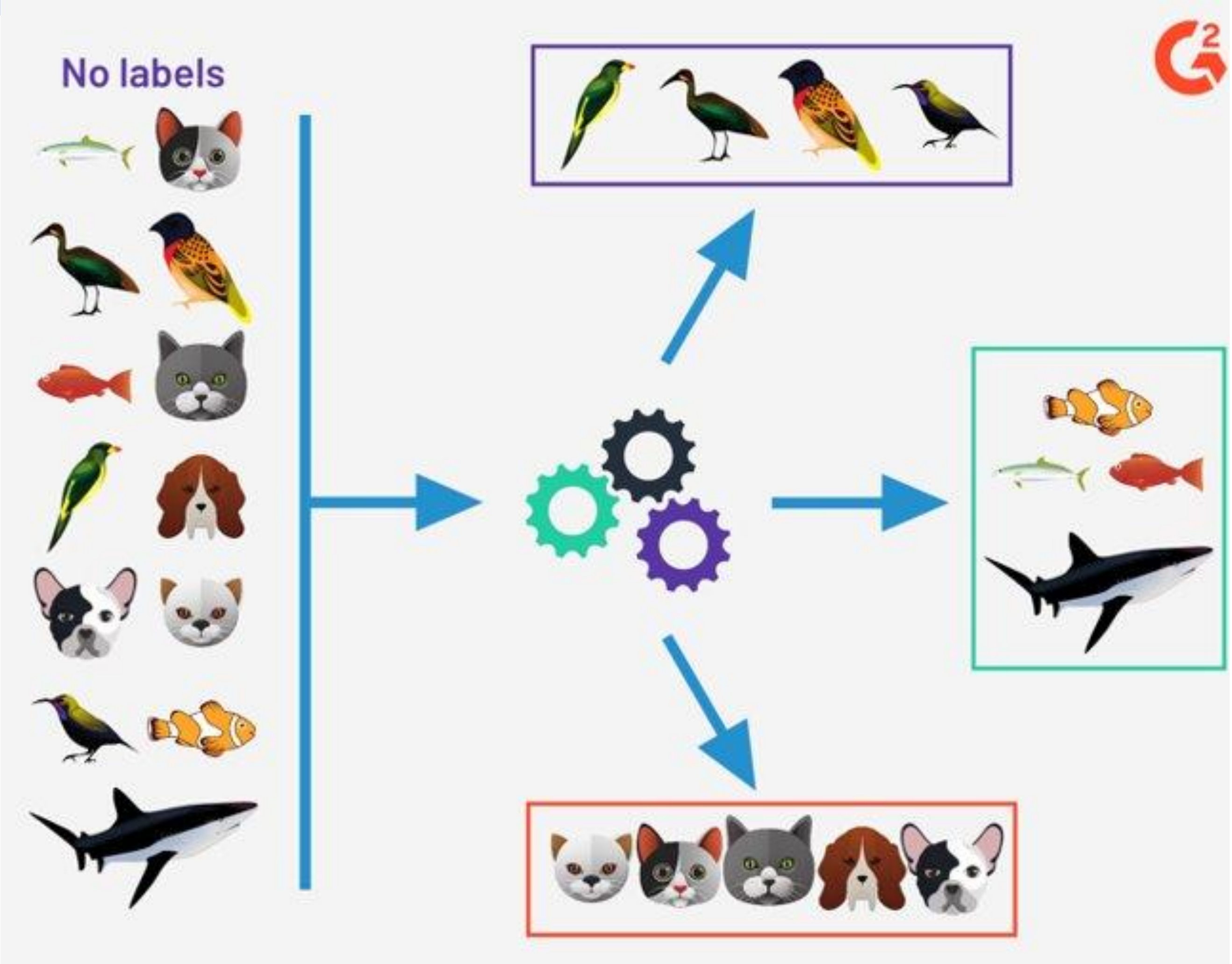
DATA FOLKZ®

#CATAPULT DATA LEADERS

Unsupervised Learning

Unsupervised Learning is a machine learning technique in which there is no target variable (y) to supervise the model.
Instead, it allows the model to work on its own to discover patterns and information that was previously undetected in data (X). It mainly deals with the unlabelled data.

Let's understand Unsupervised Learning with an example



1. Unsupervised learning works by detecting the similarities and then grouping the similar points together and form clusters

2. Whenever a new data point comes its features are matched with the clusters made and it allocated to the cluster with the most similarity .

Here , in this diagram three clusters are made based on the features , actions and functionality .

DATA FOLKZ®
#CATAPULT DATA LEADERS

Clustering

What is Clustering?

- Clustering can be considered the most important *unsupervised learning* problem.

- Clustering is the task of dividing the population or data points into several groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. It does it by finding some similar patterns in the unlabeled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

In simple words, the aim is to segregate groups with similar traits and assign them into clusters

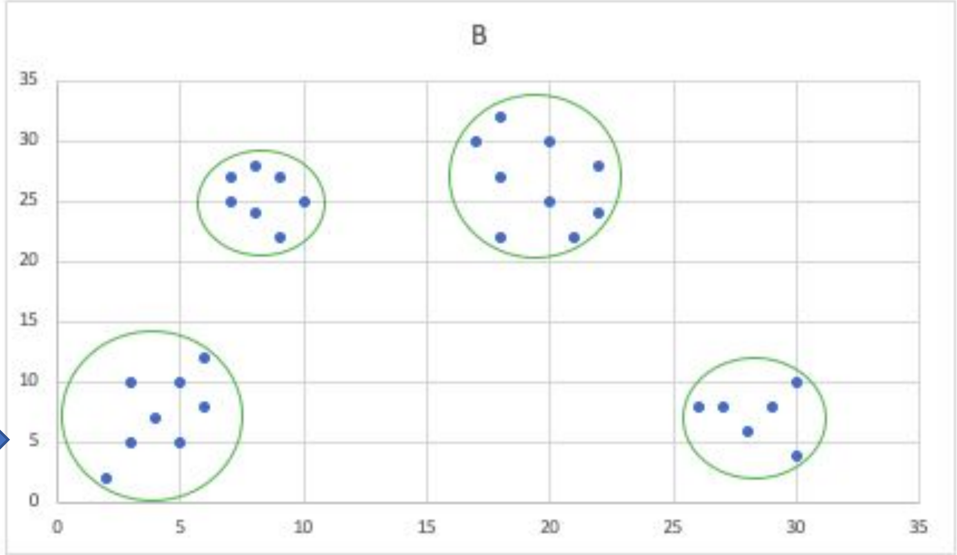
Requirements of Clustering

The main requirements that a clustering algorithm should satisfy are:

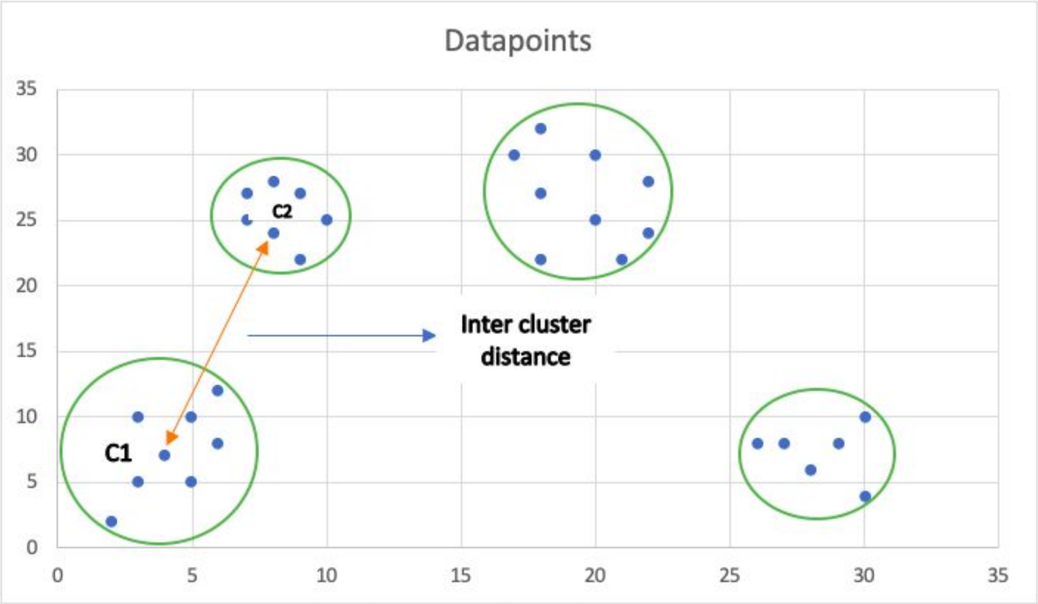
- **Scalability**
- **Ability to deal with different kinds of attributes**
- **Discovery of clusters with attribute shape**
- **High dimensionality**
- **Ability to deal with noisy data**
- **Interpretability**

Clustering Example

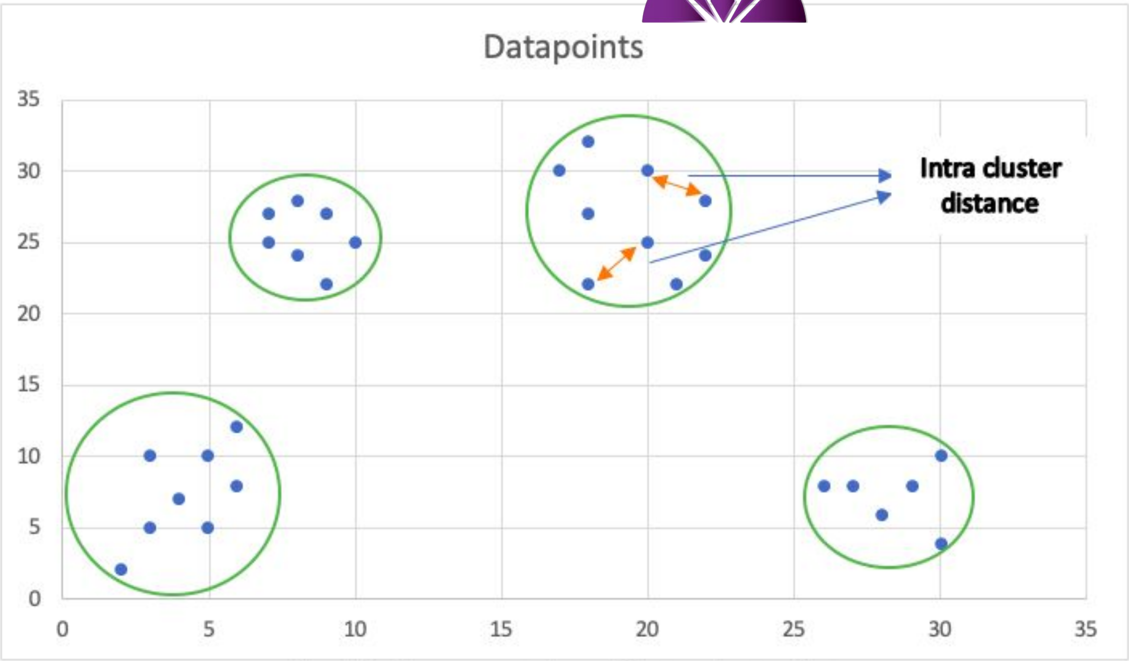
A	B
7	27
18	32
26	8
6	8
4	7
9	22
20	25
22	28
30	4
28	6
20	30
10	25
9	27
7	25
8	28
8	24
21	22
30	10
27	8
5	10
6	12
2	2
5	5
3	10
3	5
17	30
18	27
22	24
29	8
18	22



In this example , all the data points are plotted and the points which are the most nearest or the data points whose distance the closest are grouped together and forms a cluster .



Graphical representation of Intra cluster distance



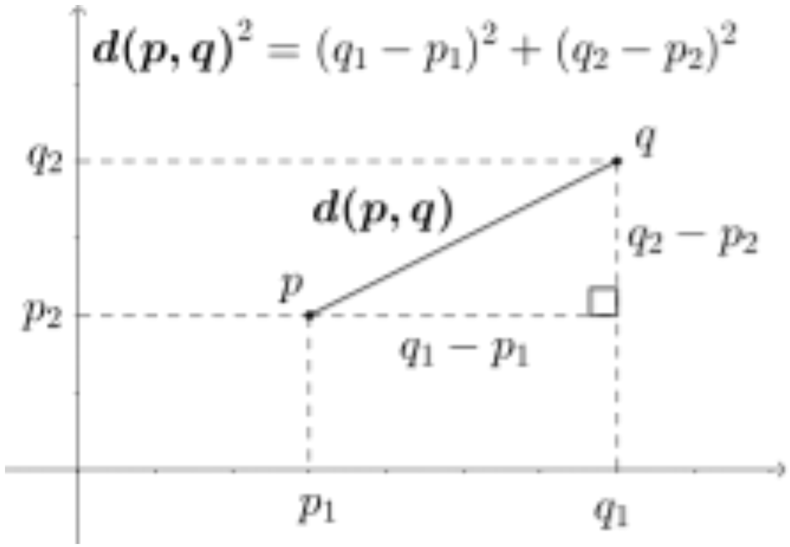
Graphical representation of Intra cluster distance

Inter cluster is the distance between two objects/data points belonging to two different clusters .

Intra cluster is the distance between two objects/data points belonging to same cluster

Euclidean distance

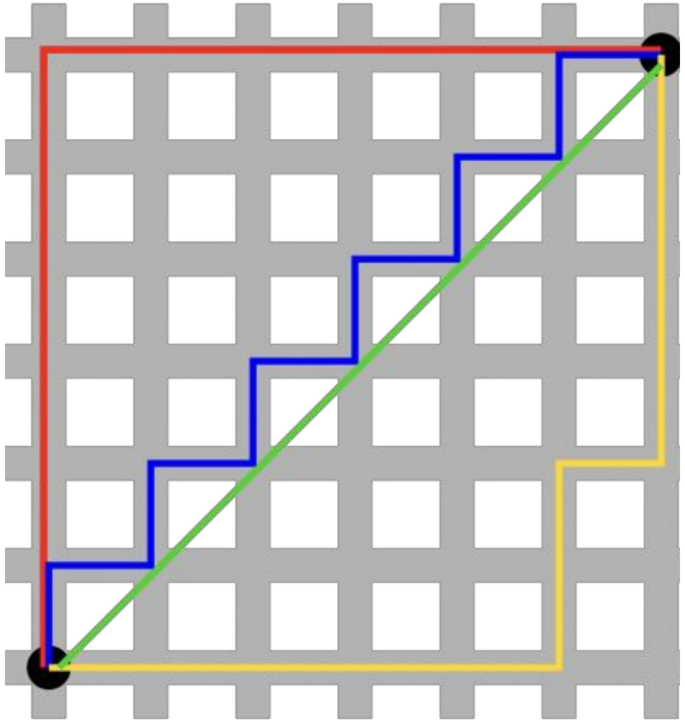
- Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space



$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Manhattan distance



- The distance between two points measured along axes at right angles. In a plane with p1 at (x1, y1) and p2 at (x2, y2), it is |x1 - x2| + |y1 - y2|

- This is known as Manhattan distance because all paths from the bottom left to top rights of this idealized city have the same distance:

Minkowski distance

- The Minkowski distance or Minkowski metric is a metric which can be considered as a generalization of both the Euclidean distance and the Manhattan distance.

$$X = (x_1, x_2, \ldots, x_n) \text{ and } Y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$$

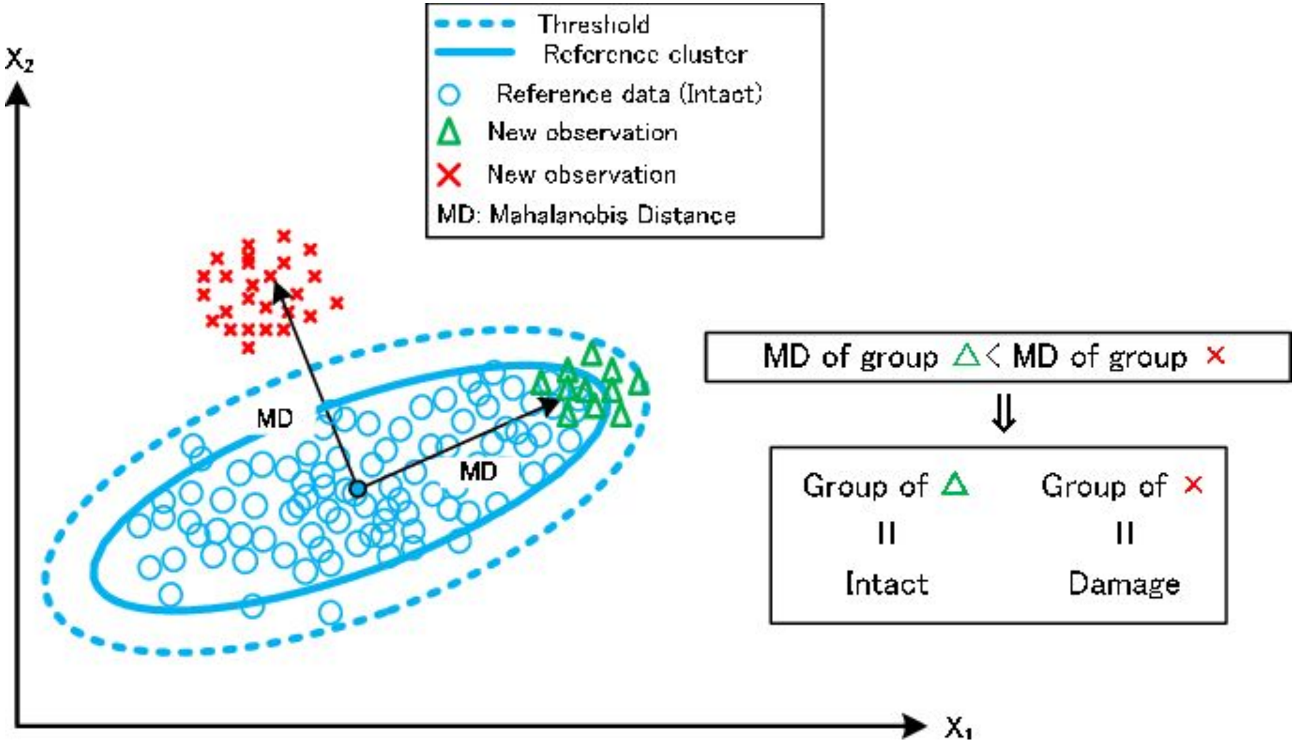
The Minkowski distance of order p (where p is an integer) between two points is defined as

$$D(X,Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Mahalanobis Distance

- The Mahalanobis distance of an observation $\vec{x} = (x_1, x_2, x_{3......}\, x_n)^T$ from a set of observations with $\vec{\mu} = (\mu_1, \mu_2, \mu_{3......}\, \mu_n)^T$ mean and covariance matrix *S* is defined as:

$$D_M(\vec{x}) = \sqrt{(\vec{x}-\vec{\mu})^T S^{-1} (\vec{x}-\vec{\mu})}.$$



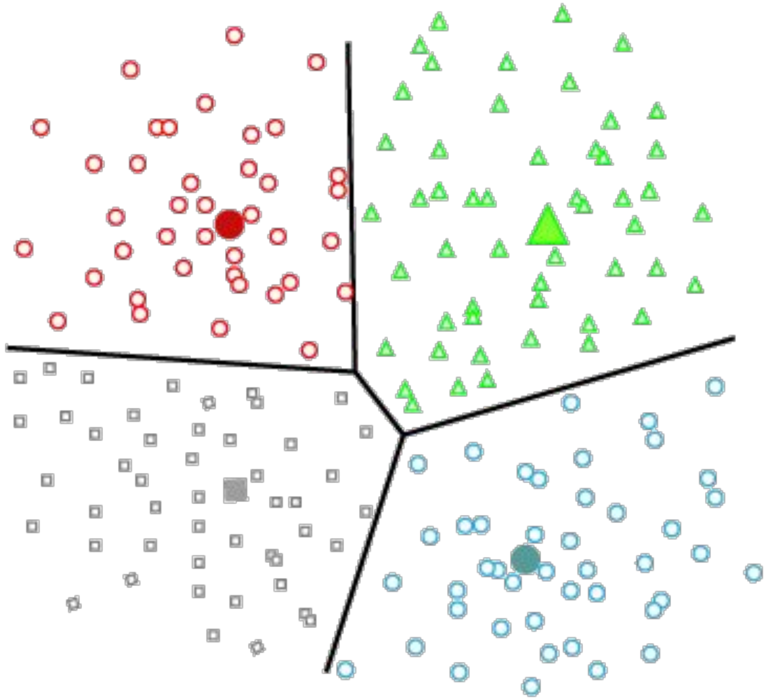
Types of Clustering Algorithms

- K-Means Clustering
- K-Means ++ Clustering
- Hierarchical Clustering
- DBSCAN (Density Based Clustering)

Most Important of all these is the K-Means Clustering.

K-Means Clustering

- K-Means attempts to classify data without having first been trained with labelled data.
- In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups.
- The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.
- Once the algorithm has been run and the groups are defined, any new data can be easily assigned to the most relevant group.



Steps for K-Means Algorithm

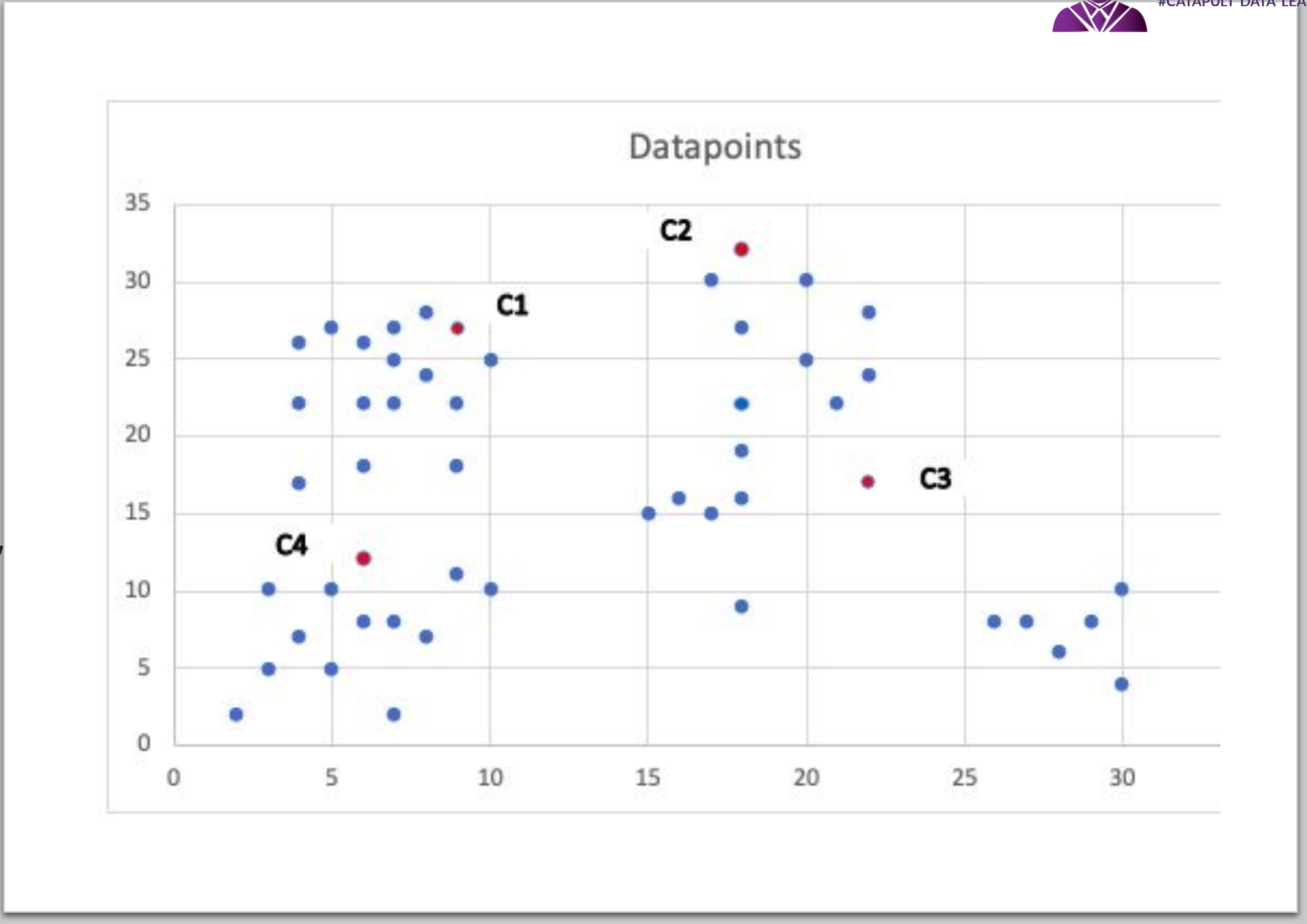
K-Means Clustering

Step 1 : Choose the number of clusters , k
Step 2 : Randomly select the centroid of each cluster
Step 3 : Find the **Euclidean distance** between each data instance and centroids of all the clusters.
Step 4 : Assign the data instances to the cluster of the centroid with **nearest distance**.
Step 5 : Calculate **new centroid values** based on the mean values of the coordinates of all the data instances from the corresponding cluster.
Step 6 : Repeat the 3 and 4 step till the model satisfies one of the below given condition :

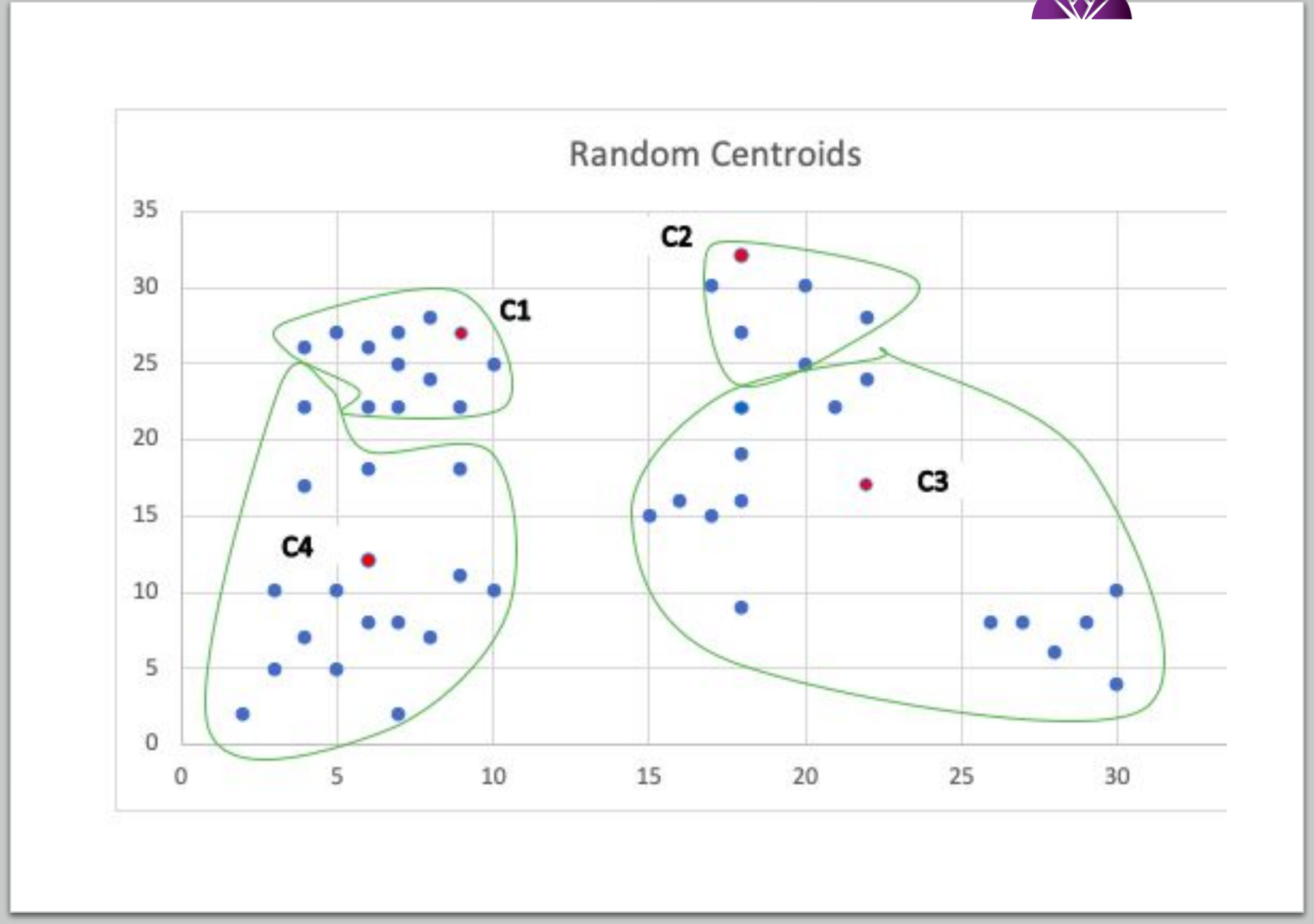
The model halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

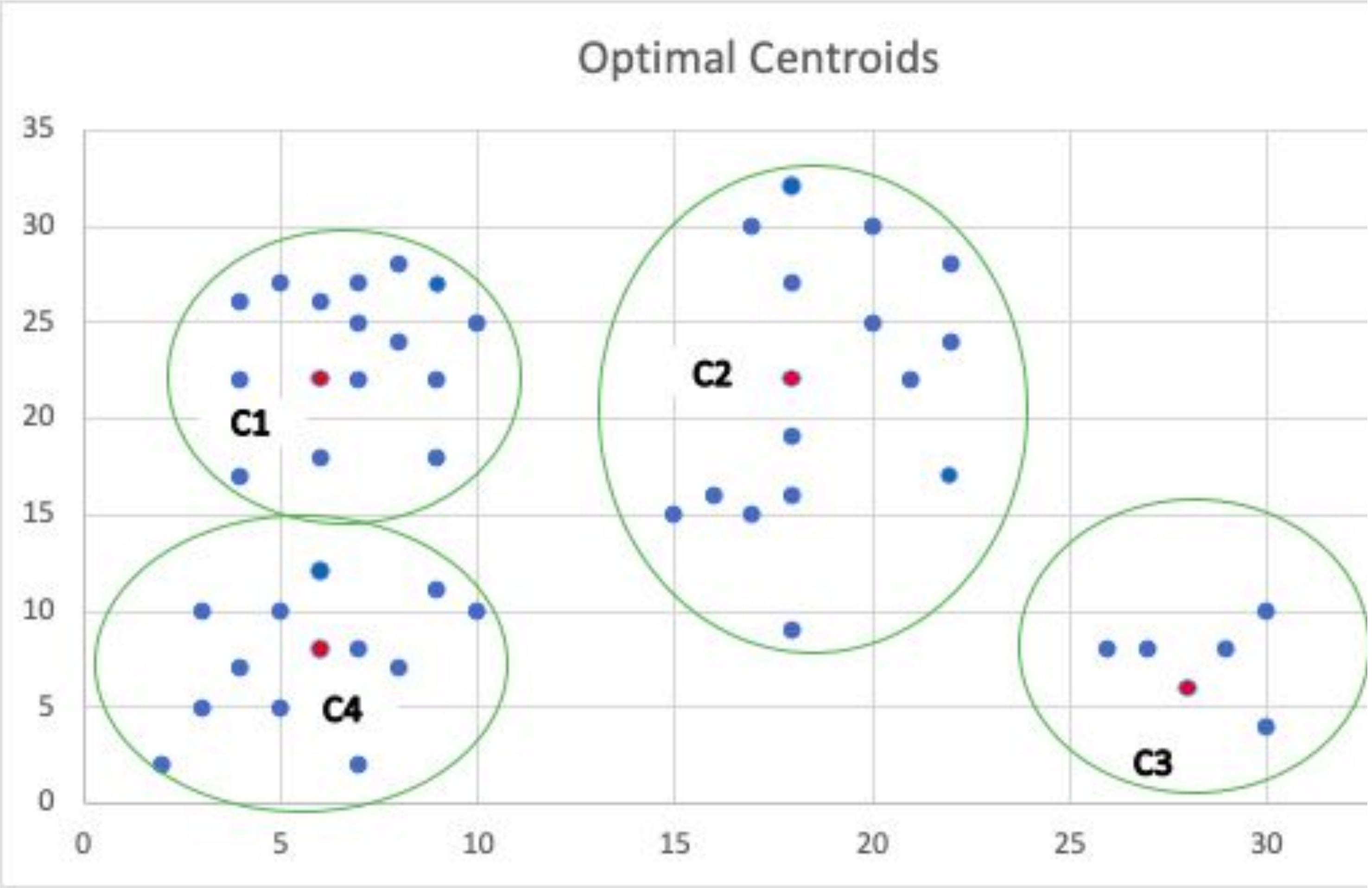
1.Select **K** random points as cluster center's called centroids. Let’s say we choose K = 4.



2. Assign each object to the group that has the closest centroid using the Distance formula .



3. When all objects have been assigned, recalculate the positions of the K centroids. Repeat Steps 2 and 3 until the centroids no longer move.



Selecting the optimal K

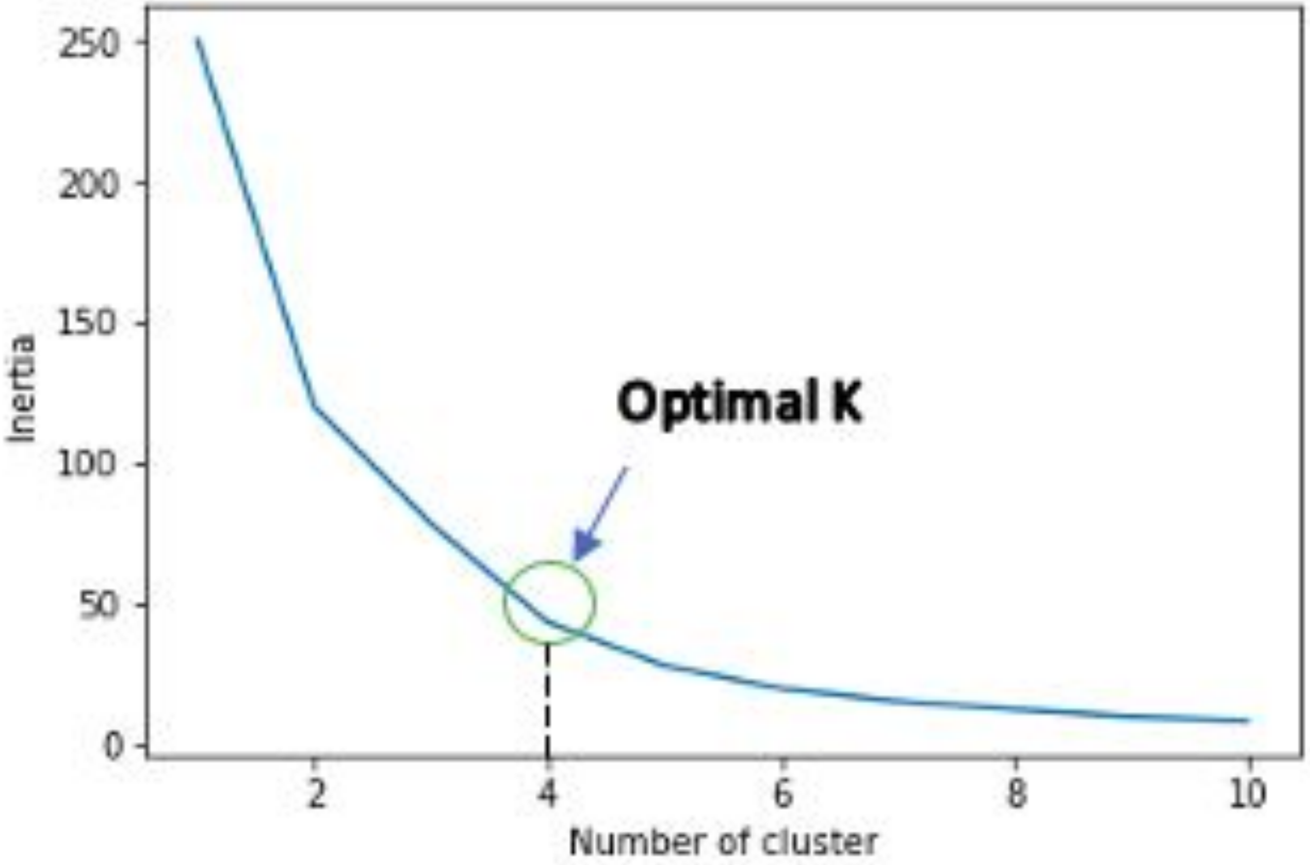
- The K-means algorithm aims to choose centroids that minimize the **inertia i.e the sum of distances of all the points within a cluster from the centroid of that cluster.**
- This means the sum of intra cluster distance should be minimum.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

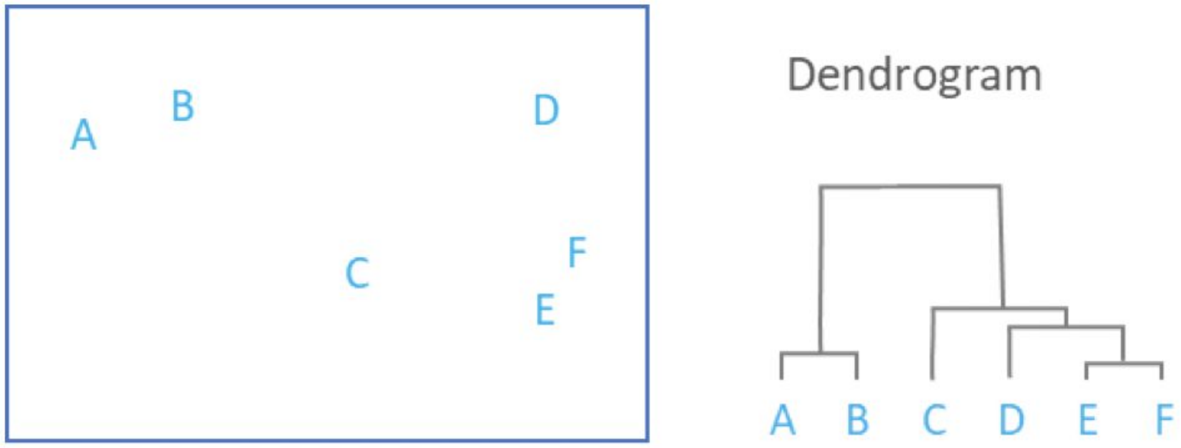
The Elbow Method

The **Elbow Method** is one of the most popular methods to determine this optimal value of k.

The number of cluster at which the inertia is minimum is choosen .



Hierarchical Clustering



Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters.

The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

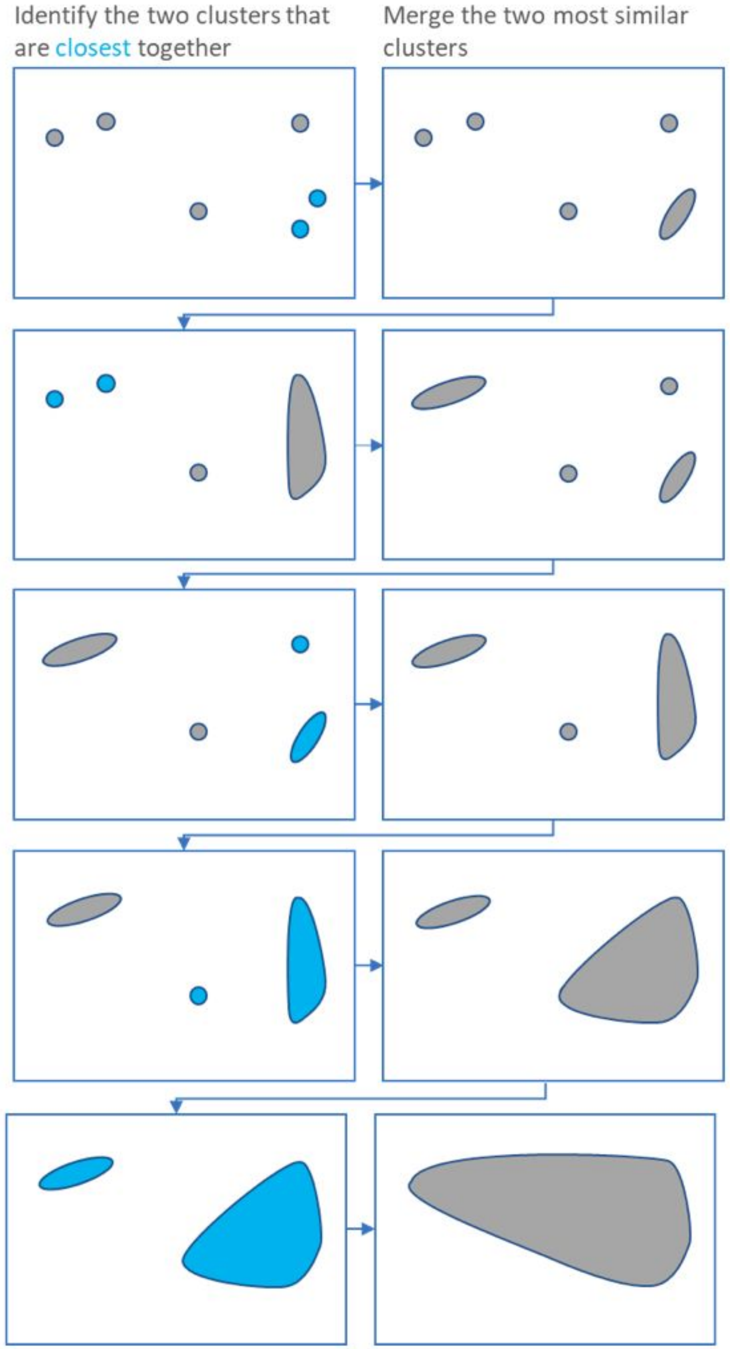
e.g: All files and folders on our hard disk are organized in a hierarchy.

This clustering technique is divided into two types:

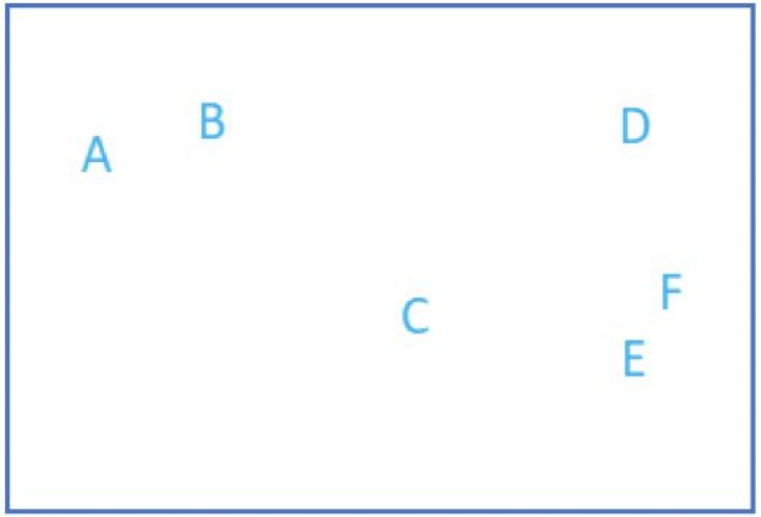
- Agglomerative
- Divisive

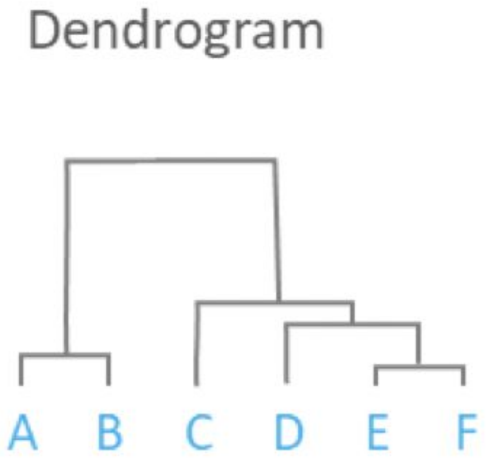
How Agglomerative Hierarchical Clustering work ?

In this technique, initially each data point is considered as an individual cluster. At each iteration, the similar clusters merge with other clusters until one cluster or K clusters are formed.


Identify the two clusters that are closest together
Merge the two most similar clusters

The main output of Hierarchical Clustering is a *dendrogram,* which shows the hierarchical relationship between the clusters



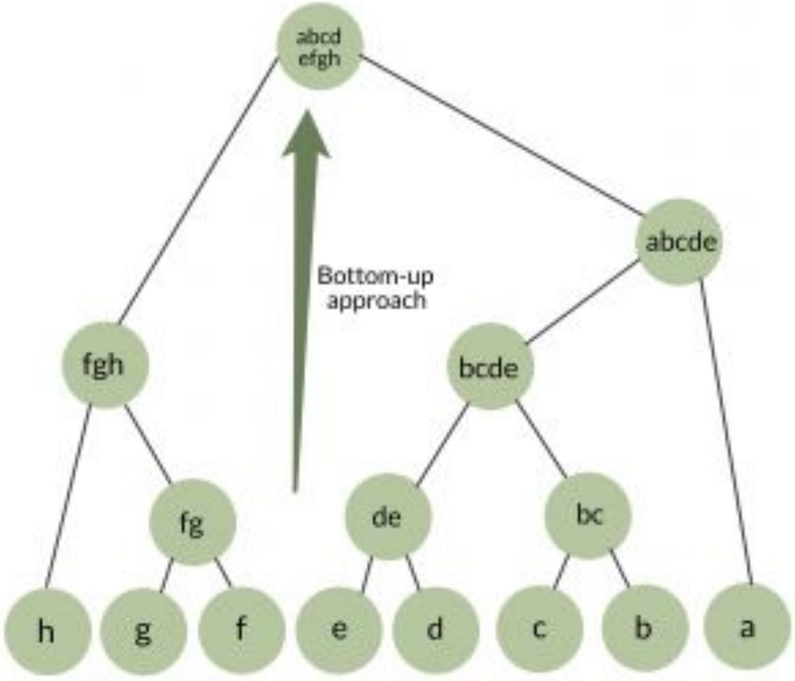


Reference : DisplayR blog

Agglomerative Hierarchical Clustering

It is also known as bottom-up approach . This clustering algorithm does not require us to prespecify the number of clusters.

Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.



Hierarchical agglomerative clustering

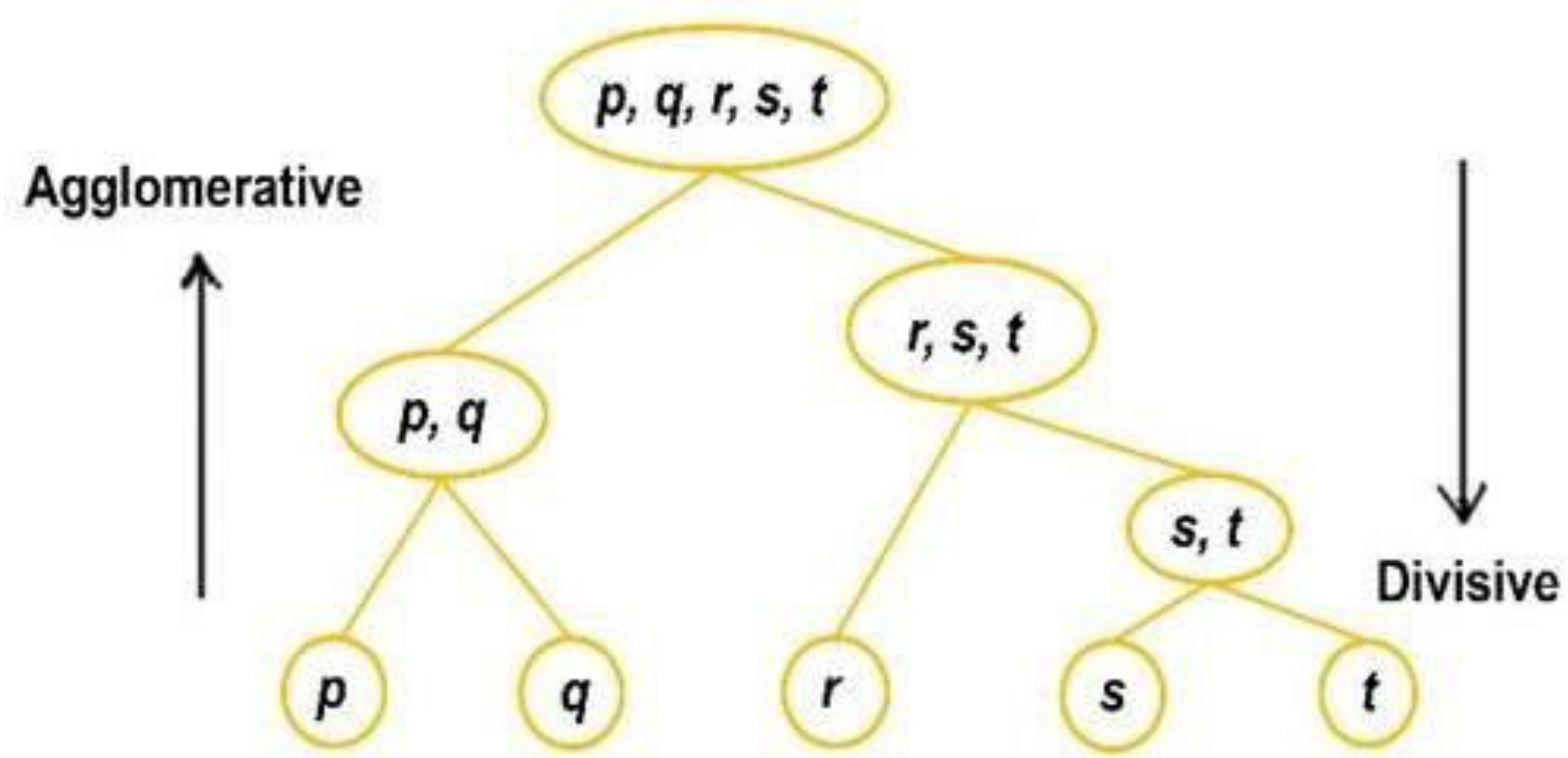
Divisive clustering

- **Divisive Hierarchical clustering** is exactly the opposite of the Agglomerative Hierarchical clustering and is not much used in the real world.

- In Divisive Hierarchical clustering, we consider all the data points as a single cluster and in each iteration, we separate the data points from the cluster which are not similar.

- Each data point which is separated is considered as an individual cluster. In the end, we'll be left with n clusters.

Divisive clustering



How to measure the similarity between the clusters ?

Calculating the similarity between two clusters is important to merge or divide the clusters. There are certain approaches which are used to calculate the similarity between two clusters:

- **Complete-linkage:** the distance between two clusters is defined as the longest distance between two points in each cluster.

- **Single-linkage:** the distance between two clusters is defined as the shortest distance between two points in each cluster. This linkage may be used to detect high values in your dataset which may be outliers as they will be merged at the end.

- **Average-linkage:** the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

- **Centroid-linkage:** finds the centroid of cluster 1 and centroid of cluster 2, and then calculates the distance between the two before merging.

The choice of linkage method entirely depends on you and there is no hard and fast method that will always give you good results. Different linkage methods lead to different clusters.

Thankyou