Unit 5
Unsupervised Learning and Reinforcement Learning

By Suchita patil

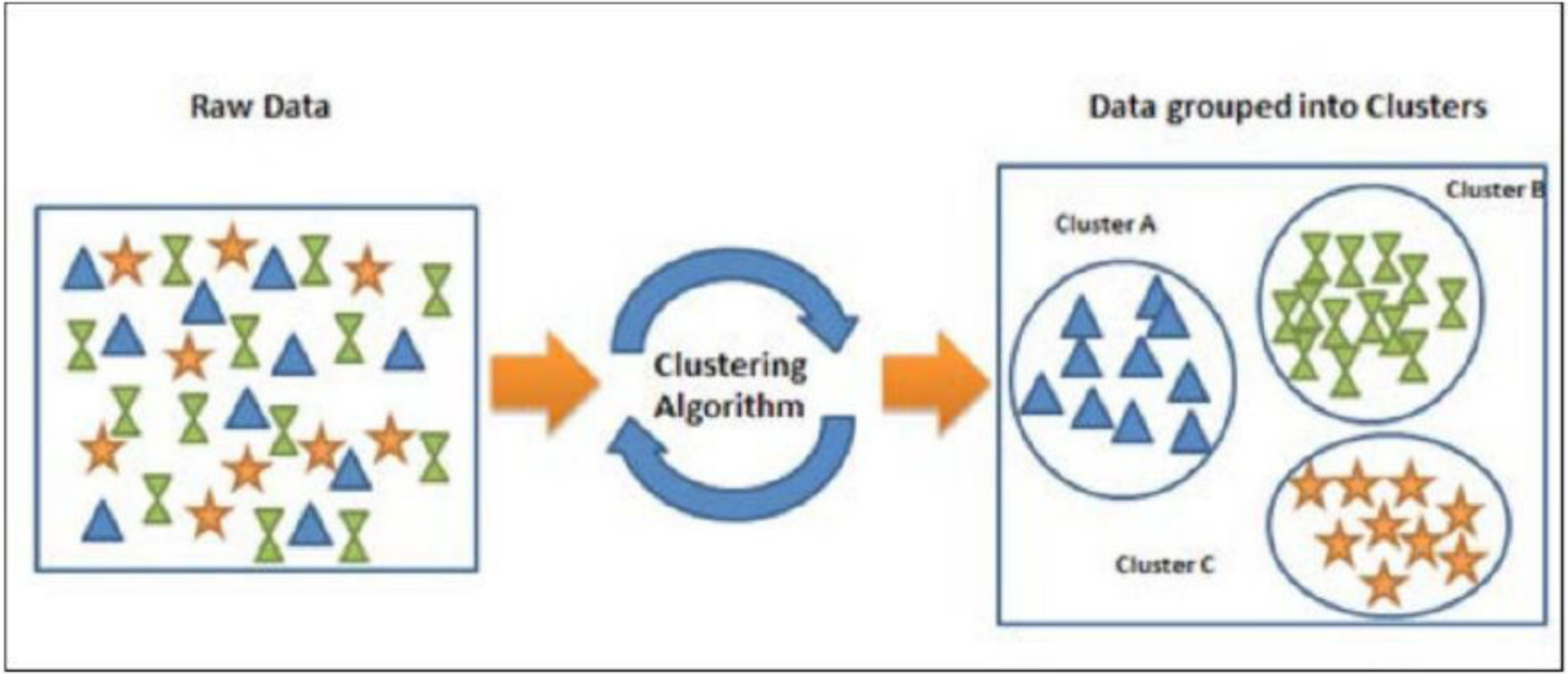
Content:

- Unsupervised learning: Introduction to clustering,
- K Means clustering,
- Hierarchical clustering,
- Association rule mining.
- Introduction to reinforcement learning – Q learning

Clustering

- It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.
- It can be defined as ***"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."***
- It does it by finding some similar patterns in the unlabelled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.
- **Example: Let's understand the clustering technique with the real-world example of Mall: When we** visit any shopping mall, we can observe that the things with similar usage are grouped together. Such as the t-shirts are grouped in one section, and trousers are at other sections, similarly, at vegetable sections, apples, bananas, Mangoes, etc., are grouped in separate sections, so that we can easily find out the things.
- The clustering technique can be widely used in various tasks. It is used by the **Amazon in its** recommendation system to provide the recommendations as per the past search of products. **Netflix also uses this technique to recommend the movies and web-series to its users as per the** watch history.

The following diagram depicts the clustering process:



Types of Clustering

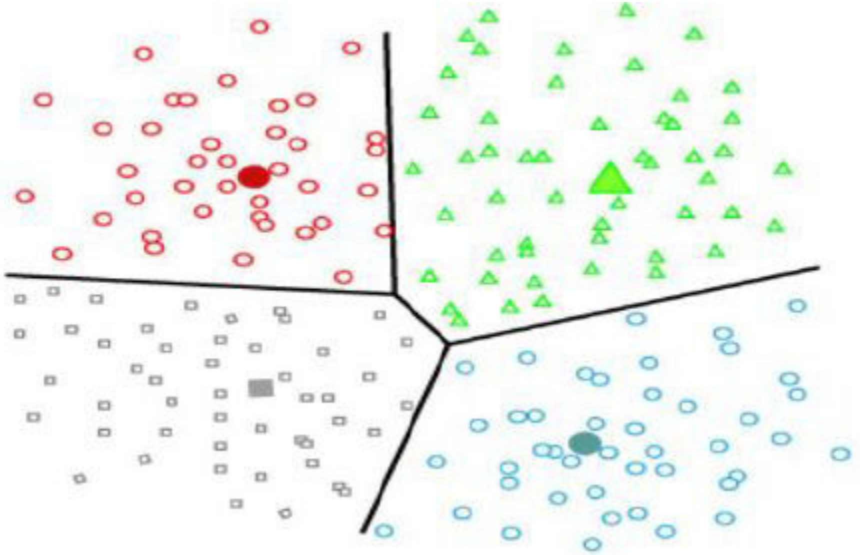
The clustering methods are broadly divided into **Hard clustering (datapoint belongs to only** one group) and **Soft Clustering (data points can belong to another group also).**

1. **Partitioning Clustering**
2. **Density-Based Clustering**
3. **Distribution Model-Based Clustering**
4. **Hierarchical Clustering**
5. **Fuzzy Clustering**

1. Partitioning Clustering

It is a type of clustering that divides the data into non-hierarchical groups. It is also known as the **centroid-based method. The most** common example of partitioning clustering is the **K-Means Clustering algorithm.**

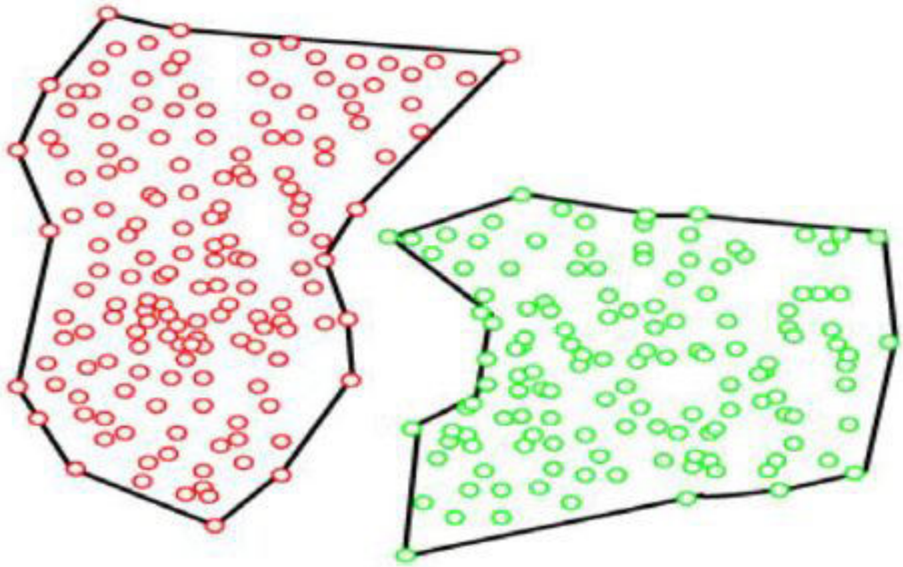
In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



2. Density-Based Clustering

The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected.

These algorithms can face difficulty in clustering the data points if the dataset has varying densities and high dimensions.



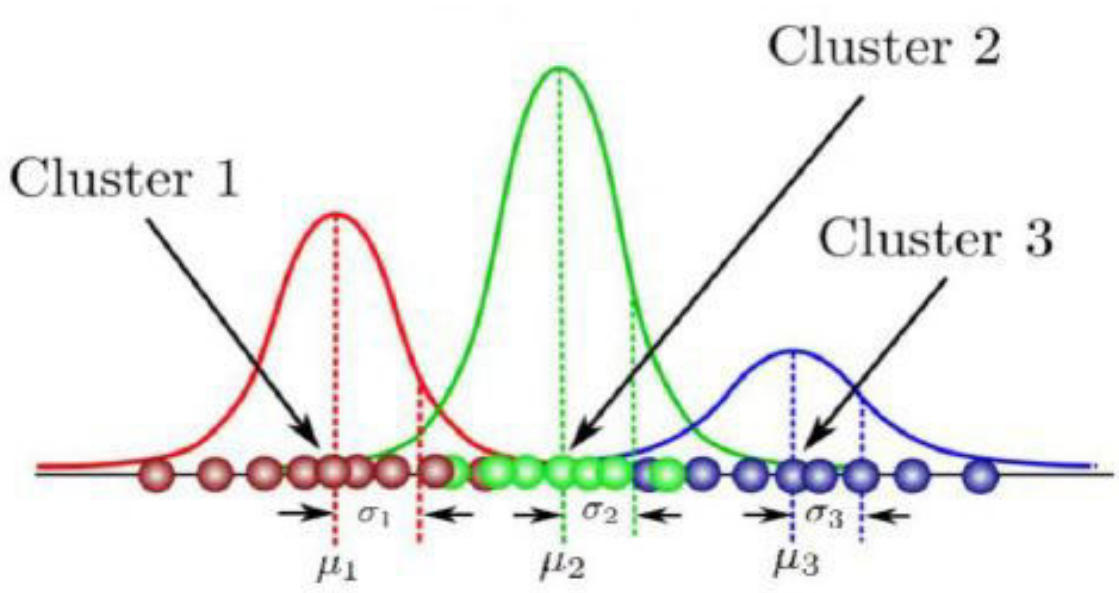
3. Distribution Model-Based Clustering

In the distribution model-based clustering method, the data is divided based on the probability of how a dataset belongs to a particular distribution.

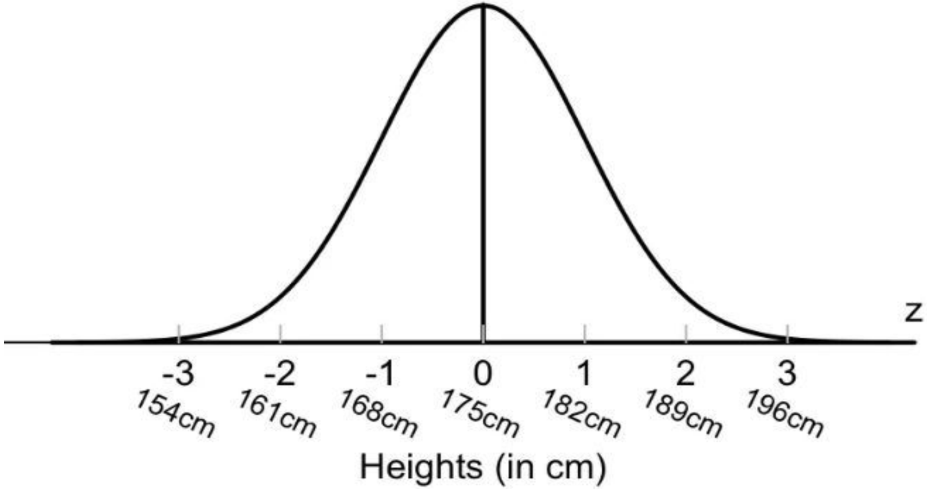
The grouping is done by assuming some distributions commonly **Gaussian Distribution.**

The example of this type is the **Expectation-Maximization Clustering algorithm that uses Gaussian Mixture Models (GMM).**

- The **Gaussian distribution**, also known as the **normal distribution**, is a continuous probability distribution that is widely used in statistical modeling and Machine Learning. It is a bell-shaped curve that is symmetrical around its mean and is characterized by its mean and standard deviation.



- **Example of a Normal Distribution**
- Many naturally-occurring phenomena appear to be normally-distributed. Take, for example, the distribution of the heights of human beings. The average height is found to be roughly 175 cm (5' 9"), counting both males and females.
- As the chart below shows, most people conform to that average. Meanwhile, taller and shorter people exist, but with decreasing frequency in the population. According to the empirical rule, 99.7% of all people will fall with +/- three standard deviations of the mean, or between 154 cm (5' 0") and 196 cm (6' 5"). Those taller and shorter than this would be quite rare (just 0.15% of the population each).



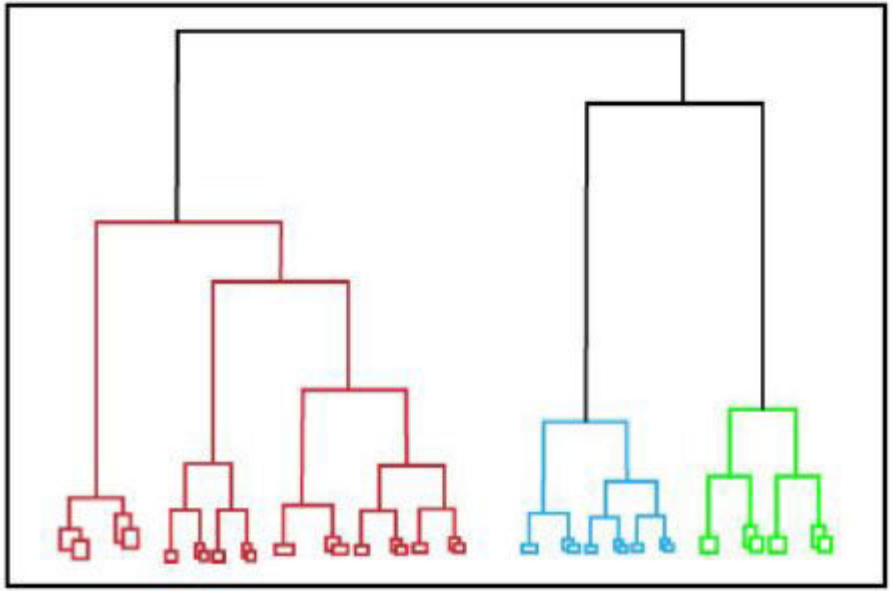
4. Hierarchical Clustering

In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a **dendrogram.**

The observations or any number of clusters can be selected by cutting the tree at the correct level.

There is no requirement of pre-specifying the number of clusters to be created.

The most common example of this method is the **Agglomerative Hierarchical algorithm.**



5. Fuzzy Clustering

Fuzzy clustering is a type of soft method in which a data object may belong to more than one group or cluster.

Each dataset has a set of membership coefficients, which depend on the degree of membership to be in a cluster.

Fuzzy C-means algorithm is the example of this type of clustering; it is sometimes also known as the Fuzzy k-means algorithm.

Clustering Algorithms

There are primarily two classes of clustering algorithm:

- **The Partitional Clustering algorithms:**
  partitional clustering algorithms define clusters that divide the dataset into mutually disjoint partitions.
- **The Hierarchical Clustering algorithms:**
  The Hierarchical clustering algorithms define clusters that have a hierarchy

Partitional clustering

In Partitional clustering every instance can be placed in one and only one of the k clusters.

The number of clusters (k) to be formed is input to this algorithm, and this one set of k clusters is the output of the partitional cluster algorithms.

One of the most commonly used partitional clustering algorithms is the k-means clustering algorithm.

The following are the steps involved in the centroid-based partitional clustering algorithm:

Input: k (the number of clusters) and d (the data set with n objects)

Output: Set of k clusters that minimize the sum of dissimilarities of all the objects
to the identified centroid

1. Identify the k objects as the first set of centroids.
2. Assign the remaining objects that are nearest to the centroid.
3. Randomly select a non-centroid object and recompute the total points that will be swapped to form a new set of centroids, until you need no more swapping.

The k-means clustering algorithm

- The k-means is a partitional clustering algorithm.
- Let the set of data points (or instances) be as follows:

  D = {x1, x2, ..., xn}

- The k-means algorithm partitions the given data into k clusters with each cluster having a center called a centroid.
- k is specified by the user.

Given k, the k-means algorithm works as follows:

```
Algorithm k-means (k, D)
1. Identify the k data points as the initial centroids (cluster centers).
2. Repeat step 1.
3. For each data point x ε D do.
4. Compute the distance from x to the centroid.
5. Assign x to the closest centroid (a centroid represents a cluster).
6. End for
7. Re-compute the centroids using the current cluster memberships until
    the stopping criterion is met.
```

Convergence or stopping criteria for the k-means clustering:

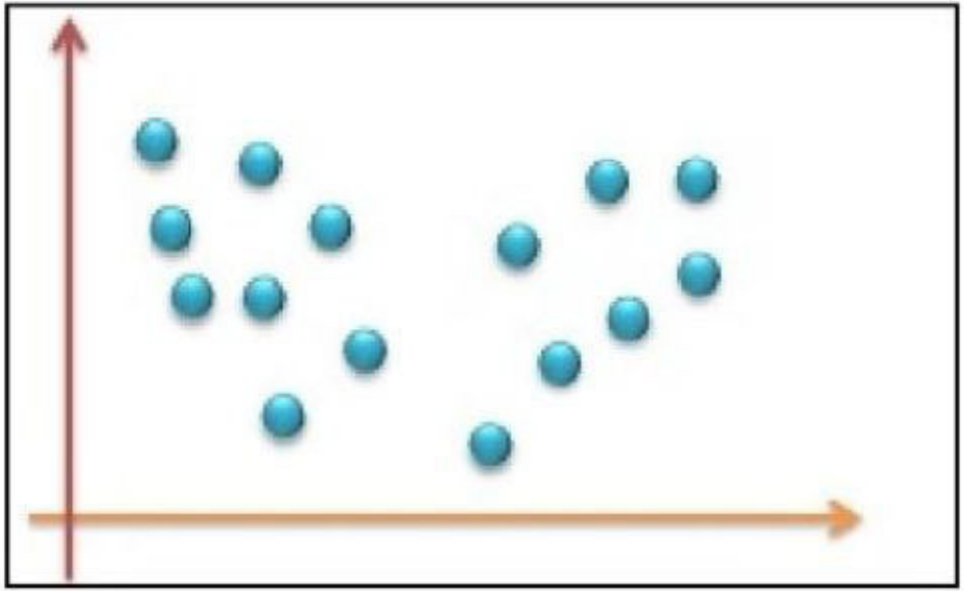
The following list describes the convergence criteria for the k-means clustering algorithm:

- There are zero or minimum number of reassignments for the data points to different clusters
- There are zero or minimum changes of centroids
- Otherwise, the decrease in the sum of squared error of prediction (SSE) is minimum

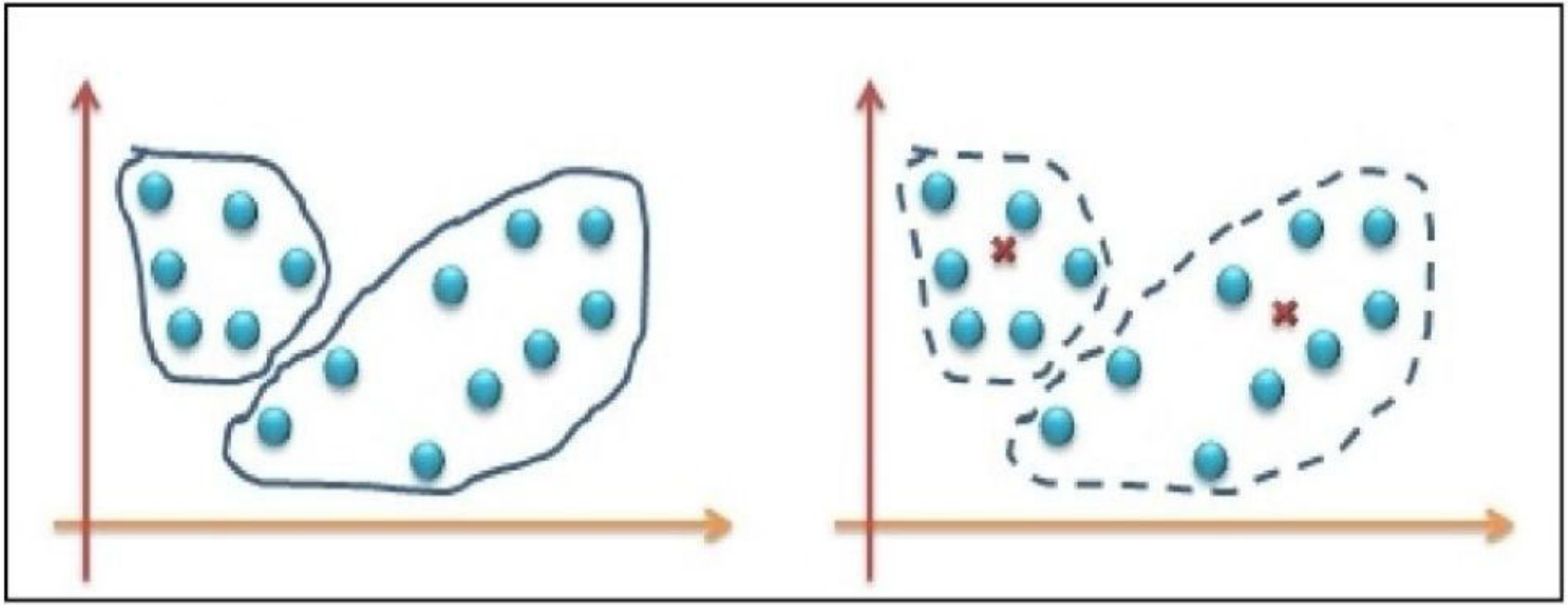
If Cj is the jth cluster, then mj is the centroid of cluster Cj(the mean vector of all the data points in Cj), and if dist(x, mj) is the distance between the data point x and centroid mj then the following example demonstrated using graphical representation explains the convergence criteria.

For example:

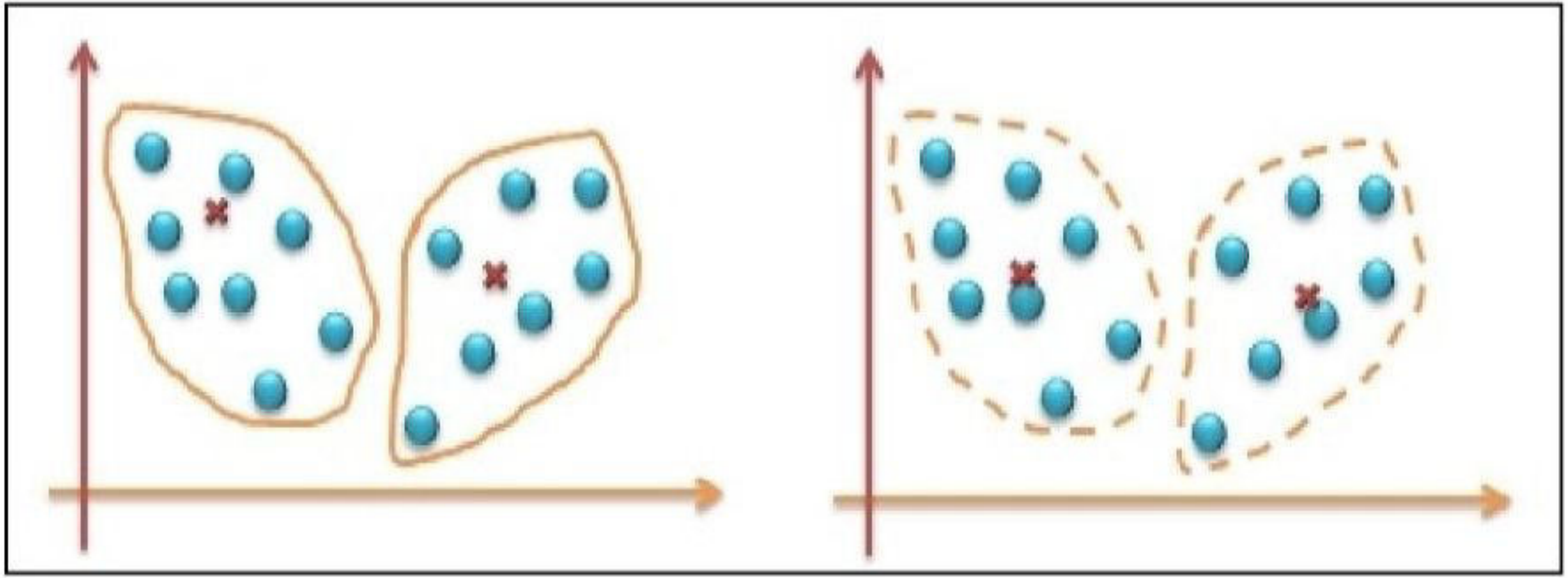
1. Identification of random k centers:



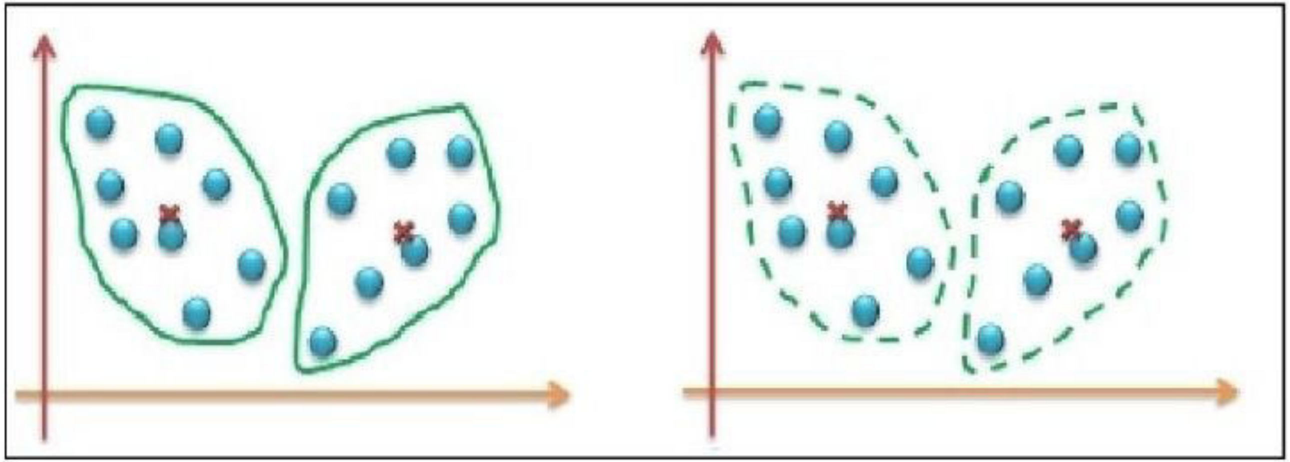
2. Iteration 1: Compute centroids and assign the clusters:



3. Iteration 2: Recompute centroids and reassign the clusters:



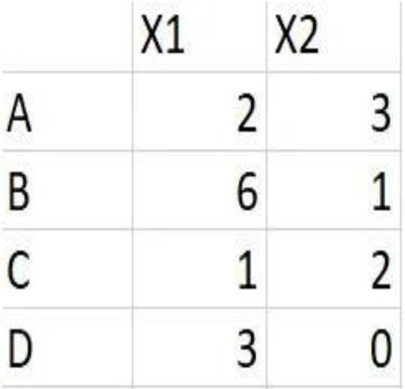
4. Iteration 3: Recompute centroids and reassign the clusters:



5. Terminate the process due to minimal changes to centroids or cluster reassignments.

Example

1. Consider 4 data points A,B,C,D as below

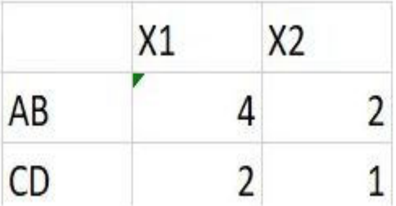


	X1	X2
A	2	3
B	6	1
C	1	2
D	3	0

Observations

2. Choose two centroids AB and CD, calculated as

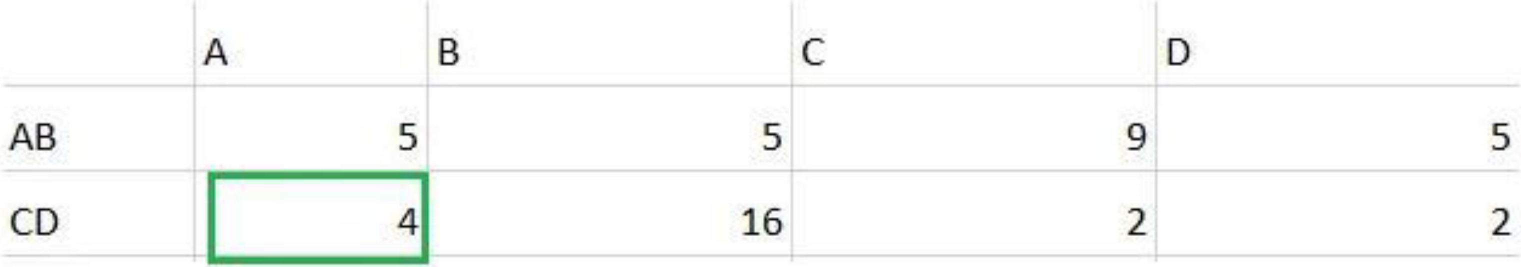
> *AB = Average of A, B*
>
> *CD = Average of C,D*



	X1	X2
AB	4	2
CD	2	1

Two centroids AB, CD

3. Calculate squared euclidean distance between all data points to the centroids AB, CD. For example distance between A(2,3) and AB (4,2) can be given by $s = (2-4)^2 + (3-2)^2$.



	A	B	C	D
AB	5	5	9	5
CD	4	16	2	2

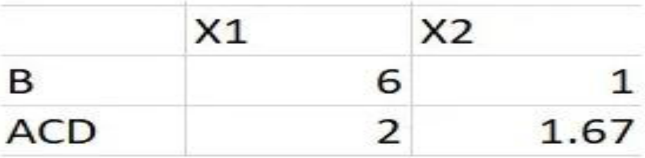
A is very near to CD than AB

4. If we observe in the fig, the highlighted *distance between (A, CD) is 4 and is less compared to (AB, A) which is 5. Since point A is close to the CD we can move A to CD cluster.*

5. There are two clusters formed so far, let recompute the centroids i.e, B, ACD similar to step 2.

$$ACD = Average\ of\ A,\ C,\ D$$

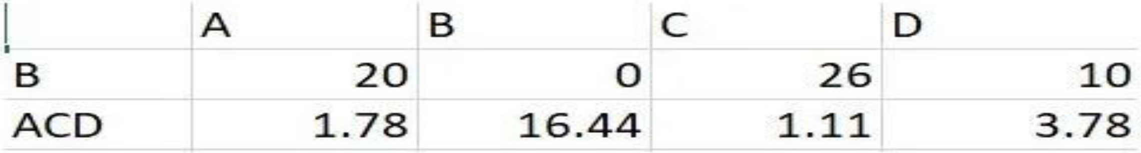
$$B = B$$



	X1	X2
B	6	1
ACD	2	1.67

New centroids B, ACD

6. As we know K-Means is iterative procedure now we have to calculate the distance of all points (A, B, C, D) to new centroids (B, ACD) similar to step 3.



	A	B	C	D
B	20	0	26	10
ACD	1.78	16.44	1.11	3.78

Clusters B, ACD

7. In the above picture, we can see respective cluster values are minimum that A is too far from cluster B and near to cluster ACD. All data points are assigned to clusters (B, ACD) based on their minimum distance. The iterative procedure ends here.

8. To conclude, we have started with two centroids and end up with two clusters, K=2.

Hierarchical Clustering

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster.

In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram.**

The hierarchical clustering technique has two approaches:

1. **Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as** single clusters and merging them until one cluster is left.
2. **Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.**

1. Agglomerative Hierarchical clustering

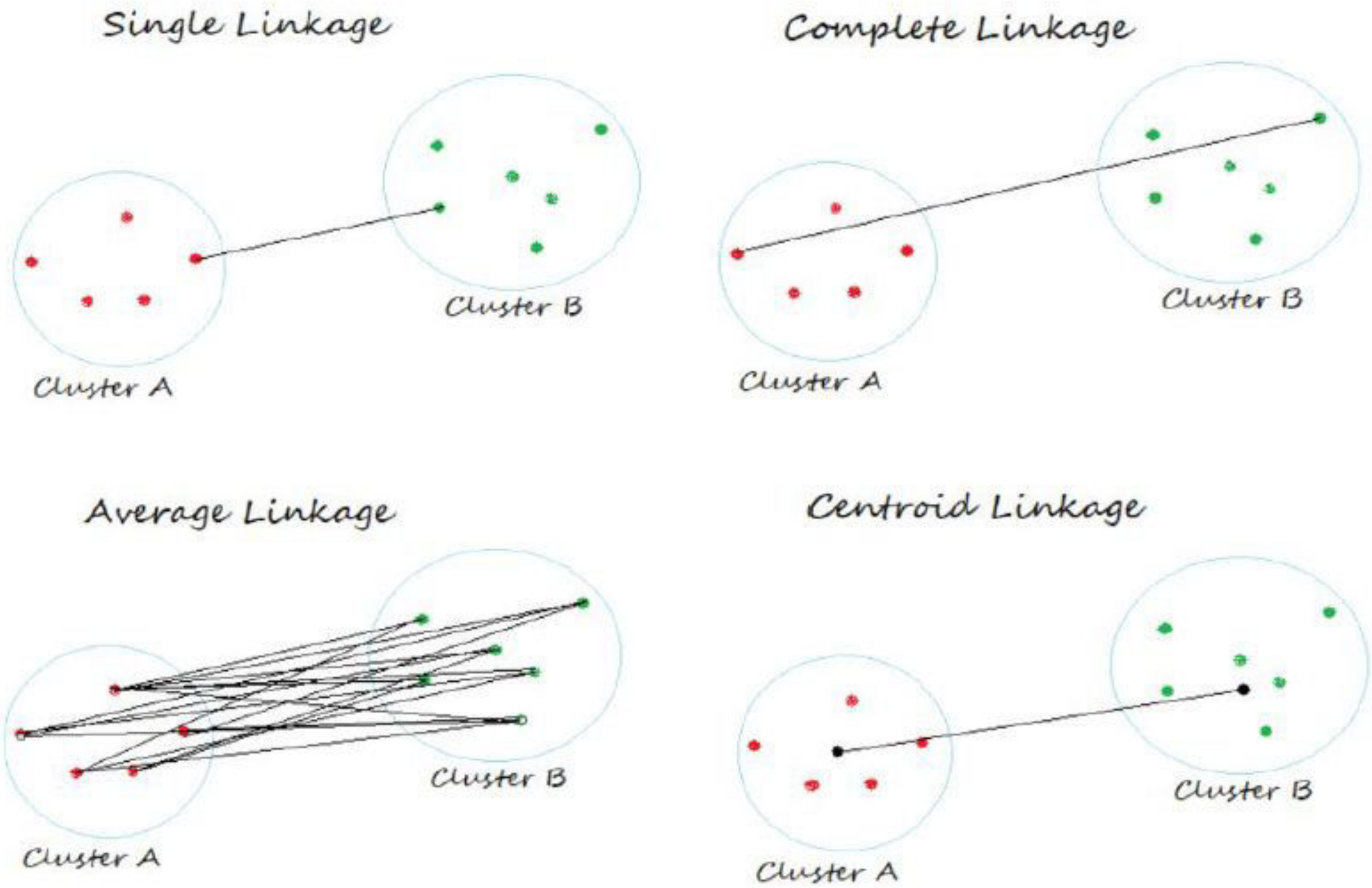
To group the datasets into clusters, it follows the **bottom-up approach.**

It means, this algorithm considers each data point as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the data points.

This hierarchy of clusters is represented in the form of the dendrogram.

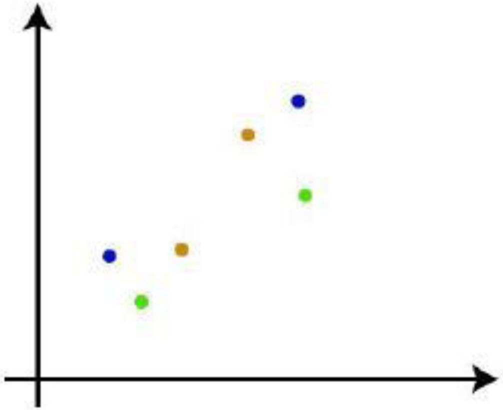
Clusters are merged based on the distance between them and to calculate the distance between the clusters we have different types of Linkage.

- In Single Linkage, the distance between two clusters is the minimum distance between members of the two clusters
- In Complete Linkage, the distance between two clusters is the maximum distance between members of the two clusters
- In Average Linkage, the distance between two clusters is the average of all distances between members of the two clusters
- In Centroid Linkage, the distance between two clusters is the distance between their centroids

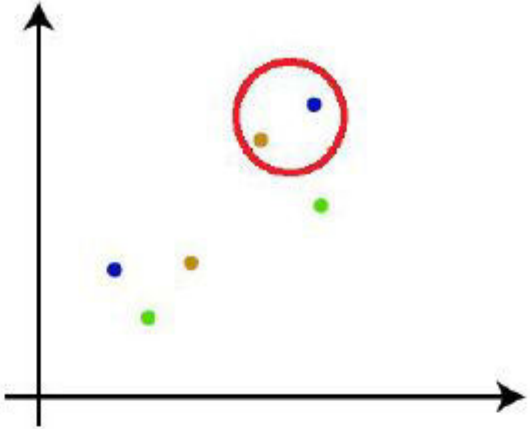

Single Linkage
Cluster A
Cluster B
Complete Linkage
Cluster A
Cluster B
Average Linkage
Cluster A
Cluster B
Centroid Linkage
Cluster A
Cluster B

How the Agglomerative Hierarchical clustering Work?

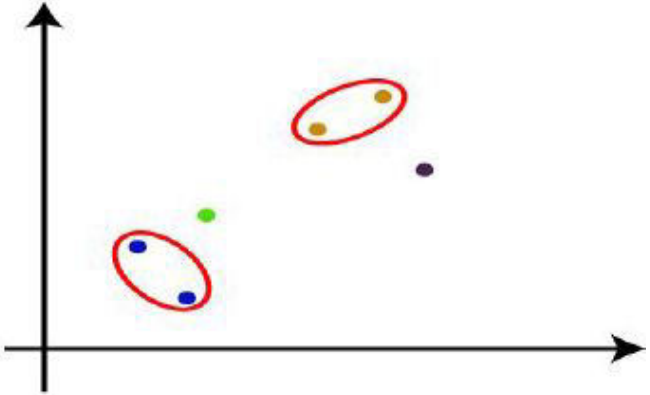
Step-1: Create each data point as a single cluster. Let's say there are N data points, so the number of clusters will also be N.



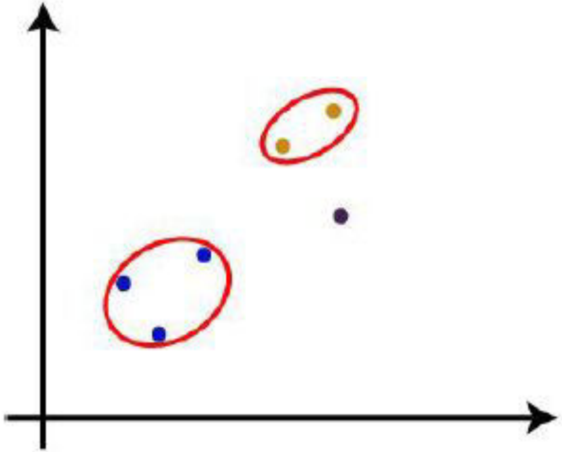
Step-2: Take two closest data points or clusters and merge them to form one cluster. So, there will now be N-1 clusters.

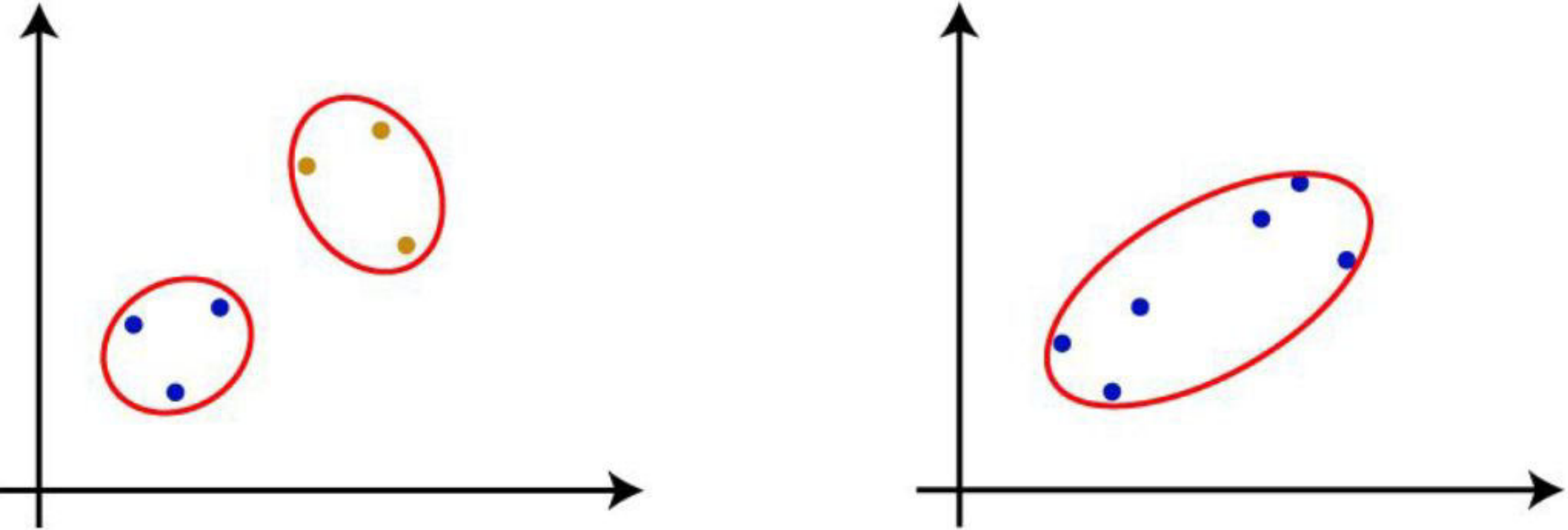


Step-3: Again, take the two closest clusters and merge them together to form one cluster. There will be N-2 clusters.



Step-4: Repeat Step 3 until only one cluster left. So, we will get the following clusters. Consider the below images:

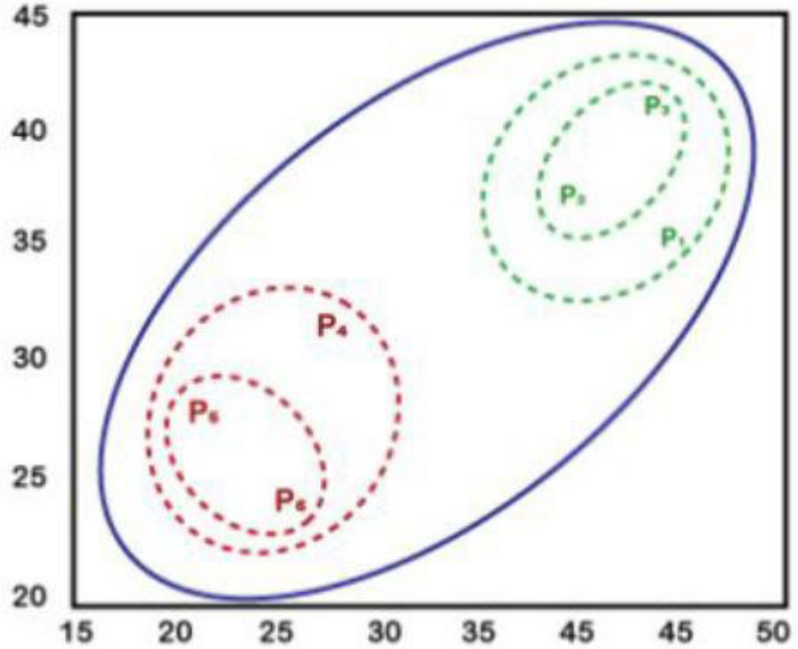


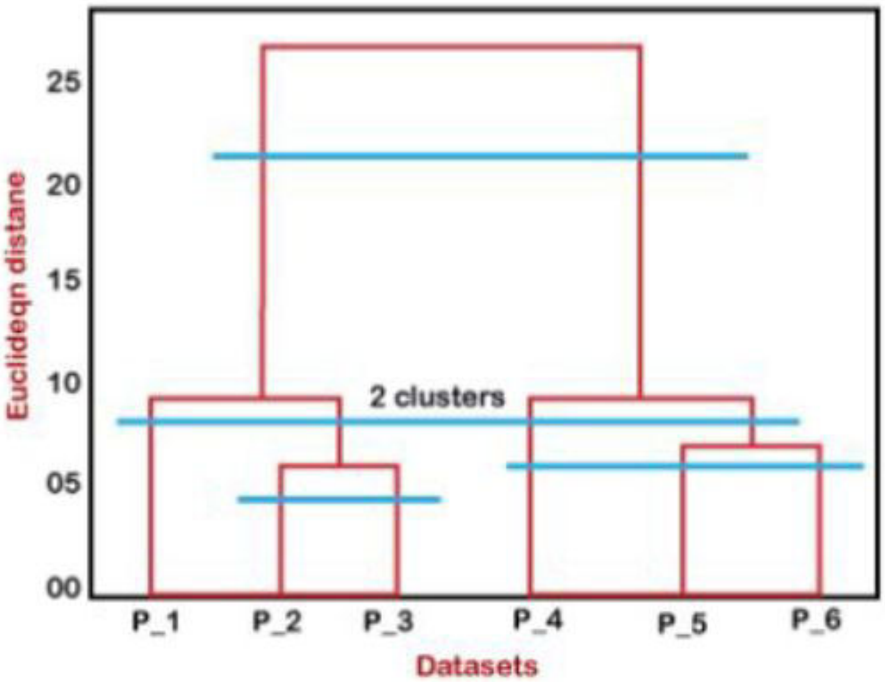


- **Step-5: Once all the clusters are combined into one big cluster, develop the dendrogram to divide the clusters as per** the problem.

Woking of Dendrogram in Hierarchical clustering

- The dendrogram is a tree-like structure that is mainly used to store each step as a memory that the HC algorithm performs.
- In the dendrogram plot, the Y-axis shows the Euclidean distances between the data points, and the x-axis shows all the data points of the given dataset.
- In the below diagram, the left part is showing how clusters are created in agglomerative clustering, and the right part is showing the corresponding dendrogram.

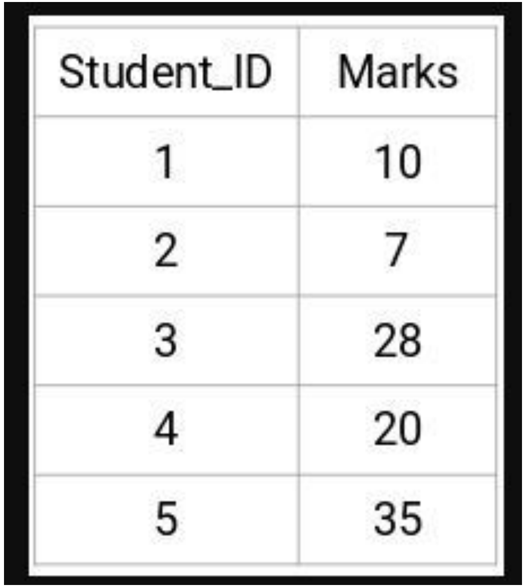

45
40
35
30
25
20
15
20
25
30
35
45
45
50
P_3
P_2
P_1
P_4
P_6
P_5


25
20
15
10
05
00
Euclideqn distane
2 clusters
P_1
P_2
P_3
P_4
P_5
P_6
Datasets

- Firstly, the data points P2 and P3 combine together and form a cluster, correspondingly a dendrogram is created, which connects P2 and P3 with a rectangular shape. The height is decided according to the Euclidean distance between the data points.
- In the next step, P5 and P6 form a cluster, and the corresponding dendrogram is created. It is higher than of previous, as the Euclidean distance between P5 and P6 is a little bit greater than the P2 and P3.
- Again, two new dendrograms are created that combine P1, P2, and P3 in one dendrogram, and P4, P5, and P6, in another dendrogram.
- At last, the final dendrogram is created that combines all the data points together.

Example

Let’s take a sample of 5 students:

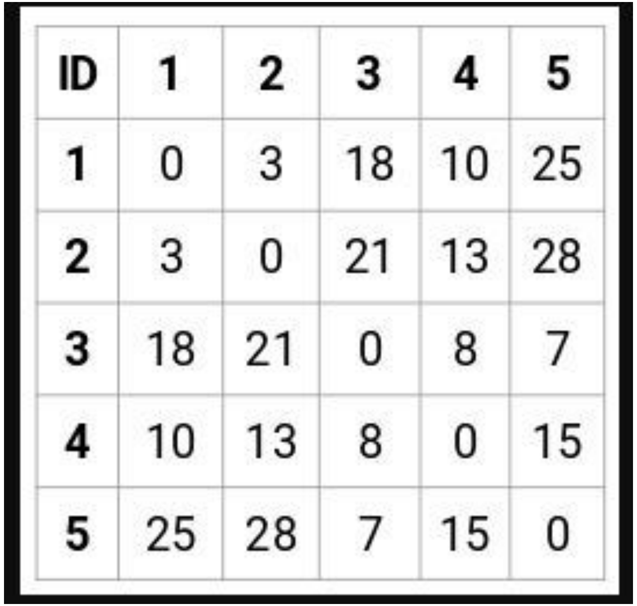


Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

Creating a Proximity Matrix

First, we will create a proximity matrix which will tell us the distance between each of these points. Since we are calculating the distance of each point from each of the other points, we will get a square matrix of shape n X n (where n is the number of observations).

Let’s make the 5 x 5 proximity matrix for our example:



ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

The diagonal elements of this matrix will always be 0 as the distance of a point with itself is always 0. We will use the Euclidean distance formula to calculate the rest of the distances. So, let’s say we want to calculate the distance between point 1 and 2:

√(10-7)^2 = √9 = 3

Similarly, we can calculate all the distances and fill the proximity matrix.