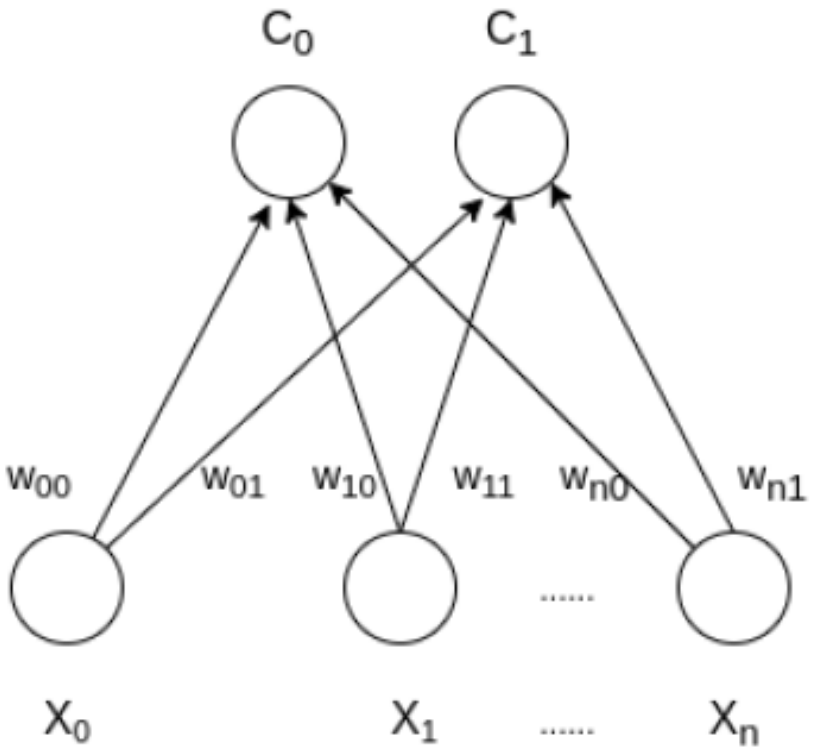
SOM – Self Organized Maps
Kmean Clustering
Hierarchical Clustering

Sandeep Chaurasia

Self Organizing Maps – Kohonen Maps

- **Self Organizing Map (or Kohonen Map or SOM)** is a type of Artificial Neural Network which is also inspired by biological models of neural systems from the 1970s.
- It follows an unsupervised learning approach and trained its network through a competitive learning algorithm.
- SOM is used for clustering and mapping (or dimensionality reduction) techniques to map multidimensional data onto lower-dimensional which allows people to reduce complex problems for easy interpretation.
- SOM has two layers, one is the Input layer and the other one is the Output layer.
- The architecture of the Self Organizing Map with two clusters and n input features of any sample is given below:



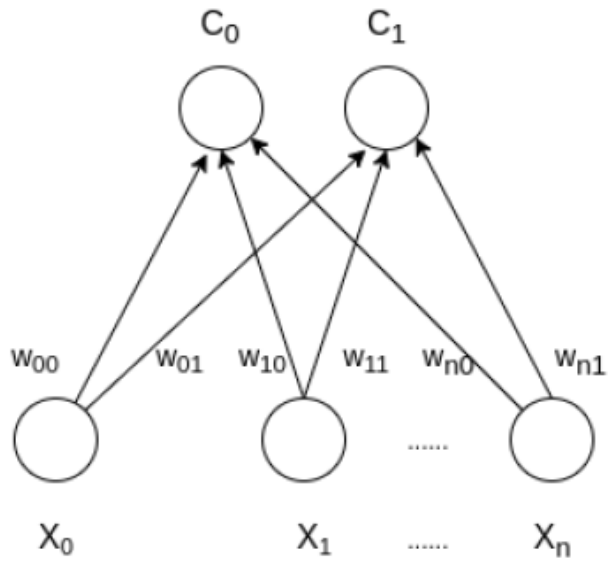
Self Organizing Maps – Kohonen Maps

- Let’s say an input data of size (m, n) where m is the number of training examples and n is the number of features in each example. F
- irst, it initializes the weights of size (n, C) where C is the number of clusters.
- Then iterating over the input data, for each training example, it updates the winning vector (weight vector with the shortest distance (e.g Euclidean distance) from training example).
- Weight updation rule is given by :

$$wij = wij(old) + alpha(t) * (xik - wij(old))$$

where alpha is a learning rate at time t, j denotes the winning vector, i denotes the i^{th} feature of training example and k denotes the k^{th} training example from the input data.

After training the SOM network, trained weights are used for clustering new examples. A new example falls in the cluster of winning vectors.



Self Organizing Maps – Kohonen Maps

- **Training:**
- **Step 1:** Initialize the weights w_{ij} random value may be assumed. Initialize the learning rate α.
- **Step 2:** Calculate squared Euclidean distance.

  $$D(j) = \Sigma (wij - xi)^2 \quad \text{where } i=1 \text{ to } n \text{ and } j=1 \text{ to } m$$
- **Step 3:** Find index J, when D(j) is minimum that will be considered as winning index.
- **Step 4:** For each j within a specific neighborhood of j and for all i, calculate the new weight.

  $$wij(new)=wij(old) + \alpha[xi - wij(old)]$$
- **Step 5:** Update the learning rule by using :

  $$\alpha(t+1) = 0.5 * t$$
- **Step 6:** Test the Stopping Condition.

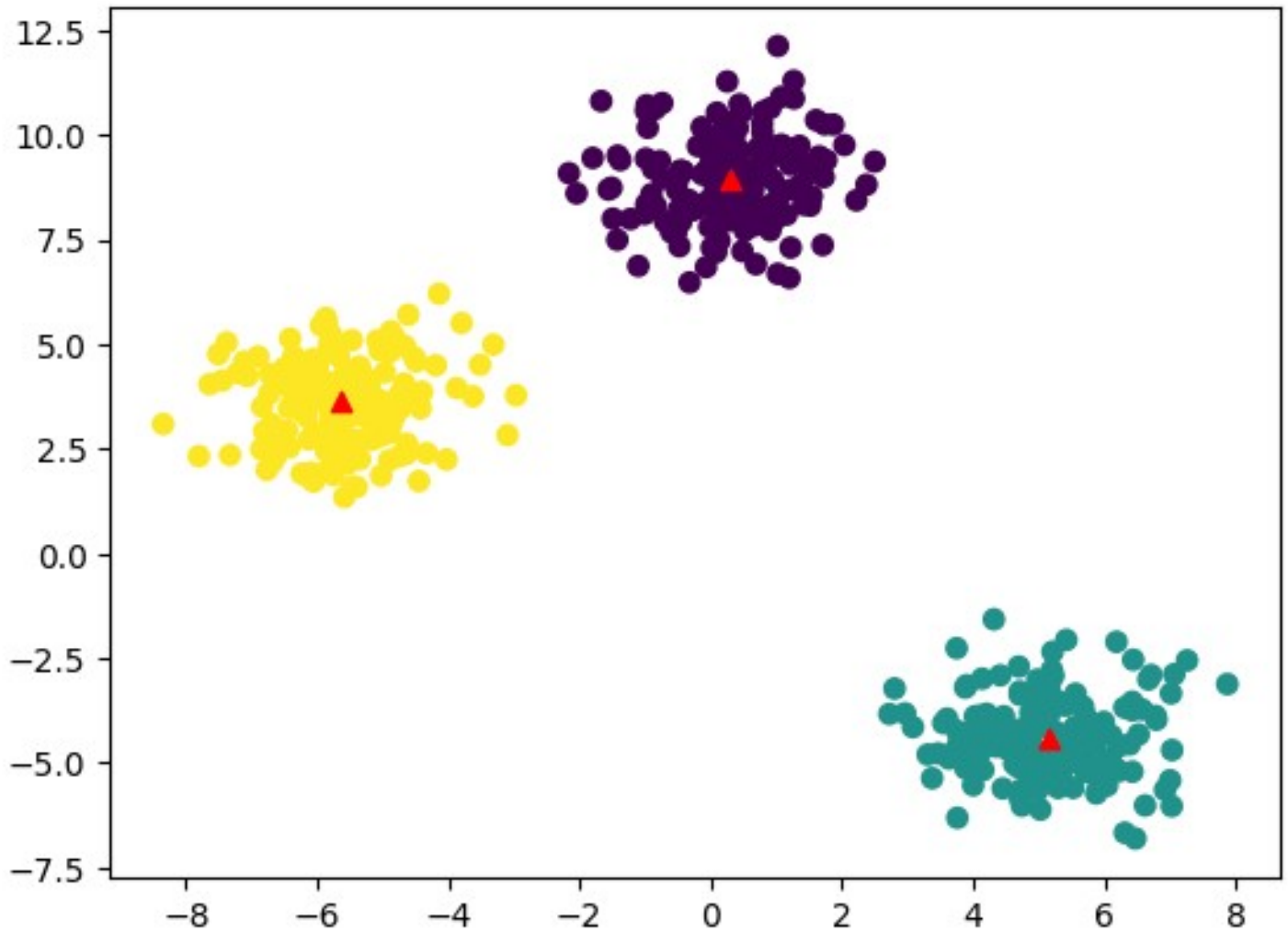
SOM Example

KMean Clustering

- K-Means Clustering is an Unsupervised Machine Learning algorithm, which groups the unlabeled dataset into different clusters.
- Unsupervised Machine Learning is the process of teaching a computer to use unlabeled, unclassified data and enabling the algorithm to operate on that data without supervision. Without any previous data training, the machine's job in this case is to organize unsorted data according to parallels, patterns, and variations.
- The goal of clustering is to divide the population or set of data points into a number of groups so that the data points within each group are more comparable to one another and different from the data points within the other groups. It is essentially a grouping of things based on how similar and different they are to one another.

- We are given a data set of items, with certain features, and values for these features (like a vector). The task is to categorize those items into groups. To achieve this, we will use the K-means algorithm; an unsupervised learning algorithm. ‘K’ in the name of the algorithm represents the number of groups/clusters we want to classify our items into.

- The algorithm works as follows:
  - First, we randomly initialize k points, called means or cluster centroids.
  - We categorize each item to its closest mean and we update the mean’s coordinates, which are the averages of the items categorized in that cluster so far.
  - We repeat the process for a given number of iterations and at the end, we have our clusters.


12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−8
−6
−4
−2
0
2
4
6
8

Example K-Mean

Hierarchal Clustering

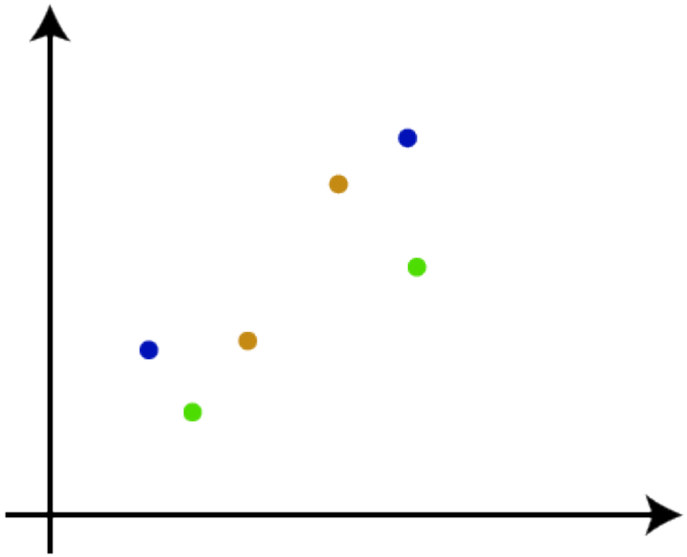
- Hierarchical Clustering in Machine Learning: Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA.
- In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

The hierarchical clustering technique has two approaches:

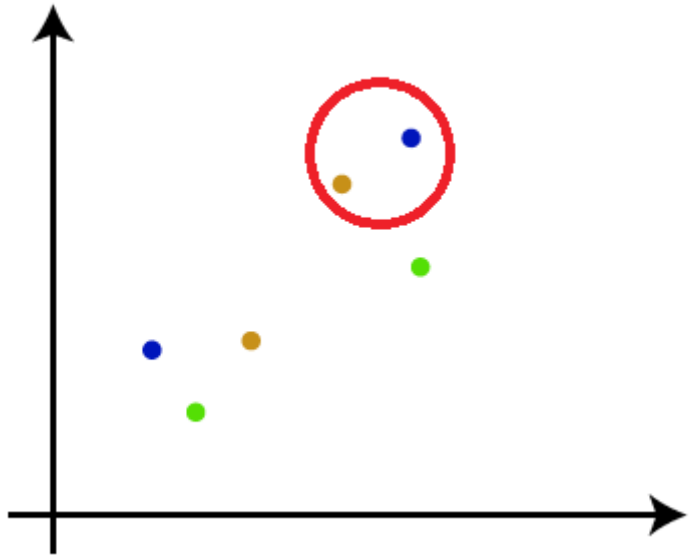
- Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
- Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

Agglomerative Hierarchical clustering

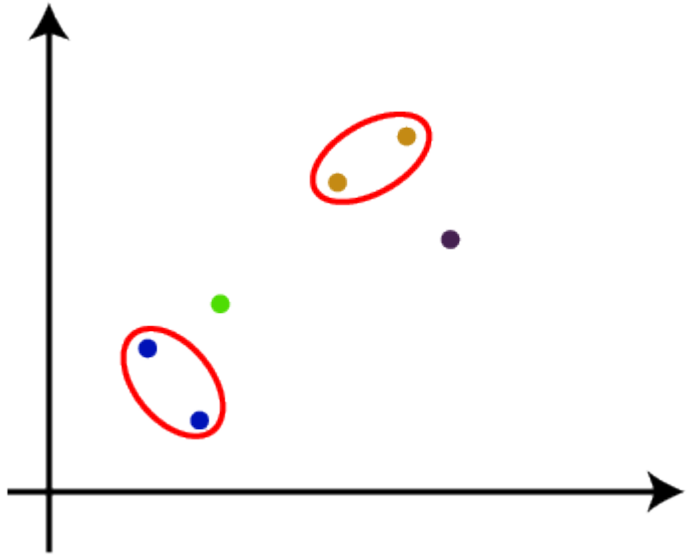
- The agglomerative hierarchical clustering algorithm is a popular example of HCA. To group the datasets into clusters, it follows the **bottom-up approach**.
- It means, this algorithm considers each dataset as a single cluster at the beginning, and then start combining the closest pair of clusters together. It does this until all the clusters are merged into a single cluster that contains all the datasets.
- This hierarchy of clusters is represented in the form of the dendrogram.



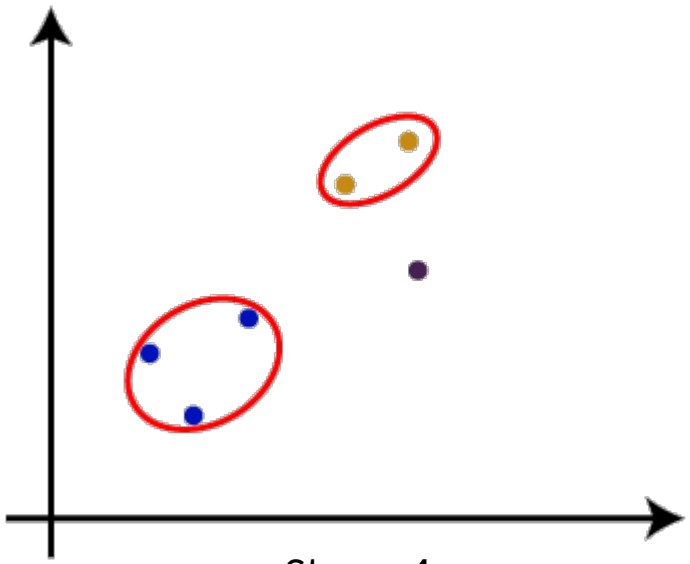
Step - 1



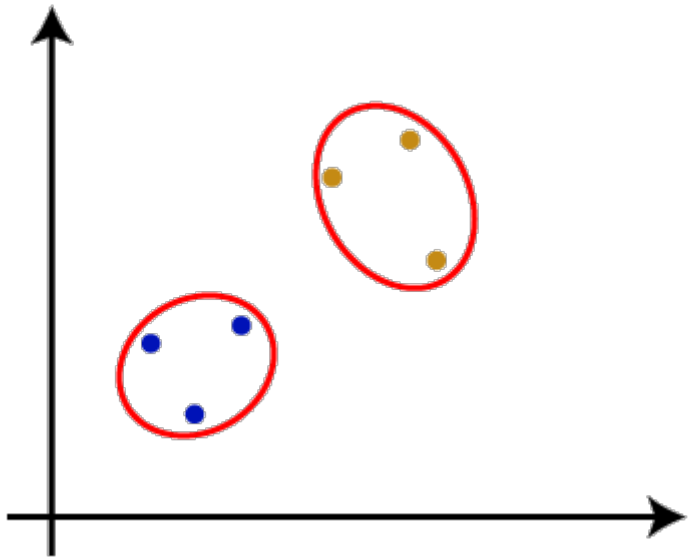
Step - 2



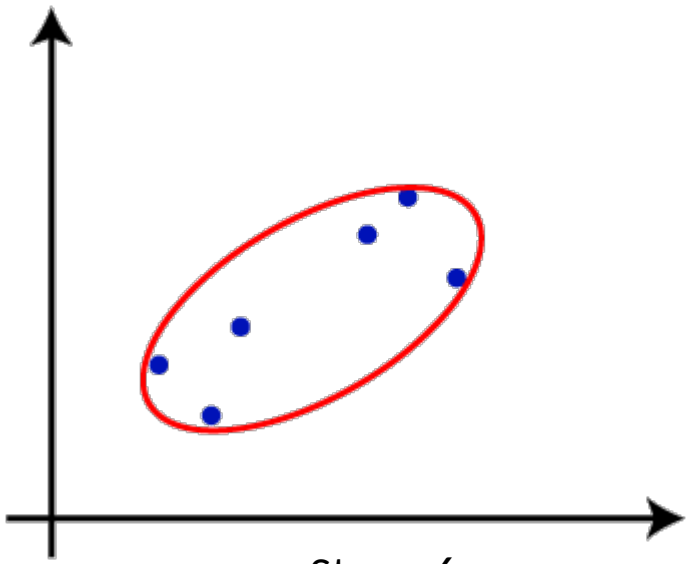
Step - 3



Step - 4



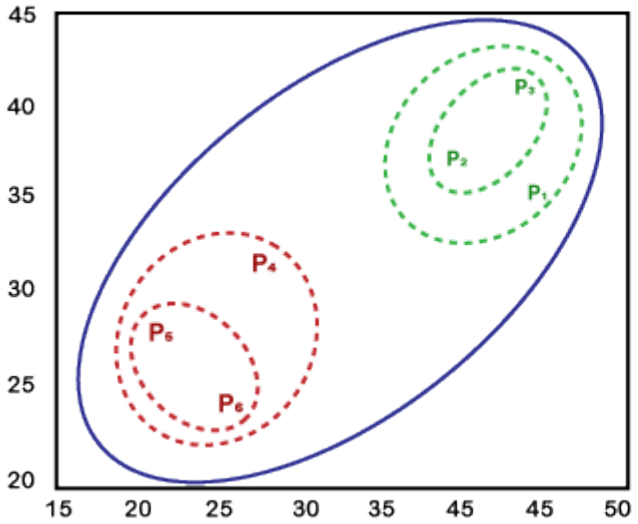
Step - 5

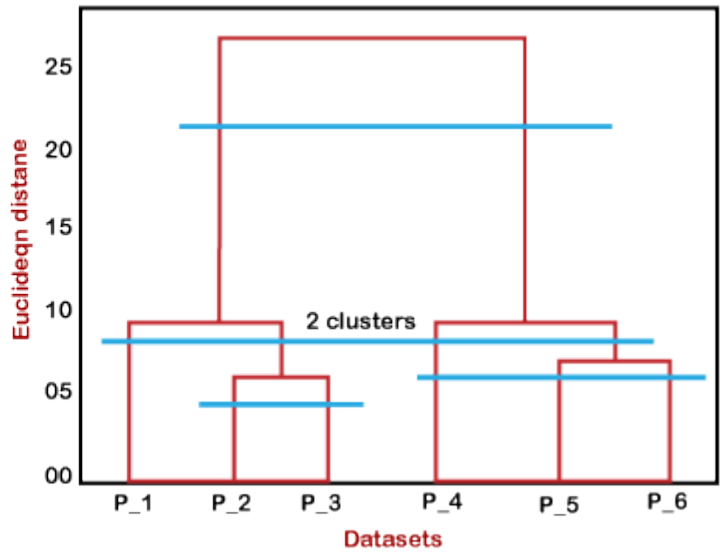


Step - 6

Working 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





Measure for the distance between two clusters

The **closest distance** between the two clusters is crucial for the hierarchical clustering. There are various ways to calculate the distance between two clusters, and these ways decide the rule for clustering. These measures are called **Linkage methods**. Some of the popular linkage methods are given below:

Single Linkage: It is the Shortest Distance between the closest points of the clusters.

Complete Linkage: It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.

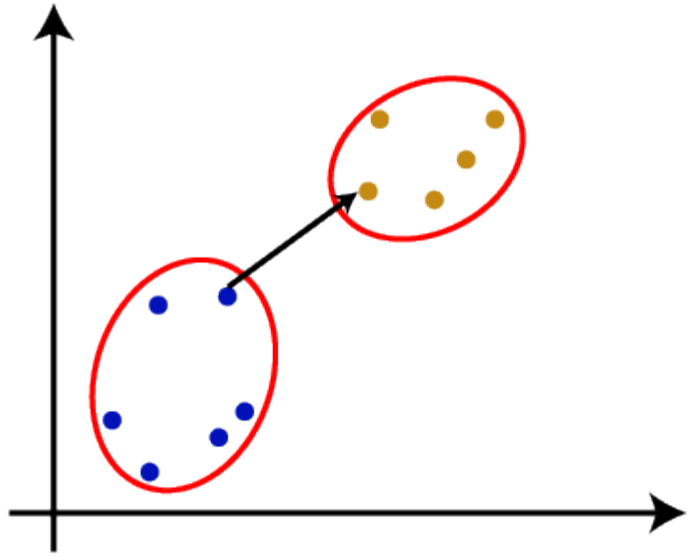
Average Linkage: It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.

Centroid Linkage: It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:

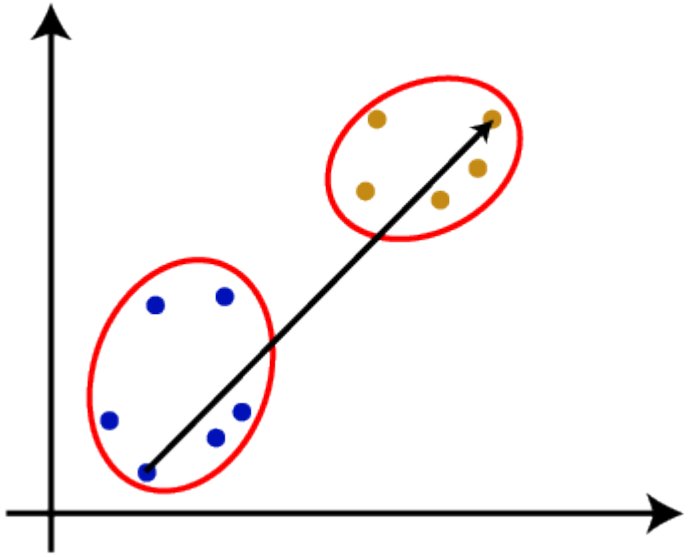
Single Linkage: It is the Shortest Distance between the closest points of the clusters.

Complete Linkage: It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.

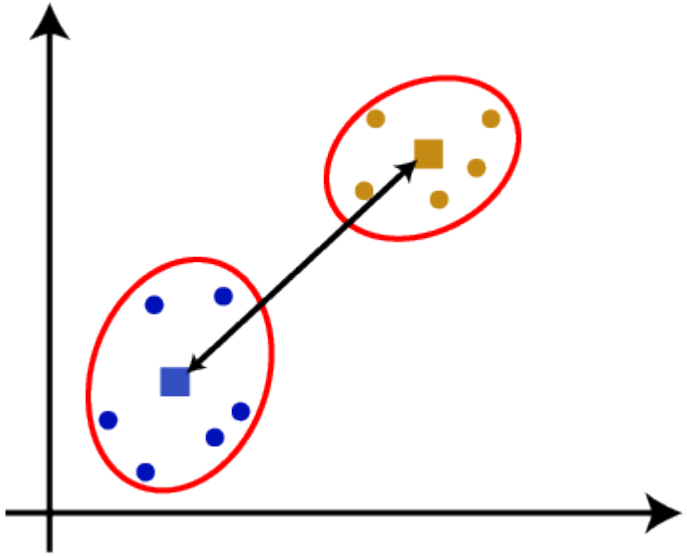
Average Linkage: It is the linkage method in which the distance between each pair of datasets is added up and then divided by the total number of datasets to calculate the average distance between two clusters. It is also one of the most popular linkage methods.



Single Linkage: It is the Shortest Distance between the closest points of the clusters



Complete Linkage: It is the farthest distance between the two points of two different clusters. It is one of the popular linkage methods as it forms tighter clusters than single-linkage.



Centroid Linkage: It is the linkage method in which the distance between the centroid of the clusters is calculated. Consider the below image:

Single Linkage Example