Clustering

K-means & Hierarchal-Clustering

What is Clustering?

Cluster analysis is a technique in machine learning that groups similar objects into clusters. K-means clustering, a popular method, aims to divide a set of objects into K clusters, minimizing the sum of squared distances between the objects and their respective cluster centers.

Clustering vs. Classification:

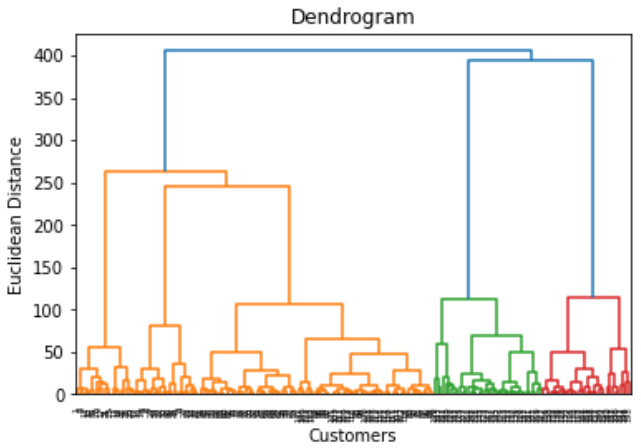
While both clustering and classification involve grouping data, they differ in their approach. Classification requires labeled data (i.e., data points already assigned to specific classes), while clustering works with unlabeled data. In essence, clustering is about discovering the classes (or clusters) themselves, whereas classification assigns data points to pre-defined classes

Clustering Models

Hierarchical clustering and **k-means clustering** are two popular techniques in the field of unsupervise used for clustering data points into distinct gr

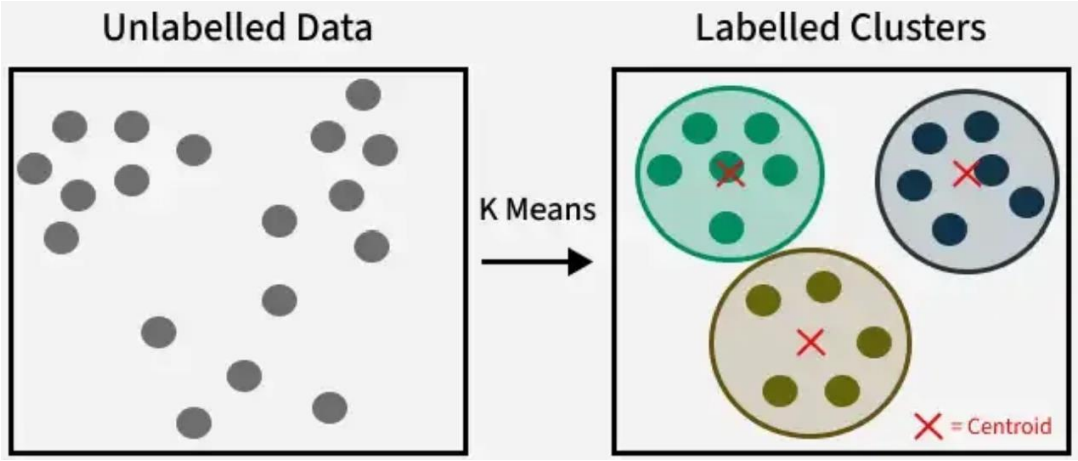
k-means clustering divides data into a predet of clusters.

hierarchical clustering creates a hierarchical t structure to represent the relationships betw clusters.



K-Means

K-Means Clustering is an Unsupervised Machine Learning algorithm which groups unlabeled dataset into different clusters. It is used to organize data into **groups based on their similarity**. and discover underlying patterns or structures within the data



K-means aims to partition a dataset into K clusters, where each data point belongs to the cluster with the nearest centroid

Applications of Clustering in Real-World Scenarios

1- Customer Segmentation

2- Document Clustering

3- Image Segmentation

4- Recommendation Engines

Advantages of K-means

1. It is very simple to implement.
2. It is scalable to a huge data set and also faster to large datasets.
3. it adapts the new examples very frequently.
4. Generalization of clusters for different shapes and sizes.

Disadvantages of K-means

1. It is sensitive to the outliers.
2. Choosing the k values manually is a tough job.
3. As the number of dimensions increases its scalability decreases.

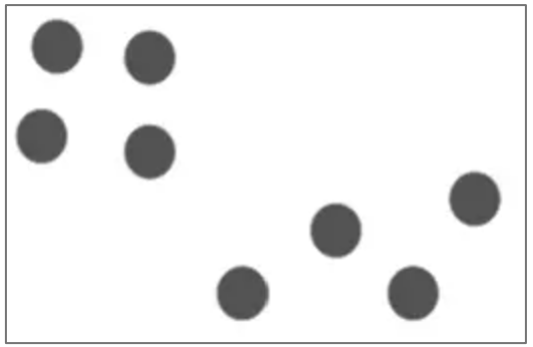
How K-Means Clustering Works?

Here's how it works:

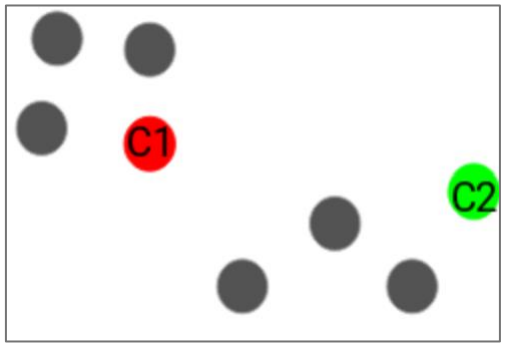
1. **Initialization**: Start by randomly selecting K points from the dataset. These points will act as the initial cluster centroids.
2. **Assignment**: For each data point in the dataset, calculate the distance between that point and each of the K centroids. Assign the data point to the cluster whose centroid is closest to it. This step effectively forms K clusters.
3. **Update centroids**: Once all data points have been assigned to clusters, recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster.
4. **Repeat**: Repeat steps 2 and 3 until convergence. Convergence occurs when the centroids no longer change significantly or when a specified number of iterations is reached.
5. **Final Result**: Once convergence is achieved, the algorithm outputs the final cluster centroids and the assignment of each data point to a cluster.

How to Apply K-Means Clustering Algorithm?

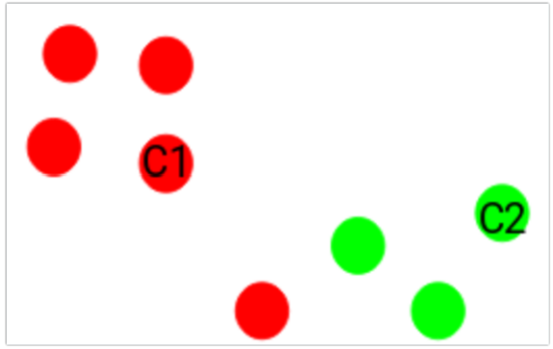
1- Choose the number of clusters *k*



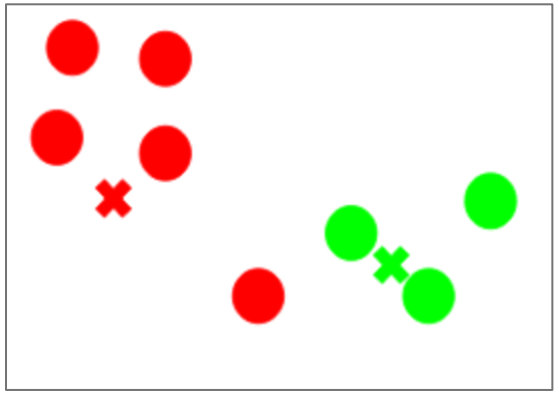
2-Select k random points from the data as centroids



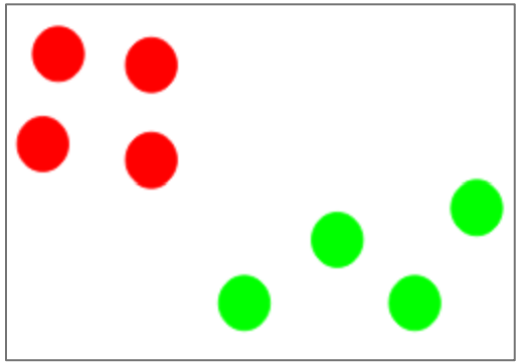
3-Assign all the points to the closest cluster centroid



4-Recompute the centroids of newly formed clusters



5-Repeat steps 3 and 4



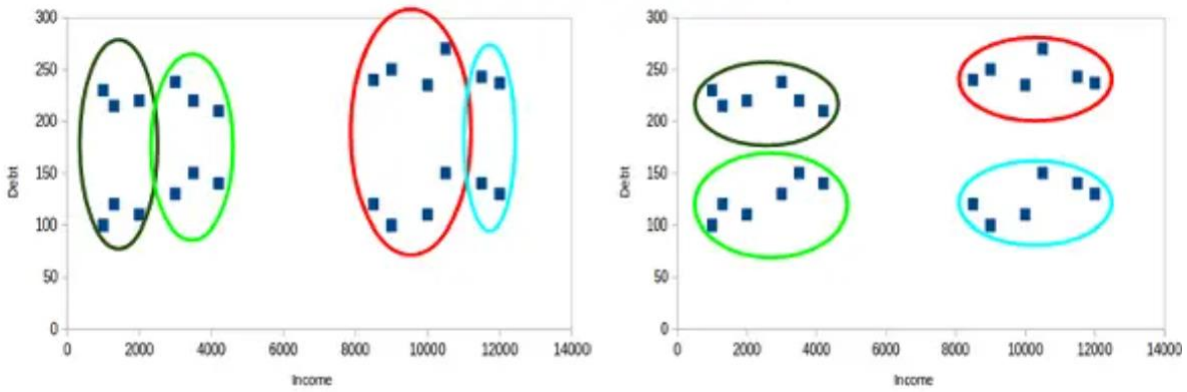
Stopping Criteria for K-Means Clustering

There are essentially three stopping criteria that can be adopted to stop the K-means algorithm:

1. Centroids of newly formed clusters do not change
2. Points remain in the same cluster
3. Maximum number of iterations is reached

The Different Evaluation Metrics for Clustering

Inertia

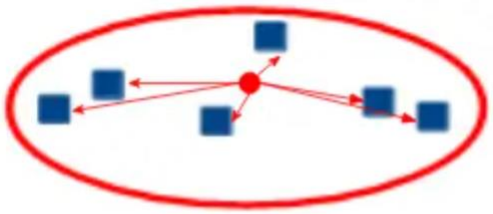


Case - I

Case - II

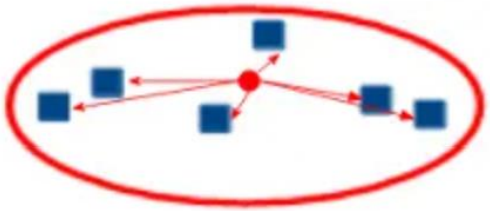
It tells us how far the points within a cluster are. So, inertia actually calculates the sum of distances of all the points within a cluster from the centroid of that cluster

the final inertial value is the sum of all these distances

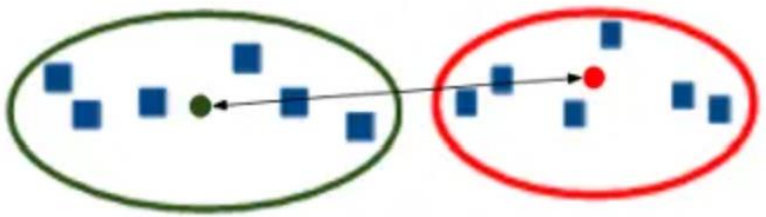


Intra cluster distance

Dunn Index



Intra cluster distance

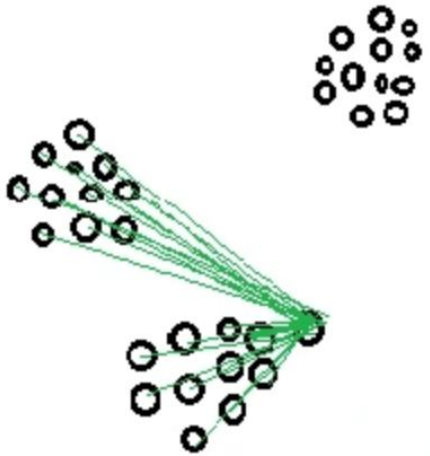


Inter cluster distance

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

silhouette

The silhouette score measures the similarity of each point to its own cluster compared to other clusters, and the silhouette plot visualizes these scores for each sample. A high silhouette score indicates that the clusters are well separated,



It calculates the average distance of points within its cluster a (i) and the average distance of the points to its next closest cluster called b (i).

in Worst case s(i) = -1

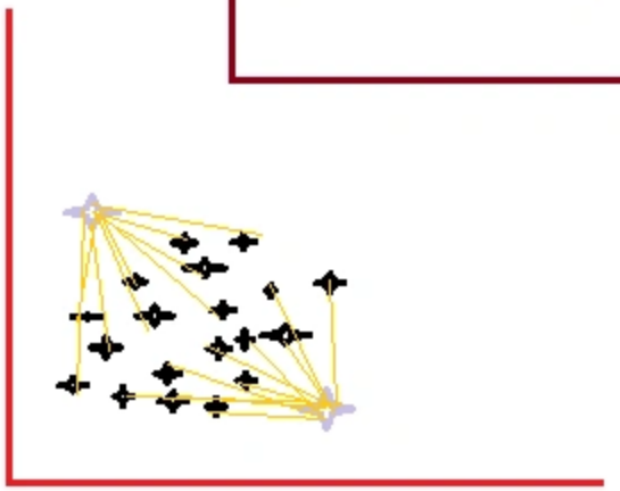
$$s(i) = \frac{b(i) - a(i)}{\text{larger of } b(i) \text{ and } a(i)}$$

a(i) = average distance inside cluster
b(i) = average distance nearest other cluster

How to choose the Best value of K?

One of the most challenging tasks in this clustering algorithm is to choose the right values of k.

with in cluster sum of square (wss) = $\sum_{i=1}^{n} (Ci + Xi)^2$



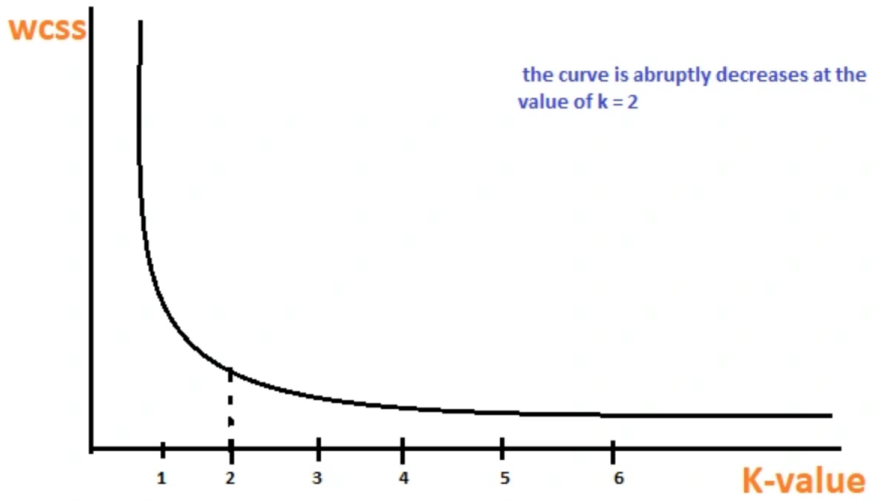
when k = 1

wcss = high

Elbow Method

Elbow is one of the most famous methods by which you can select the right value of k and boost your model performance. We also perform the hyperparameter tuning to chose the best value of k.

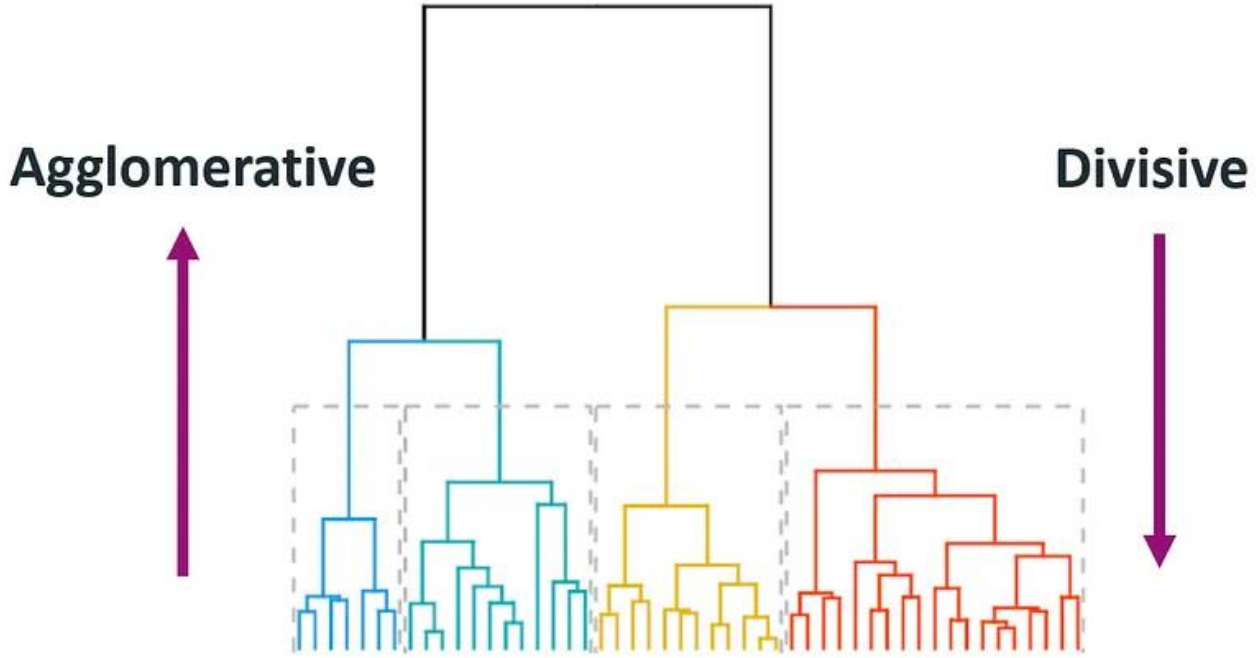
When the value of k is 1, the within-cluster sum of the square will be high. As the value of k increases, the within-cluster sum of square value will decrease.



Hierarchical clustering

Hierarchical clustering

is an unsupervised learning method for clustering data points. The algorithm builds clusters by measuring the dissimilarities between data.



Types of Hierarchical Clustering

1. Agglomerative Clustering (Bottom –Up)
2. Divisive clustering(Top=Down)

Applications of Hierarchical Clustering

Biological Taxonomy

Document Clustering

Image Segmentation

Customer Segmentation

Anomaly Detection

Advantages Of Hierarchical Clustering:

No need to specify the number of clusters beforehand

Dendrogram visualization

Flexibility in cluster shapes

Easy to understand and implement

Flexibility in distance metrics

Dis Advantages Of Hierarchical Clustering:

It is sensitive to outliers.

Hierarchical clustering is computationally expensive.

Hierarchical clustering methods require a predetermined number of clusters before they can begin clustering

The algorithm does not provide any flexibility when dealing with multi-dimensional data sets

The results of Agglomerative or divisive clustering can sometimes be difficult to interpret the results

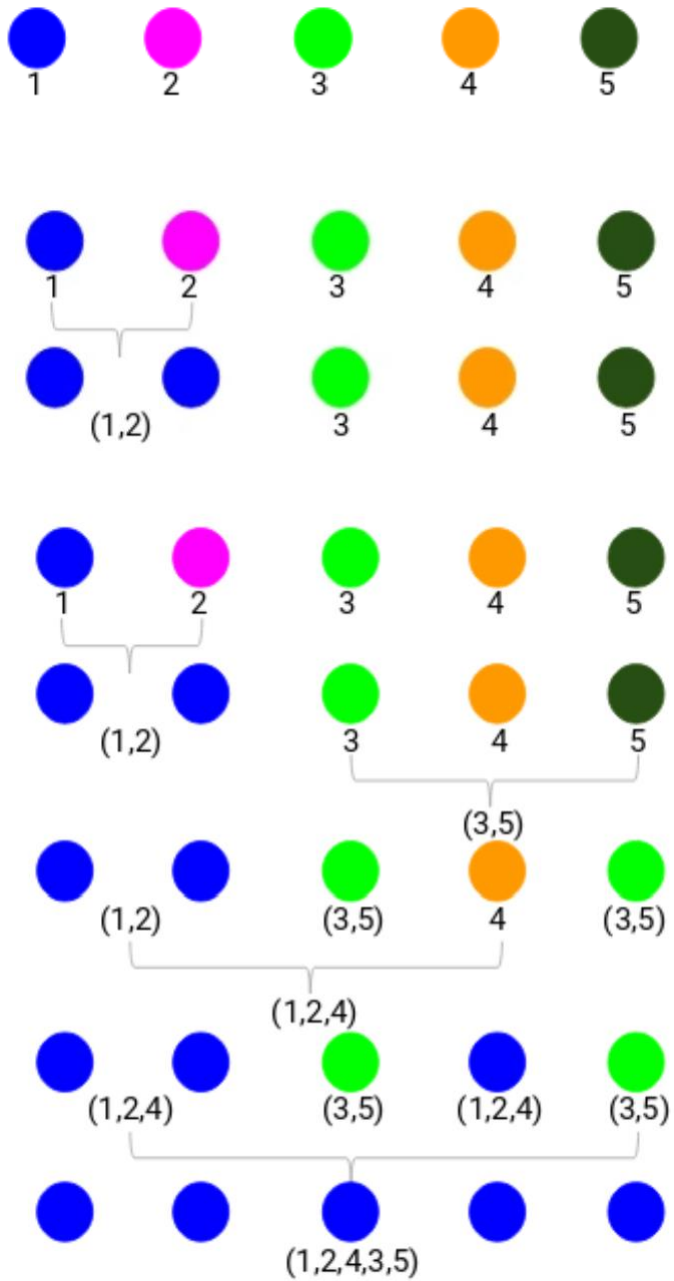
Hierarchical clustering (Step By Step**)**

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

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

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

ID	(1,2)	3	4	5
(1,2)	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0



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

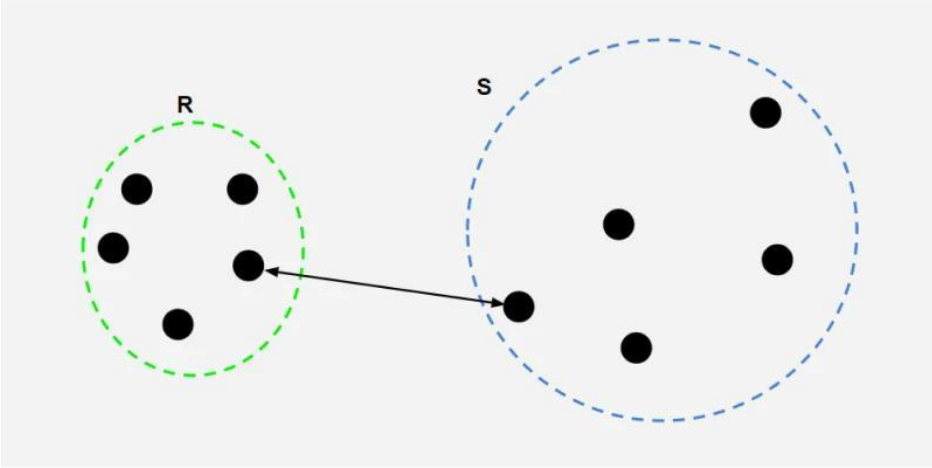
Types of Linkages in Hierarchical Clustering

Hierarchical Clustering is used to group similar data points and organize data in a tree-like structure. Key part of this process is **linkage** which **calculates the distance between clusters before they are merged or divided**.
Different types of linkage is used measure this distance differently. In this article, we'll look at different linkage methods and see how they affect the cluster formation

1. Single Linkage

For two clusters **R** and **S** the **single linkage returns the minimum distance between two points**.

$$L(R,S) = min(D(i,j)), i \epsilon R, j \epsilon S$$

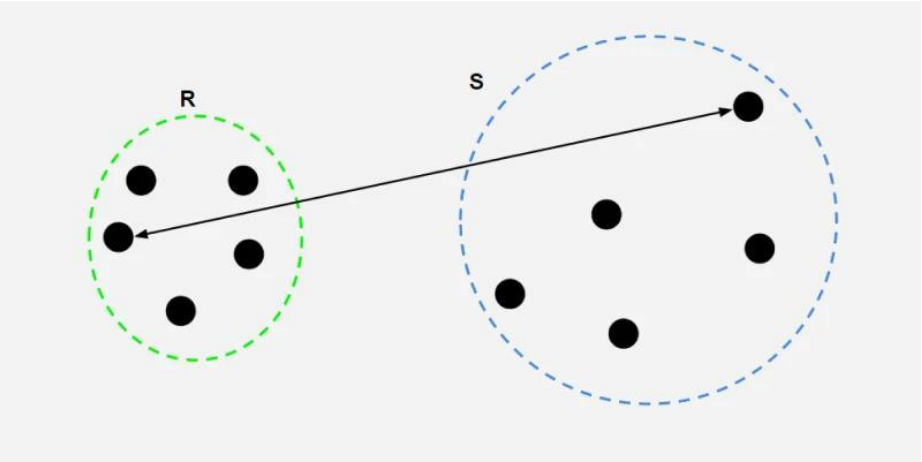


This method creates long, chain-like clusters because it is sensitive to outliers and can connect clusters based on a very small number of close points.

2. Complete Linkage

For two clusters **R** and **S** the **complete linkage returns the maximum distance between two points**.

$$L(R,S) = \max(D(i,j)),\ i \in R,\ j \in S$$



It tends to create compact and spherical clusters because it is more sensitive to outliers and tries to make sure that the clusters are not too far.

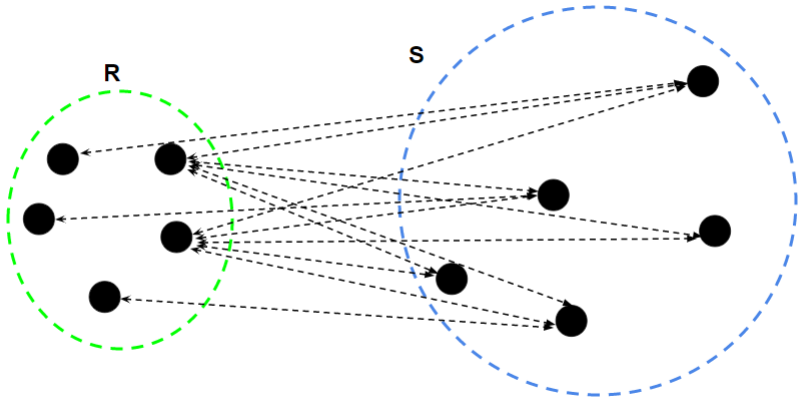
3. Average Linkage

It returns the **average distance between all pairs of points from two clusters**. This method maintain a **balance between single and complete linkage** by considering all pairs of points not just the closest or farthest point.

$$L(R,S) = \frac{1}{n_R \times n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), i \in R, j \in S$$

where

- n_R : Number of data-points in R
- n_S : Number of data-points in S



This method maintain a balance between single and complete linkage by considering all pairs of points not just the closest or farthest point

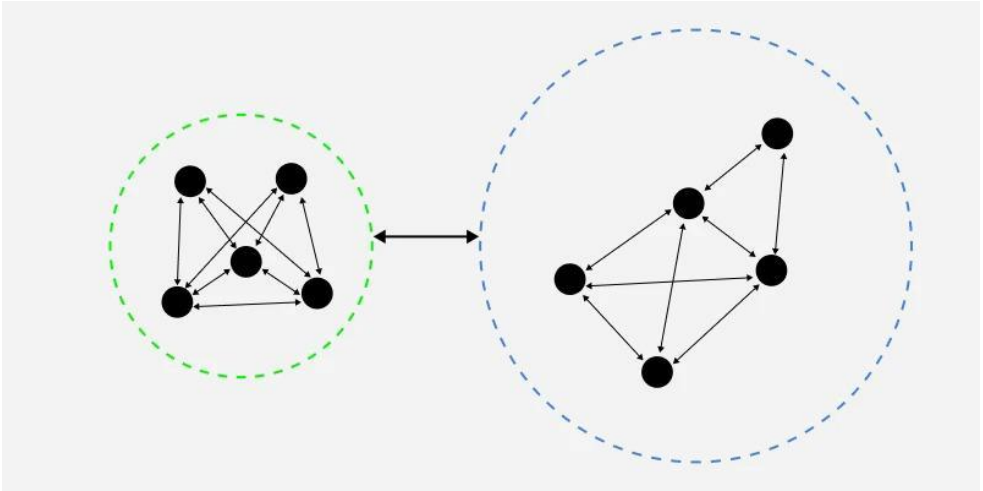
4. Ward's Linkage

It calculates the distance between **two clusters by looking at total spread or variance increase when the clusters are combined**.

$$L(R,S) = \frac{n_R+n_S}{n_R \times n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), \quad i \in R, j \in S$$

where

- n_R and n_S are the sizes of clusters R and S
- D(i, j) is the distance between points $i \in R$ and $j \in S$.



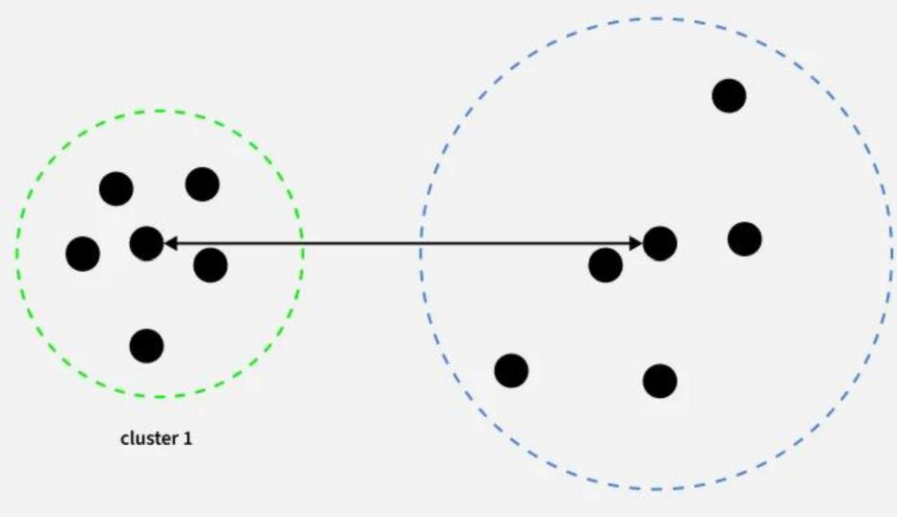
This method creates compact, well-separated clusters by making sure that data within each cluster is as similar as possible.

5. Centroid Linkage

It calculates the distance between two clusters based on the **distance between their central points i.e the average of all points in the cluster**.

$$L(R,S) = D(\bar{R}, \bar{S})$$

- $\bar{R}$ and $\bar{S}$ are the centroids (mean points) of clusters R and S
- $D(\bar{R}, \bar{S})$ is the distance between the centroids of clusters R and S.



This method works well when clusters are round or evenly shaped but it may not be the best for irregularly shaped clusters.