Module IV: Unsupervised Learning

Unsupervised Learning:

Clustering,

K-means clustering,

Hierarchical clustering,

Association,

Apriori algorithm,

Principal component analysis

Unsupervised Learning

- Unsupervised learning is a type of machine learning that analyzes and models data without labelled responses or predefined categories.
- Unsupervised learning algorithms work solely with input data and aim to discover hidden patterns, structures or relationships within the dataset independently, without any human intervention or prior knowledge of the data's meaning.

Clustering

- Clustering or cluster analysis is a machine learning technique, which groups the unlabelled 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."***

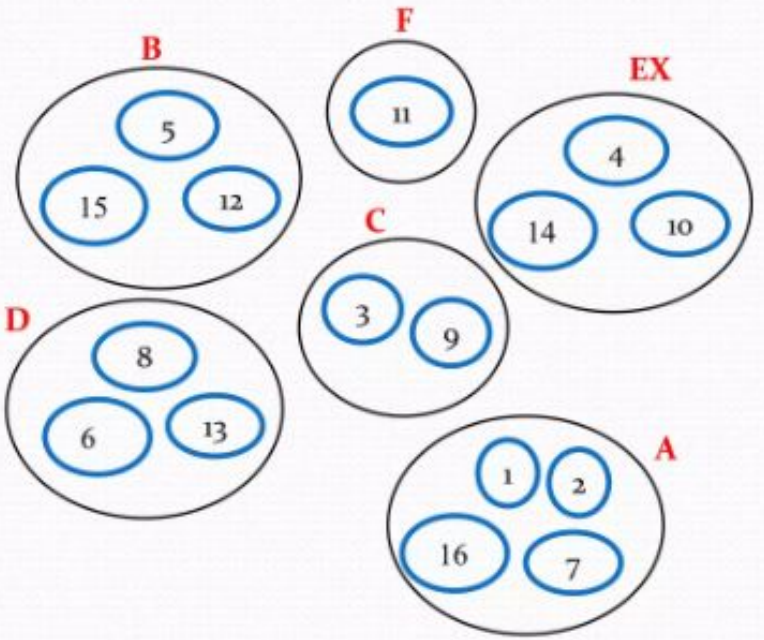
Clustering

For example, if we have customer purchase data, clustering can group customers with similar shopping habits. These clusters can then be used for targeted marketing, personalized recommendations or customer segmentation.

Table 12.1: Tabulation of Marks

Roll No	Mark	Grade
1	80	A
2	70	A
3	55	C
4	91	EX
5	65	B
6	35	D
7	76	A
8	40	D
9	50	C
10	85	EX
11	25	F
12	60	B
13	45	D
14	95	EX
15	63	B
16	88	A

Figure 12.1: Group representation of dataset in Table 15.1



Similarity Measures

Similarity between the objects is a numerical measure of the degree to which the objects are alike.

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Hierarchical Clustering

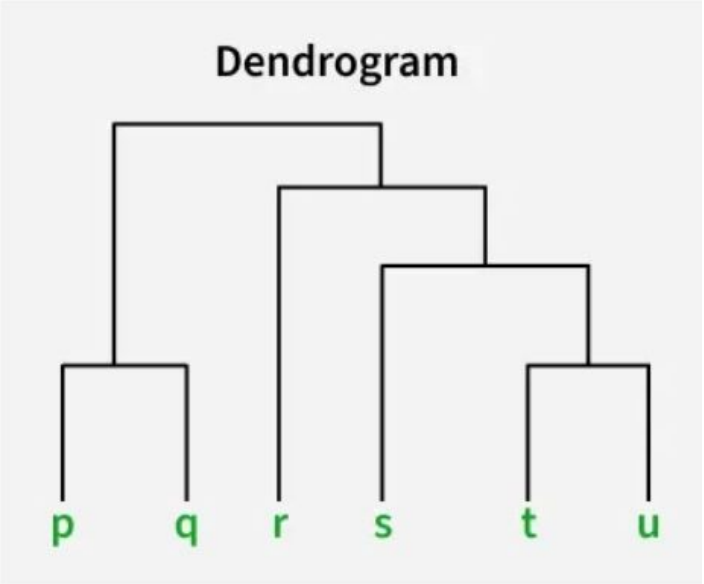
- Hierarchical Clustering is an unsupervised learning technique that builds a hierarchy of clusters by either merging or splitting them.
- This approach can be categorized into two types: Agglomerative (Bottom-up) and Divisive (Top-down).
- The algorithm builds clusters step by step either by progressively merging smaller clusters or by splitting a large cluster into smaller ones.
- The process is often visualized using a dendrogram, which helps to understand data similarity.

Imagine we have four fruits with different weights: an apple (100g), a banana (120g), a cherry (50g) and a grape (30g). Hierarchical clustering starts by treating each fruit as its own group.

- Start with each fruit as its own cluster.
- Merge the closest items: grape (30g) and cherry (50g) are grouped first.
- Next, apple (100g) and banana (120g) are grouped.
- Finally, these two clusters merge into one.
- Finally all the fruits are merged into one large group, showing how hierarchical clustering progressively combines the most similar data points.

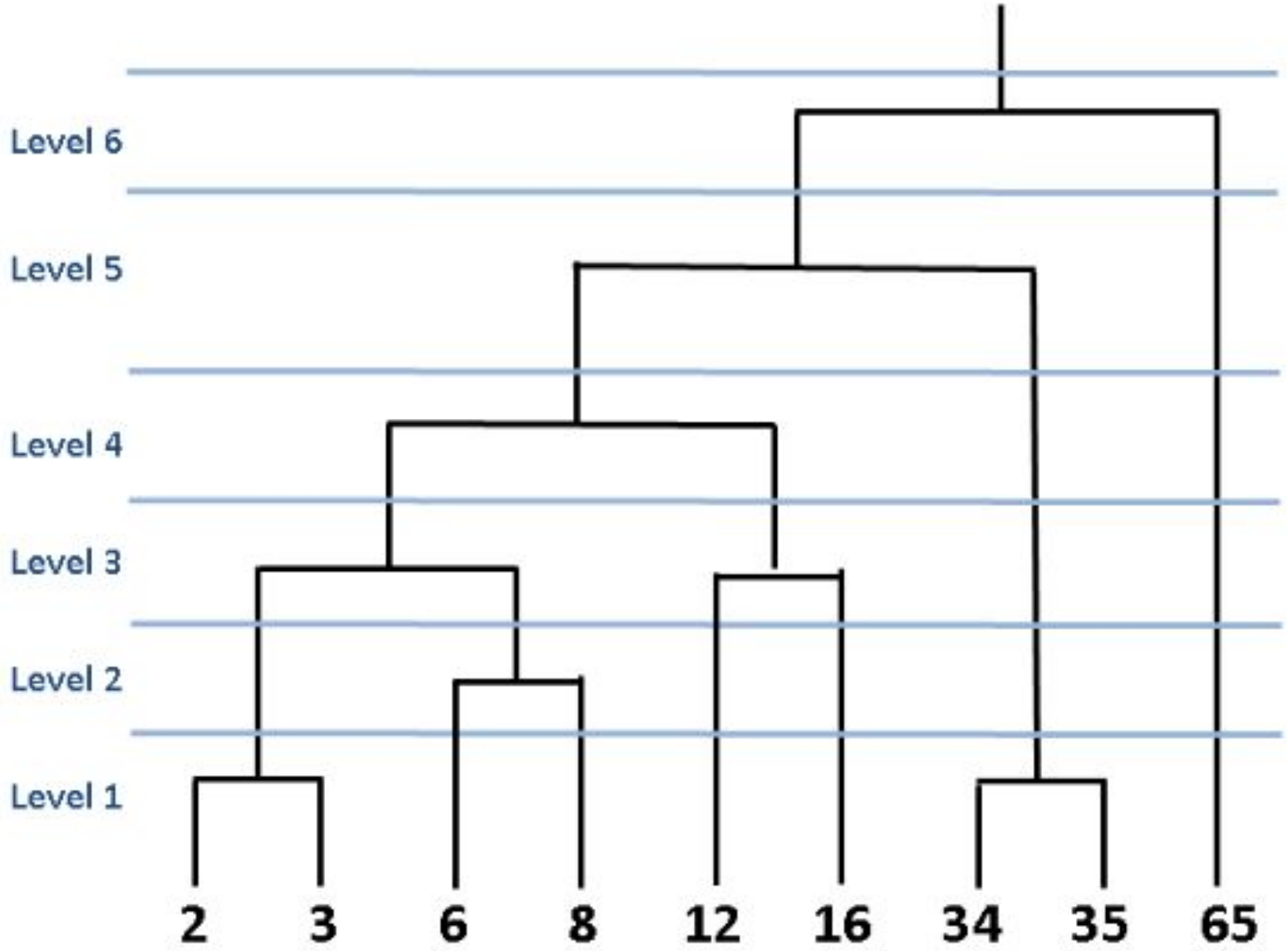
Dendrogram

- A dendrogram is like a family tree for clusters.
- It shows how individual data points or groups of data merge together.
- The bottom shows each data point as its own group and as we move up, similar groups are combined.
- The lower the merge point, the more similar the groups are. It helps us see how things are grouped step by step.



Dendrogram

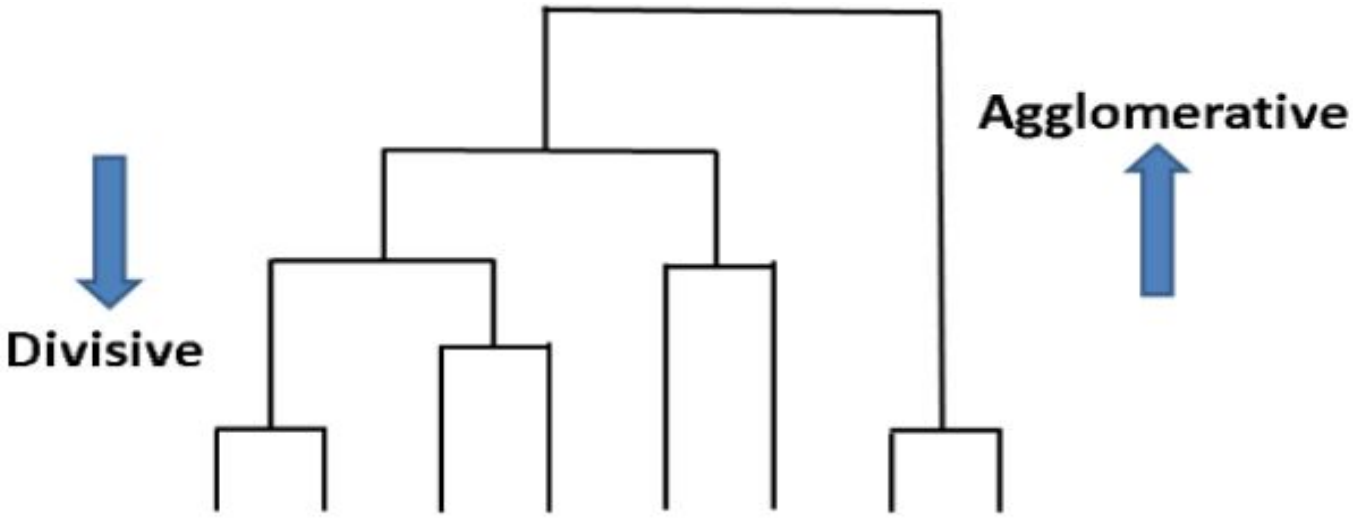
- At the bottom of the dendrogram the points P, Q, R, S and T are all separate.
- As we move up, the closest points are merged into a single group.
- The lines connecting the points show how they are progressively merged based on similarity.
- The height at which they are connected shows how similar the points are to each other; the shorter the line the more similar they are


Level 6
Level 5
Level 4
Level 3
Level 2
Level 1
2
3
6
8
12
16
34
35
65

Types of Hierarchical Clustering

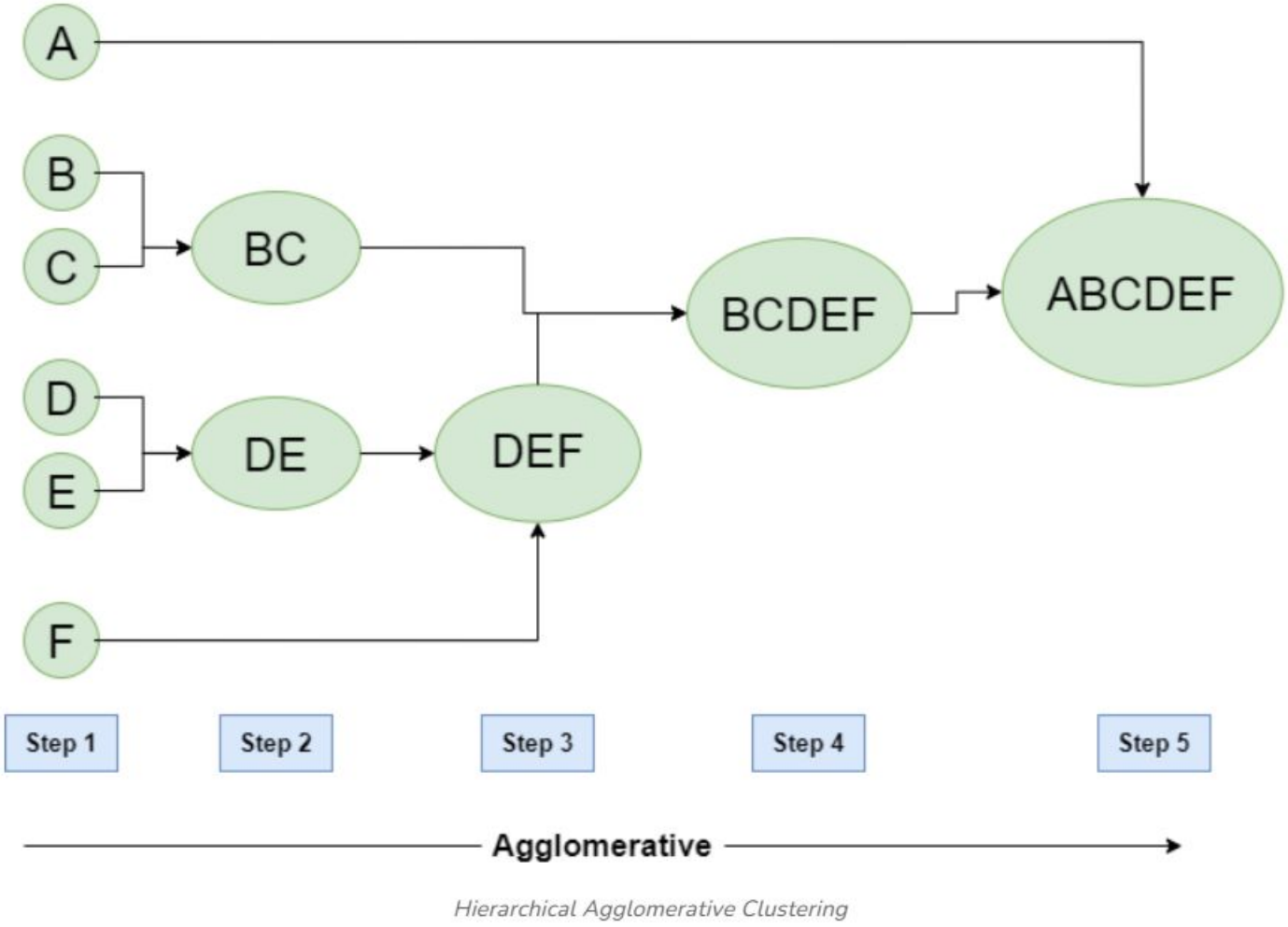
There are two main types of hierarchical clustering.

- Agglomerative Clustering
- Divisive clustering



Hierarchical Agglomerative Clustering

- It is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC).
- Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerate pairs of clusters until all clusters have been merged into a single cluster that contains all data.



Hierarchical Agglomerative Clustering

Workflow for Hierarchical Agglomerative clustering

- **Start with individual points:** Each data point is its own cluster. For example if we have 5 data points we start with 5 clusters each containing just one data point.
- **Calculate distances between clusters**: Calculate the distance between every pair of clusters. Initially since each cluster has one point this is the distance between the two data points.
- **Merge the closest clusters**: Identify the two clusters with the smallest distance and merge them into a single cluster.
- **Update distance matrix**: After merging we now have one less cluster. Recalculate the distances between the new cluster and the remaining clusters.
- **Repeat steps 3 and 4**: Keep merging the closest clusters and updating the distance matrix until we have only one cluster left.
- **Create a dendrogram**: As the process continues we can visualize the merging of clusters using a tree-like diagram called a dendrogram. It shows the hierarchy of how clusters are merged.

Linkage Methods

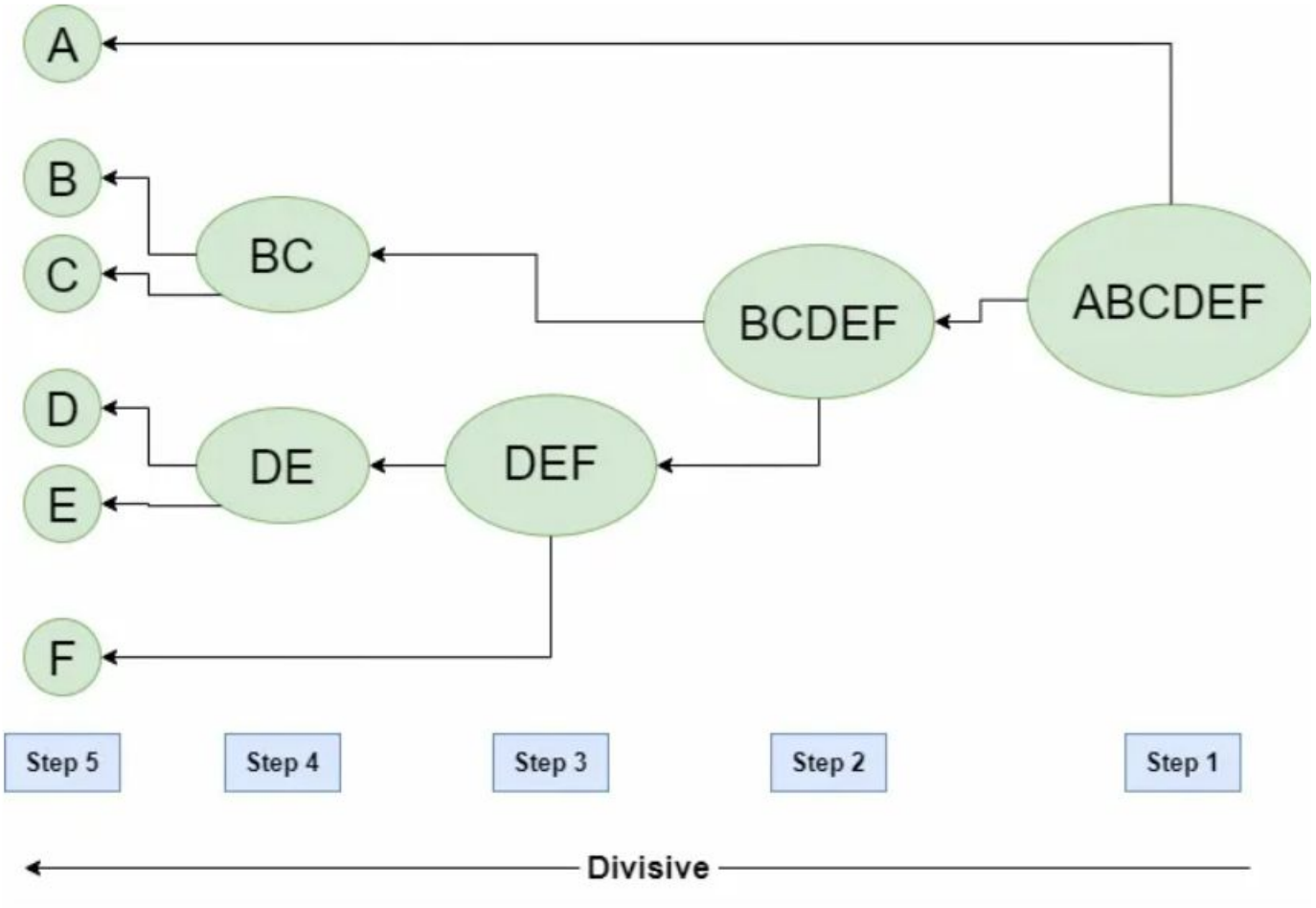
- **Single linkage**: Distance between the **closest** pair of points (one from each cluster).
- **Complete linkage**: Distance between the **farthest** pair of points.
- **Average linkage**: Average distance between all pairs of points (across clusters).

Hierarchical Divisive clustering

- Divisive clustering is also known as a top-down approach.
- Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been split into singleton clusters.

Workflow for Hierarchical Divisive clustering

- **Start with all data points in one cluster:** Treat the entire dataset as a single large cluster.
- **Split the cluster**: Divide the cluster into two smaller clusters. The division is typically done by finding the two most dissimilar points in the cluster and using them to separate the data into two parts.
- **Repeat the process**: For each of the new clusters, repeat the splitting process: Choose the cluster with the most dissimilar points and split it again into two smaller clusters.
- **Stop when each data point is in its own cluster**: Continue this process until every data point is its own cluster or the stopping condition (such as a predefined number of clusters) is met.



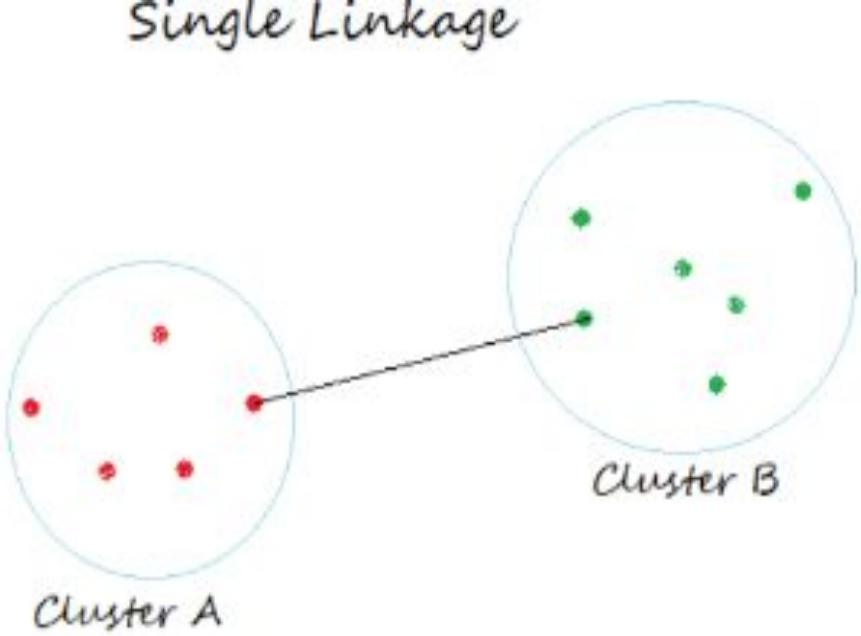
Divisive Clustering

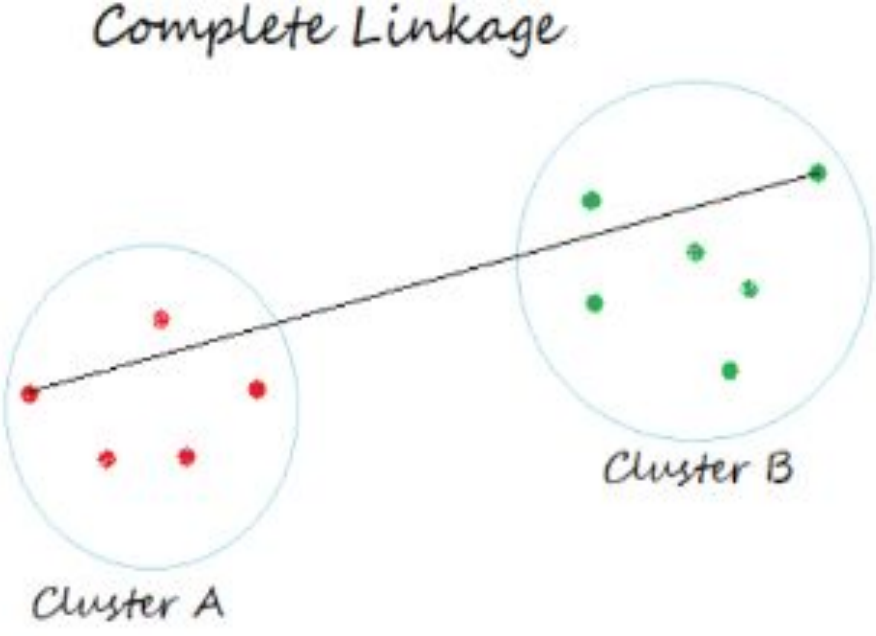
- In **Single Linkage**, the distance between two clusters is the minimum distance between members of the two clusters

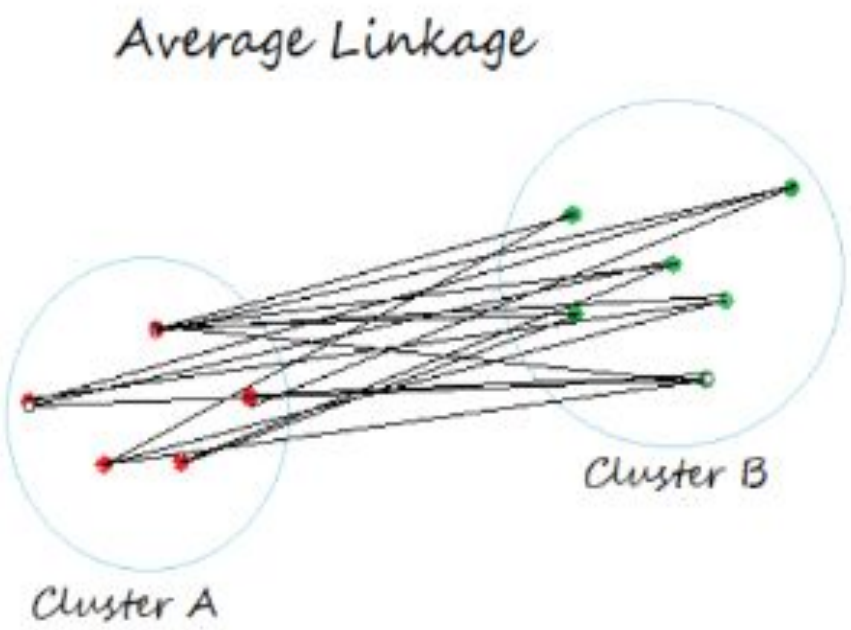
- In **Complete Linkage**, the distance between two clusters is the maximum distance between members of the two clusters

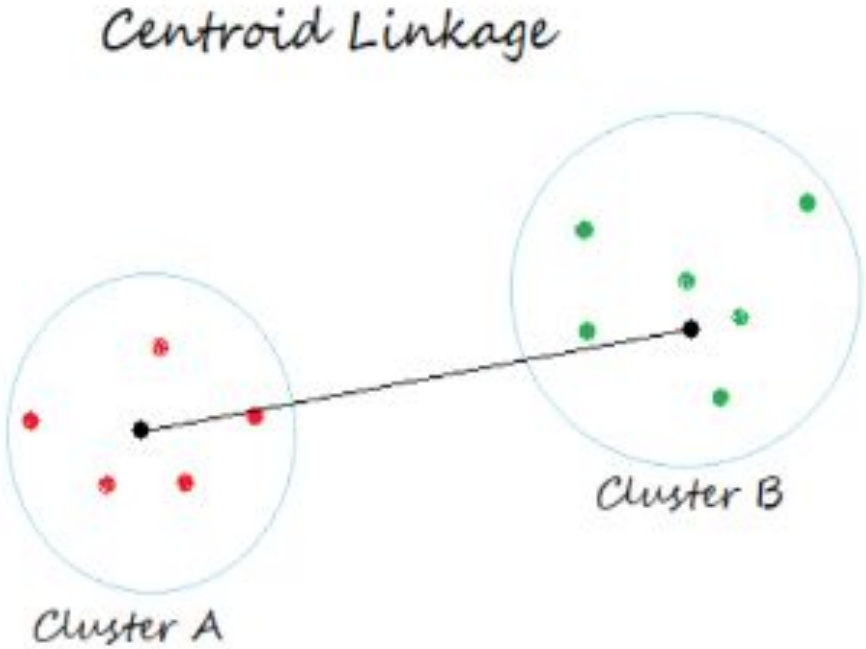
- In **Average Linkage**, the distance between two clusters is the average of all distances between members of the two clusters

- In **Centroid Linkage**, the distance between two clusters is is the distance between their centroids


Single Linkage
Cluster B
Cluster A


Complete Linkage
Cluster B
Cluster A


Average Linkage
Cluster B
Cluster A


Centroid Linkage
Cluster B
Cluster A

K-Means Algorithm

- K-Means Clustering is an unsupervised machine learning algorithm that helps group data points into clusters based on their inherent similarity.
- Unlike supervised learning, where we train models using labeled data, K-Means is used when we have data that is not labeled and the goal is to uncover hidden patterns or structures.

Working of K-Means Clustering

- Suppose 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. “k" represents the number of groups or clusters we want to classify our items into
- The algorithm will categorize the items into “k" groups or clusters of similarity. To calculate that similarity we will use the Euclidean distance as a measurement.

The algorithm works as follows:

- **Initialization:** We begin by randomly selecting k cluster centroids.
- **Assignment Step:** Each data point is assigned to the nearest centroid, forming clusters.
- **Update Step:** After the assignment, we recalculate the centroid of each cluster by averaging the points within it.
- **Repeat:** This process repeats until the centroids no longer change or the maximum number of iterations is reached.
- The goal is to partition the dataset into k clusters such that data points within each cluster are more similar to each other than to those in other clusters.

K-Means Clustering – Solved Example

- Suppose that the data mining task is to cluster points into three clusters,
- where the points are
- $A1(2, 10)$, $A2(2, 5)$, $A3(8, 4)$, $B1(5, 8)$, $B2(7, 5)$, $B3(6, 4)$, $C1(1, 2)$, $C2(4, 9)$.
- The distance function is Euclidean distance.
- Suppose initially we assign $A1$, $B1$, and $C1$ as the center of each cluster, respectively.

Current Centroids:
A1: (2, 10)
B1: (6, 6)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			2	10	6	6	1.5	1.5		
A1	2	10	0.00		5.66		6.52		1	1
A2	2	5	5.00		4.12		1.58		3	3
A3	8	4	8.49		2.83		6.52		2	2
B1	5	8	3.61		2.24		5.70		2	2
B2	7	5	7.07		1.41		5.70		2	2
B3	6	4	7.21		2.00		4.53		2	2
C1	1	2	8.06		6.40		1.58		3	3
C2	4	9	2.24		3.61		6.04		2 →	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Current Centroids:
A1: (3, 9.5)
B1: (6.5, 5.25)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			3	9.5	6.5	5.25	1.5	3.5		
A1	2	10	1.12		6.54		6.52		1	1
A2	2	5	4.61		4.51		1.58		3	3
A3	8	4	7.43		1.95		6.52		2	2
B1	5	8	2.50		3.13		5.70		2	1
B2	7	5	6.02		0.56		5.70		2	2
B3	6	4	6.26		1.35		4.53		2	2
C1	1	2	7.76		6.39		1.58		3	3
C2	4	9	1.12		4.51		6.04		1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Current Centroids:
A1: (3.67, 9)
B1: (7, 4.33)
C1: (1.5, 3.5)

Data Points			Distance to						Cluster	New Cluster
			3.67	9	7	4.33	1.5	3.5		
A1	2	10	1.94		7.56		6.52		1	1
A2	2	5	4.33		5.04		1.58		3	3
A3	8	4	6.62		1.05		6.52		2	2
B1	5	8	1.67		4.18		5.70		1	1
B2	7	5	5.21		0.67		5.70		2	2
B3	6	4	5.52		1.05		4.53		2	2
C1	1	2	7.49		6.44		1.58		3	3
C2	4	9	0.33		5.55		6.04		1	1

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Consider 10 points A1(2,5), A2(2,3), A3(4,5), A4(4,2), A5(6,5), A6(8,9), A7(2,1), A8(4,3), A9(1,6) and A10(3,7). Find 3 clusters after 2 epochs considering A1, A4 and A6 as the initial cluster centres. Use Euclidean distance as the distance function.

Thank You