
UNIT 11 –UNSUPERVISED LEARNING MODELS

11.0 Introduction

11.1 Expected Learning Outcomes

11.2 Unsupervised Learning Models

11.2.1 K-Means Clustering

11.2.2 Hierarchical Clustering

11.2.3 Agglomerative Clustering

11.2.4 Association Rules

11.3 Summary

11.4 Answers

11.5 References/Further Readings

11.0 INTRODUCTION

In the previous units, the focus was on supervised learning techniques where models are trained using labeled data to predict outcomes. However, in many real-world scenarios, labeled data is not available, and the objective is to discover hidden patterns, structures, or relationships within the data itself. This is where **unsupervised learning** plays a crucial role. Unsupervised learning models are designed to analyse and interpret datasets without predefined outputs, enabling the extraction of meaningful insights from raw and unstructured data.

Unsupervised learning is widely used in areas such as customer segmentation, market basket analysis, anomaly detection, and pattern recognition. Unlike supervised learning, where the model learns from known outcomes, unsupervised learning explores the inherent structure of the data. It groups similar data points together, identifies associations among variables, and reveals underlying distributions. This makes it an essential tool in data analysis, especially when dealing with large volumes of unlabelled data.

This unit introduces the fundamental concepts and techniques of unsupervised learning models for data analysis. It begins with an overview of different unsupervised learning approaches and then focuses on clustering techniques such as **K-Means Clustering**, **Hierarchical Clustering**, and **Agglomerative Clustering**, which are used to group similar data points into clusters based on their characteristics. These methods help in identifying natural groupings within the data and are widely applied in business and scientific domains.

In addition to clustering, the unit also covers **Association Rule Learning**, which is used to discover interesting relationships and patterns among variables in large datasets. This technique is particularly useful in applications like market basket analysis, where it helps in identifying combinations of items frequently purchased together.

Overall, this unit aims to build a strong conceptual understanding of unsupervised learning models and their applications in data analysis. By the end of this unit, learners will be able to

understand how patterns are discovered without labelled data and how different unsupervised techniques can be applied to solve real-world problems.

11.1 EXPECTED LEARNING OUTCOMES

After completing this unit, you will be able to:

- understand the concept and significance of unsupervised learning in data analysis.
- explore different unsupervised learning models and their applications.
- learn the working and implementation of K-Means Clustering technique.
- understand Hierarchical and Agglomerative Clustering methods for grouping data.
- analyse patterns and relationships using Association Rule Learning.
- develop the ability to apply clustering and association techniques to real-world datasets.

11.2 UNSUPERVISED LEARNING MODELS

Unsupervised learning models play a crucial role in data analysis by enabling the discovery of hidden patterns, structures, and relationships within data that does not contain predefined labels. Unlike supervised learning, where the outcome is known in advance, unsupervised learning focuses on exploring the inherent organization of the data. This makes it particularly useful in real-world scenarios such as customer segmentation, pattern recognition, and market analysis, where labelled data is often unavailable or difficult to obtain.

One of the most widely used unsupervised learning techniques is **K-Means Clustering**, which aims to partition the dataset into a predefined number of clusters. It works by assigning data points to clusters in such a way that the distance between points within the same cluster is minimized, while the distance between different clusters is maximized. This method is simple, efficient, and highly effective for large datasets, making it a popular choice for grouping similar data points based on their characteristics.

Another important approach is **Hierarchical Clustering**, which builds a hierarchy of clusters rather than forming them in a single step. It organizes data into a tree-like structure called a dendrogram, allowing us to visualize how clusters are formed at different levels of similarity. This method does not require the number of clusters to be specified in advance, providing flexibility in understanding the data structure.

A specific type of hierarchical clustering is **Agglomerative Clustering**, which follows a bottom-up approach. In this method, each data point initially forms its own cluster, and clusters are gradually merged based on similarity until a single cluster is formed or a stopping condition is met. This approach is intuitive and useful for understanding how smaller groups combine to form larger clusters.

In addition to clustering techniques, **Association Rule Learning** is another important unsupervised method that focuses on discovering relationships between variables in large datasets. It identifies patterns such as items that frequently occur together, making it highly valuable in applications like market basket analysis. For example, it can reveal that customers who purchase one product are likely to purchase another, helping businesses make informed decisions.

Together, these unsupervised learning models provide powerful tools for analyzing data without labelled outcomes. While K-Means focuses on partitioning data into distinct groups, hierarchical and agglomerative clustering offer a more structured view of relationships among data points, and association rules uncover meaningful connections between variables. These techniques form the foundation for deeper exploration, and each of them will be studied in detail in the subsequent sections of this unit.

☛ Check Your Progress 1

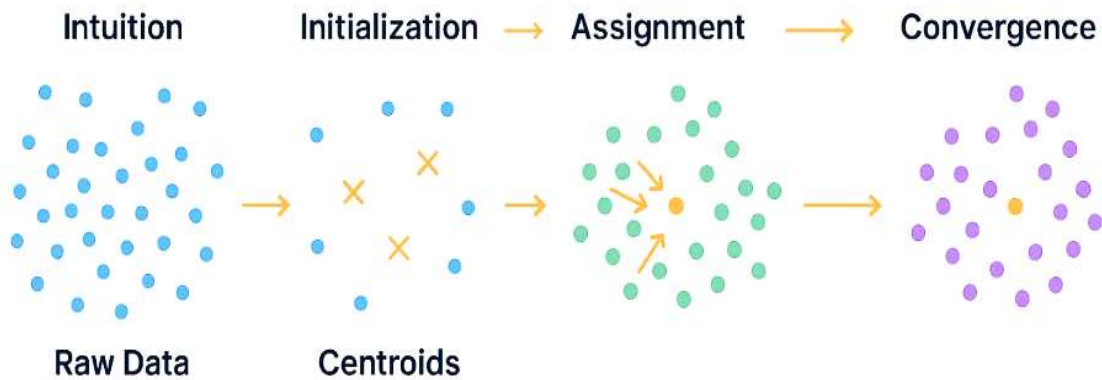
- Q1. What is Unsupervised Learning? How does it differ from supervised learning?
.....
.....
- Q2. List the applications of Unsupervised Learning in real-world scenarios.
.....
.....
- Q3. What are Unsupervised Learning Models? Explain their significance in data analysis.
.....
.....
- Q4. Briefly explain different types of Unsupervised Learning techniques.
.....
.....

11.2.1 K-MEANS CLUSTERING

K-Means Clustering is one of the most widely used unsupervised learning algorithms for grouping data into clusters based on similarity. The main objective of K-Means is to partition a dataset into **K distinct clusters** such that data points within the same cluster are more similar to each other than to those in other clusters. Since it does not rely on labeled data, K-Means helps in discovering hidden patterns and structures within the dataset.

The algorithm works iteratively by assigning data points to clusters and updating the cluster centers, known as **centroids**, until convergence is achieved. Initially, K cluster centers are selected randomly. Each data point is then assigned to the nearest centroid based on a distance measure, typically the Euclidean distance. After assigning all points, new centroids are calculated as the mean of all points in each cluster. This process repeats until the centroids no longer change significantly.

K-MEANS CLUSTERING



The figure illustrates the working of the K-Means clustering algorithm in a step-by-step visual manner, helping to understand how clusters are formed iteratively. In the initial stage, a set of data points is scattered in the feature space, and a predefined number of cluster centers (centroids) are selected randomly. These initial centroids may not represent the actual grouping of data, but they serve as the starting point for the algorithm.

In the next stage, known as the **assignment step**, each data point is assigned to the nearest centroid based on a distance measure, typically the Euclidean distance. This results in the formation of preliminary clusters, where points closer to a particular centroid are grouped together. At this stage, the clusters may still be uneven or inaccurate because the centroids were chosen randomly.

Following this, the **update step** takes place, where the centroids are recalculated. Each centroid is repositioned to the mean (average) of all the data points assigned to its cluster. This shift in centroid location reflects a better representation of the cluster's center. As a result, the cluster boundaries begin to change.

The algorithm then repeats the assignment and update steps iteratively. With each iteration, data points may switch clusters as centroids move to more optimal positions. Gradually, the centroids stabilize, meaning their positions no longer change significantly. At this point, the clusters become fixed, and the algorithm converges to a final solution.

Overall, the figure demonstrates that K-Means clustering is an **iterative optimization process** where clusters are continuously refined by minimizing the distance between data points and their respective centroids. It visually emphasizes how initial random groupings evolve into well-defined clusters through repeated reassignment and centroid updates, making the underlying concept of the algorithm easy to understand.

Mathematically, the objective of K-Means is to minimize the following function:

$$J = \sum_{i=1}^K \sum_{x \in C_i} \|x - \mu_i\|^2$$

where K is the number of clusters, C_i represents the i^{th} cluster, x is a data point, and μ_i is the centroid of cluster C_i . The term $\|x - \mu_i\|^2$ represents the squared Euclidean distance between a point and its cluster centroid. The algorithm aims to minimize this total within-cluster variance.

To understand this concept intuitively, consider a real-life example of customer segmentation in a shopping mall. Customers can be grouped based on their spending behaviour and visit frequency. K-Means will automatically group customers into clusters such as high spenders, moderate spenders, and low spenders. This helps businesses design targeted marketing strategies.

Let us now understand K-Means with a numerical example.

Consider the following data points:

$$(2,2), (4,4), (6,6), (8,8)$$

Let the number of clusters be $K = 2$.

Step 1: Initialize Centroids

Assume initial centroids:

$$\mu_1 = (2,2), \quad \mu_2 = (8,8)$$

Step 2: Assign Points to Nearest Centroid

Distance formula:

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

Now calculate distances:

For point (4,4):

Distance to μ_1 :

$$\sqrt{(4-2)^2 + (4-2)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

Distance to μ_2 :

$$\sqrt{(4-8)^2 + (4-8)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$$

So, (4,4) belongs to Cluster 1.

For point (6,6):

Distance to μ_1 :

$$\sqrt{(6-2)^2 + (6-2)^2} = \sqrt{16+16} = \sqrt{32} \approx 5.66$$

Distance to μ_2 :

$$\sqrt{(6-8)^2 + (6-8)^2} = \sqrt{4+4} = \sqrt{8} \approx 2.83$$

So, (6,6) belongs to Cluster 2.

Step 3: Form Clusters

Cluster 1: (2,2), (4,4)

Cluster 2: (6,6), (8,8)

Step 4: Recalculate Centroids

$$\mu_1 = \left(\frac{2+4}{2}, \frac{2+4}{2} \right) = (3,3)$$

$$\mu_2 = \left(\frac{6+8}{2}, \frac{6+8}{2} \right) = (7,7)$$

Step 5: Reassign Points

Now check again—assignments remain same.

So, algorithm converges.

Final Clusters are Cluster 1: (2,2), (4,4) and Cluster 2: (6,6), (8,8) Thus, the Final centroids are: (3,3) and (7,7)

The algorithm successfully divided the dataset into two clusters based on proximity. Points closer to (3,3) belong to one group, and points closer to (7,7) belong to another. This demonstrates how K-Means groups similar data points together. In real-world scenarios, such clustering helps in identifying patterns such as grouping customers, documents, or products based on similarity.

Let's understand the working of K-Means algorithm with one more Numerical

Example 1: Apply the K-Means Clustering algorithm to group the following data points into 2 clusters. The given data points are: A(1,1), B(2,1), C(4,3), D(5,4). Assume the initial centroids are: $C_1 = (1,1)$, $C_2 = (5,4)$.

Using the Euclidean distance measure, perform the clustering step by step until the centroids do not change. Also write the final clusters and interpret the result.

Solution:

In this problem, we have to divide the given data points into **K = 2 clusters** using the K-Means algorithm.

The algorithm works by:

- choosing initial centroids,
- assigning each point to the nearest centroid,
- recalculating the centroids,
- repeating the process until no change occurs.

Formula Used in K-Means Clustering is as follows:

(a) Euclidean Distance Formula

The distance between a data point (x_1, y_1) and a centroid (x_2, y_2) is:

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

(b) Centroid Formula

The centroid of a cluster is the mean of all points in that cluster.

$$\mu = \left(\frac{\sum x}{n}, \frac{\sum y}{n} \right)$$

Where,

- K : number of clusters
- C_1, C_2 : centroids of cluster 1 and cluster 2
- d : distance between a point and a centroid
- (x, y) : coordinates of a data point
- μ : new centroid of a cluster
- n : number of points in a cluster

Given Data:

- Data points: $A(1,1)$, $B(2,1)$, $C(4,3)$, $D(5,4)$
- Initial centroids: $C_1 = (1,1)$, $C_2 = (5,4)$

The stepwise solution is as follows:

Iteration 1: Assign Points to Nearest Centroid

We now calculate the distance of each point from both centroids.

Point A(1,1)

Distance from $C_1 = (1,1)$:

$$d(A, C_1) = \sqrt{(1-1)^2 + (1-1)^2} = \sqrt{0+0} = 0$$

Distance from $C_2 = (5,4)$:

$$d(A, C_2) = \sqrt{(1-5)^2 + (1-4)^2} = \sqrt{16+9} = \sqrt{25} = 5$$

Since $0 < 5$, point A belongs to **Cluster 1**.

Point B(2,1)

Distance from $C_1 = (1,1)$:

$$d(B, C_1) = \sqrt{(2-1)^2 + (1-1)^2} = \sqrt{1+0} = 1$$

Distance from $C_2 = (5,4)$:

$$d(B, C_2) = \sqrt{(2-5)^2 + (1-4)^2} = \sqrt{9+9} = \sqrt{18} \approx 4.24$$

Since $1 < 4.24$, point B belongs to **Cluster 1**.

Point C(4,3)

Distance from $C_1 = (1,1)$:

$$d(C, C_1) = \sqrt{(4-1)^2 + (3-1)^2} = \sqrt{9+4} = \sqrt{13} \approx 3.61$$

Distance from $C_2 = (5,4)$:

$$d(C, C_2) = \sqrt{(4-5)^2 + (3-4)^2} = \sqrt{1+1} = \sqrt{2} \approx 1.41$$

Since $1.41 < 3.61$, point C belongs to **Cluster 2**.

Point D(5,4)

Distance from $C_1 = (1,1)$:

$$d(D, C_1) = \sqrt{(5-1)^2 + (4-1)^2} = \sqrt{16+9} = \sqrt{25} = 5$$

Distance from $C_2 = (5,4)$:

$$d(D, C_2) = \sqrt{(5-5)^2 + (4-4)^2} = 0$$

Since $0 < 5$, point D belongs to **Cluster 2**.

Clusters After Iteration 1

So the clusters formed are:

Cluster 1: A(1,1), B(2,1)

Cluster 2: C(4,3), D(5,4)

Recalculate New Centroids

Now we compute the new centroid of each cluster.

New Centroid of Cluster 1

Points in Cluster 1: (1,1), (2,1)

Using centroid formula:

$$C'_1 = \left(\frac{1+2}{2}, \frac{1+1}{2} \right)$$

$$C'_1 = \left(\frac{3}{2}, \frac{2}{2} \right) = (1.5, 1)$$

New Centroid of Cluster 2

Points in Cluster 2: (4,3), (5,4)

$$C'_2 = \left(\frac{4+5}{2}, \frac{3+4}{2} \right)$$

$$C'_2 = \left(\frac{9}{2}, \frac{7}{2} \right) = (4.5, 3.5)$$

Iteration 2: Reassign Points Using New Centroids

Now use the new centroids:

$$C_1 = (1.5, 1), C_2 = (4.5, 3.5)$$

Point A(1,1)

$$d(A, C_1) = \sqrt{(1-1.5)^2 + (1-1)^2} = \sqrt{0.25} = 0.5$$

$$d(A, C_2) = \sqrt{(1-4.5)^2 + (1-3.5)^2} = \sqrt{12.25 + 6.25} = \sqrt{18.5} \approx 4.30$$

So A remains in **Cluster 1**.

Point B(2,1)

$$d(B, C_1) = \sqrt{(2-1.5)^2 + (1-1)^2} = \sqrt{0.25} = 0.5$$

$$d(B, C_2) = \sqrt{(2-4.5)^2 + (1-3.5)^2} = \sqrt{6.25 + 6.25} = \sqrt{12.5} \approx 3.54$$

So B remains in **Cluster 1**.

Point C(4,3)

$$d(C, C_1) = \sqrt{(4 - 1.5)^2 + (3 - 1)^2} = \sqrt{6.25 + 4} = \sqrt{10.25} \approx 3.20$$

$$d(C, C_2) = \sqrt{(4 - 4.5)^2 + (3 - 3.5)^2} = \sqrt{0.25 + 0.25} = \sqrt{0.5} \approx 0.71$$

So C remains in **Cluster 2**.

Point D(5,4)

$$d(D, C_1) = \sqrt{(5 - 1.5)^2 + (4 - 1)^2} = \sqrt{12.25 + 9} = \sqrt{21.25} \approx 4.61$$

$$d(D, C_2) = \sqrt{(5 - 4.5)^2 + (4 - 3.5)^2} = \sqrt{0.25 + 0.25} = \sqrt{0.5} \approx 0.71$$

So D remains in **Cluster 2**.

Check for Convergence: After the second iteration, the cluster assignments did not change. Therefore, the algorithm has **converged**.

So, the Final Answer is:

Final Cluster 1: A(1,1), B(2,1)

Final Cluster 2: C(4,3), D(5,4)

and

Final Centroids: $C_1 = (1.5, 1)$, $C_2 = (4.5, 3.5)$

Results of the above numerical showcased that the K-Means algorithm successfully divided the given four points into two meaningful clusters based on their distance from the centroids. Points A(1,1) and B(2,1) are closer to each other, so they form one cluster. Similarly, points C(4,3) and D(5,4) are grouped into another cluster because they lie close together in the feature space.

This result shows that K-Means works by identifying natural groupings in the data. In real-life applications, such grouping can be useful for customer segmentation, document clustering, image compression, and pattern recognition. The final centroids represent the center of each cluster and summarize the overall position of points in that group.

To solve a K-Means numerical, always follow this sequence:

- First, note the value of K , the number of clusters.
- Second, write the initial centroids.
- Third, calculate the distance of each point from every centroid.
- Fourth, assign each point to the nearest centroid.
- Fifth, recompute the centroids by taking the mean of all points in each cluster.
- Sixth, repeat the process until the assignments no longer change.

So the complete flow is:

Initial Centroids → Distance Calculation → Cluster Assignment → Centroid Update
→ Repeat

This repeated reassignment and updating is the core idea of K-Means clustering.

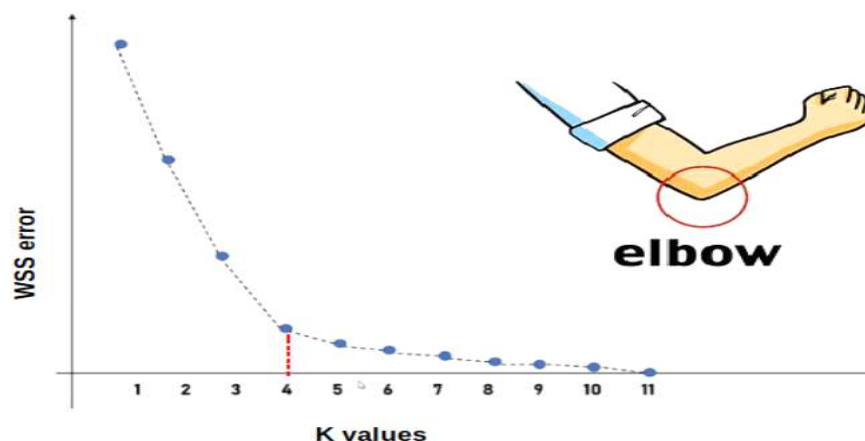
Before concluding this section, we are going to discuss an important method called “Elbow Method” The brief description of this method is as follows:

Elbow Method in K-Means Clustering: The Elbow Method is a widely used technique to determine the optimal number of clusters (K) in K-Means Clustering. Since K-Means requires the number of clusters to be specified beforehand, the elbow method helps in identifying the most suitable value of K by evaluating how well the data fits into different cluster groupings. The method is based on minimizing the Within-Cluster Sum of Squares (WCSS), which measures the total squared distance between data points and their respective cluster centroids. Mathematically, WCSS is expressed as:

$$WCSS = \sum_{i=1}^K \sum_{x \in C_i} \|x - \mu_i\|^2$$

As the number of clusters increases, the value of WCSS decreases because data points are assigned to clusters whose centroids are closer to them. Initially, this decrease is rapid, as adding more clusters significantly improves the grouping of data. However, after a certain point, the reduction in WCSS becomes slower, indicating that adding more clusters does not contribute much to improving the model. This point of diminishing returns is referred to as the “Elbow point.”

Elbow method



In the graph:

- The **X-axis** represents the number of clusters K .
- The **Y-axis** represents the WCSS (error).
- As K increases, WCSS decreases sharply at first.

- After a certain value of K , the decrease becomes gradual.

The point where the curve bends like an “elbow” indicates the best value of K .

In a typical elbow method graph, the number of clusters (K) is plotted on the X-axis, and the WCSS value is plotted on the Y-axis. The curve shows a sharp decline initially and then gradually levels off. The point where the curve bends or forms an “elbow” represents the optimal number of clusters. This is because it balances the trade-off between minimizing error and avoiding unnecessary complexity.

Example 2: To understand this with a numerical example, consider a dataset where the WCSS values are calculated for different values of K .

K (Number of Clusters)	WCSS
1	500
2	300
3	200
4	180
5	170

Solution: From the above table it can be analysed that:

- From $K = 1$ to $K = 2$, WCSS decreases significantly ($500 \rightarrow 300$).
- From $K = 2$ to $K = 3$, there is still a large reduction ($300 \rightarrow 200$).
- From $K = 3$ onwards, the decrease becomes small ($200 \rightarrow 180 \rightarrow 170$).

Thus, the major improvement stops after $K = 3$. So, Optimal number of clusters $K = 3$

Table shown in the above example showcased that for $K = 1$, WCSS = 500; for $K = 2$, WCSS = 300; for $K = 3$, WCSS = 200; for $K = 4$, WCSS = 180; and for $K = 5$, WCSS = 170. It can be observed that there is a significant reduction in WCSS from $K = 1$ to $K = 3$, but after $K = 3$, the decrease becomes minimal.

Therefore, the optimal number of clusters is $K = 3$, as it provides a good balance between accuracy and simplicity.

From an interpretational perspective, the elbow point indicates the division of the data into three clusters which captures most of the inherent structure in the dataset. Increasing the number of clusters beyond this point leads to only marginal improvements while increasing computational complexity. Thus, the elbow method helps prevent both underfitting (too few clusters) and overfitting (too many clusters).

Thus, the elbow method can be understood as a practical way to decide how many groups should be formed in the data. It emphasizes finding a balance where the clusters are meaningful without making the model unnecessarily complex. Overall, the elbow method is a

simple yet effective tool for selecting the optimal value of K in K-Means clustering and plays a crucial role in improving clustering performance.

The advantages and Limitations of K-Means algorithm are as follows:

Advantages of K-Means Clustering

- K-Means is simple and easy to implement.
- It is computationally efficient and works well with large datasets.
- The algorithm converges quickly and is easy to understand.
- It is widely used in practical applications such as customer segmentation and image compression.

Limitations of K-Means Clustering

- K-Means requires the number of clusters (K) to be specified in advance, which may not always be known.
- It is sensitive to the initial selection of centroids, which can affect the final result.
- It does not perform well with clusters of different shapes and sizes.
- It is also sensitive to outliers and noise in the data.

We learned that the K-Means can be understood as a process of repeatedly grouping and adjusting until the best cluster centers are found. The key idea is minimizing the distance between data points and their respective centroids. The algorithm alternates between assigning points to clusters and updating centroids, gradually improving the clustering quality. This iterative refinement makes K-Means a powerful yet simple technique for unsupervised learning.

Also, it is learned the K-Means Clustering is a foundational algorithm in unsupervised learning that helps in discovering natural groupings within data. By minimizing within-cluster variance and iteratively updating centroids, it provides a practical and efficient way to analyze unlabeled datasets. Despite its limitations, its simplicity and effectiveness make it one of the most popular clustering techniques in machine learning.

☛ Check Your Progress 2

Q1. What is K-Means Clustering? Explain its objective.

.....
.....

Q2. Explain the working steps of the K-Means algorithm

.....
.....

Q3. What is the role of centroid in K-Means Clustering?.

.....
.....

Q4. Explain the Elbow Method for selecting optimal number of clusters (K)..

11.2.2 HIERARCHICAL CLUSTERING

Hierarchical Clustering is an important unsupervised learning algorithm used to group similar data points into clusters based on their similarity or distance. Unlike K-Means Clustering, where the number of clusters must be specified in advance, hierarchical clustering builds a **tree-like structure (called a dendrogram)** that represents the arrangement of clusters at different levels of similarity. This allows us to explore the data at multiple levels and choose the number of clusters based on interpretation.

Hierarchical clustering works in two main ways:

- **Agglomerative (bottom-up approach)** and
- **Divisive (top-down approach).**

In this unit, the focus is primarily on the agglomerative approach, where each data point initially forms its own cluster, and clusters are gradually merged step by step based on similarity until all points belong to a single cluster.

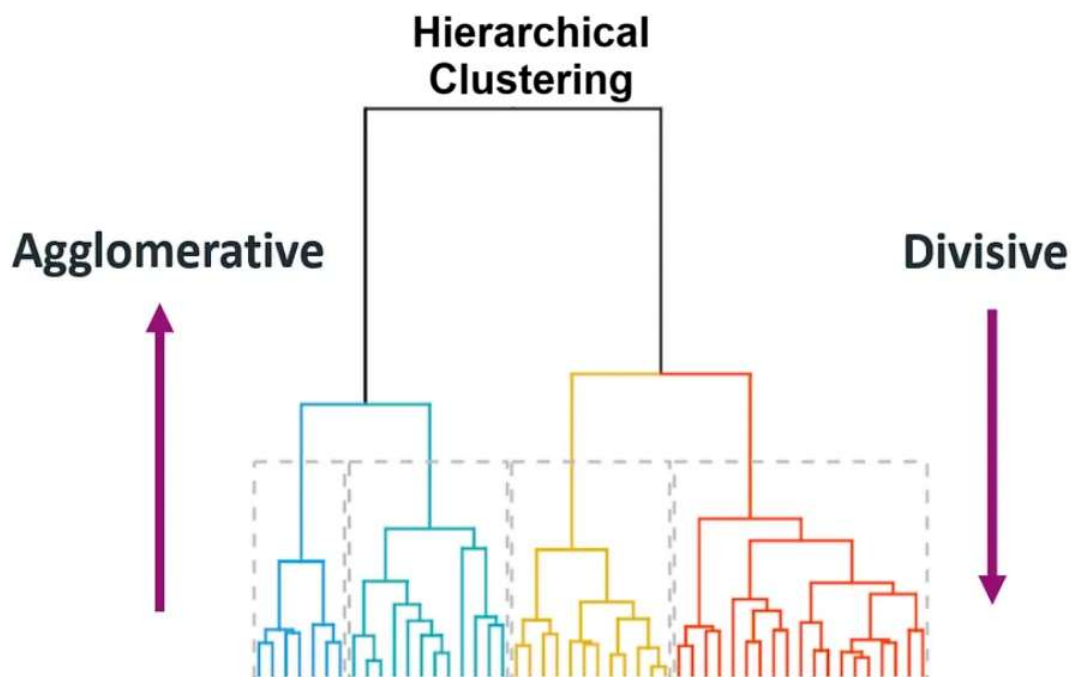


Figure: Dendrogram

The figure shows a **dendrogram**, which is a tree-like diagram used to visualize hierarchical clustering. Each leaf node represents a data point, and branches represent the merging of clusters. The height at which two clusters merge indicates the distance between them. By cutting the dendrogram at a certain level, we can obtain the desired number of clusters.

The agglomerative hierarchical clustering algorithm follows these steps:

- Initially, each data point is treated as a separate cluster.
- Then, the distance between all pairs of clusters is calculated.
- The two closest clusters are merged into one cluster.
- After merging, the distance matrix is updated, and the process repeats.
- This continues until all data points are grouped into a single cluster or until a stopping criterion is reached.

To determine how clusters are merged, hierarchical clustering uses distance measures and linkage criteria.

(a) Distance Measure (Euclidean Distance)

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

(b) Linkage Methods

- Single Linkage: distance between closest points of clusters
- Complete Linkage: distance between farthest points
- Average Linkage: average distance between all points
- Centroid Linkage: distance between cluster centroids

To understand this clearly, consider grouping students based on their marks in subjects. Initially, each student is treated as an individual cluster. Students with similar marks are grouped together step by step, forming larger clusters. Finally, we obtain groups such as high performers, average performers, and low performers. The dendrogram helps visualize how these groups are formed.

Example 3: Consider the following data points: $A(1,1)$, $B(2,1)$, $C(4,3)$; apply Hierarchical clustering and identify the clusters.

Solution:

Step 1: Compute Distance Matrix

Using Euclidean distance:

$$d(A, B) = \sqrt{(1 - 2)^2 + (1 - 1)^2} = 1$$

$$d(A, C) = \sqrt{(1 - 4)^2 + (1 - 3)^2} = \sqrt{9 + 4} = \sqrt{13} \approx 3.61$$

$$d(B, C) = \sqrt{(2 - 4)^2 + (1 - 3)^2} = \sqrt{4 + 4} = \sqrt{8} \approx 2.83$$

Step 2: Merge Closest Clusters

The smallest distance is:

$$d(A, B) = 1$$

So merge A and B:

$$\text{Cluster 1} = \{A, B\}$$

Step 3: Update Distances

Now calculate distance between cluster $\{A, B\}$ and point C using **single linkage**:

$$d(\{A, B\}, C) = \min(d(A, C), d(B, C))$$

$$d = \min(3.61, 2.83) = 2.83$$

Step 4: Final Merge

Merge $\{A, B\}$ and C . Thus, the Final Cluster = $\{A, B, C\}$

Final Clustering Structure

Step 1: A and B merged

Step 2: (A,B) merged with C

The algorithm first grouped A and B because they are closest to each other. Then, it added C to this cluster at a higher distance level. This shows that A and B are more similar compared to C. The dendrogram visually represents this relationship, where A and B merge earlier than C.

The advantages and Limitation of Hierarchical clustering are as follows:

Advantages of Hierarchical Clustering

- Hierarchical clustering does not require the number of clusters to be specified in advance.
- It provides a clear visual representation through dendrograms.
- It is flexible and can capture complex relationships between data points.
- It is useful for exploratory data analysis.

Limitations of Hierarchical Clustering

- The algorithm is computationally expensive for large datasets.
- Once a merge or split is done, it cannot be undone.
- It is sensitive to noise and outliers. The results may vary depending on the choice of linkage method.

So, Hierarchical clustering can be understood as a process of gradually grouping similar data points step by step. Instead of directly forming clusters, it builds a hierarchy of clusters that can be visualized and interpreted at different levels. The dendrogram helps in deciding how many clusters should be formed by cutting the tree at a suitable level.

This section concludes with the understanding that the Hierarchical clustering is a powerful technique for analysing data structure without predefined labels. By building a hierarchy of clusters and representing it through dendrograms, it provides deep insights into relationships among data points. Despite its computational limitations, it remains a valuable tool for clustering and exploratory analysis in machine learning.

In the subsequent section we extend our discussion to Agglomerative Clustering in more detail.

🔍 Check Your Progress 3

Q1. What is Hierarchical Clustering? Explain its key concept..

.....
.....

Q2. Differentiate between Agglomerative and Divisive approaches in Hierarchical Clustering.

.....
.....

Q3. What is a dendrogram and how is it used in clustering?

.....
.....

Q4. Explain different linkage methods used in Hierarchical Clustering.

.....
.....

11.2.3 AGGLOMERATIVE CLUSTERING

Agglomerative Clustering is a type of hierarchical clustering that follows a **bottom-up approach**. This means the algorithm starts with the smallest possible units and gradually builds larger clusters. Initially, each data point is considered as a separate cluster. Then, the algorithm looks for the two clusters that are closest to each other and merges them into a single cluster. This merging process continues step by step until all data points are combined into one large cluster. The entire process can be visualized using a **dendrogram**, which shows how clusters are formed and merged at different levels.

In this approach, we often use a method called **single linkage** to measure the distance between clusters. According to single linkage, the distance between two clusters is defined as the **minimum distance between any two points**, where one point belongs to the first cluster and the other belongs to the second cluster. In simple terms, we look for the closest pair of points between two clusters to decide how far apart those clusters are.

To calculate distances, we use a distance formula. Since the data points in this example are one-dimensional, the distance between two points x_i and x_j is simply the absolute difference between them:

$$d(x_i, x_j) = |x_i - x_j|$$

This formula gives us how far apart two individual data points are.

When dealing with clusters instead of individual points, we use the single linkage formula:

$$d(C_1, C_2) = \min \{d(x_i, x_j)\}$$

Here, C_1 and C_2 are two clusters, and x_i and x_j are points belonging to those clusters. The distance between the clusters is the smallest distance among all possible pairs of points taken from the two clusters.

In terms of variables, x_i and x_j represent individual data points, $d(x_i, x_j)$ represents the distance between two points, C_1 and C_2 represent clusters, and $d(C_1, C_2)$ represents the distance between those clusters. The term “min” simply means we select the smallest distance among all possible point pairs.

Overall, agglomerative clustering can be understood as a process of repeatedly merging the closest data points or clusters, using simple distance calculations, until a complete hierarchy of clusters is formed.

In the above section we learned that Agglomerative Clustering is one of the most commonly used approaches in hierarchical clustering, following a bottom-up strategy. It begins by treating each data point as an individual cluster and then progressively merges the closest clusters step by step until all data points are grouped into a single cluster or a stopping criterion is reached. This gradual merging process forms a hierarchy of clusters, which can be visualized using a dendrogram.

Also, we understood that Agglomerative clustering is essentially a type of hierarchical clustering, and in most practical scenarios, when we refer to hierarchical clustering, we are often referring to agglomerative clustering. The key idea is to build a hierarchy by combining smaller clusters into larger ones based on similarity or distance. This makes it intuitive and easy to understand, as it mimics natural grouping processes where similar items are combined to form broader categories.

We learned in our earlier section that the two i.e. Agglomerative and Hierarchical Clustering are closely related in the sense that the Hierarchical clustering is a broader concept that includes two main approaches:

- **Agglomerative Clustering (Bottom-Up)**
- **Divisive Clustering (Top-Down)**

Agglomerative clustering is the **most widely used form** of hierarchical clustering. In this method, the hierarchy is built from the bottom by merging clusters. On the other hand, hierarchical clustering as a concept includes both merging (agglomerative) and splitting (divisive) strategies.

Thus, we can say that Hierarchical clustering is the overall concept whereas the Agglomerative clustering relates to a specific implementation (bottom-up approach)

Now, we extend our discussion on Agglomerative clustering by Comparing it with the Divisive Clustering

It is to be noted that, the Agglomerative clustering works as (bottom-up approach) in contrast to **divisive clustering**, which follows a **top-down approach**. In divisive clustering, all data points initially belong to a single cluster, and the algorithm repeatedly splits the cluster into smaller clusters until each data point becomes a separate cluster or a stopping condition is met.

The key differences between the two are as follows:

- Agglomerative clustering starts with many small clusters and merges them, whereas divisive clustering starts with one large cluster and divides it.
- Agglomerative clustering is more commonly used because it is computationally simpler, while divisive clustering is more complex and less frequently used in practice.
- Agglomerative clustering builds the hierarchy from bottom to top, whereas divisive clustering builds it from top to bottom.

here, we revisit the Working of Agglomerative Clustering, which follows the following steps:

Initially, each data point is treated as an individual cluster.

- The distance between all pairs of clusters is calculated.
- The two closest clusters are merged into one cluster.
- After merging, the distance matrix is updated based on the chosen linkage method.
- This process continues iteratively until all points are grouped into a single cluster or until a desired number of clusters is obtained.

We also learned that the Different linkage methods can be used to measure the distance between clusters, such as:

- Single linkage (minimum distance)
- Complete linkage (maximum distance)
- Average linkage (average distance)
- Centroid linkage (distance between centroids)

Example 4: Consider the following data points representing students' scores:

$$A(1,1), B(2,1), C(5,4), D(6,5)$$

Use Agglomerative Clustering to prepare clusters

Solution:

Step 1: Initial Clusters

Each point is its own cluster:

$$\{A\}, \{B\}, \{C\}, \{D\}$$

Step 2: Compute Distances

- Distance between A and B is small
- Distance between C and D is small

Step 3: Merge Closest Clusters

Merge A and B:

$$\{A, B\}, \{C\}, \{D\}$$

Merge C and D:

$$\{A, B\}, \{C, D\}$$

Step 4: Final Merge

Merge both clusters:

$$\{A, B, C, D\}$$

In this example, points A and B are grouped together first because they are closest to each other, and similarly, C and D form another cluster. Finally, these two clusters are merged at a higher distance level. This shows how agglomerative clustering builds a hierarchy of clusters step by step based on similarity.

If we represent this using a dendrogram, A and B will merge at a lower level, while the merge between the two larger clusters will occur at a higher level, indicating lesser similarity.

One more example for better understanding of Agglomerative clustering

Example 5: Apply the Agglomerative Clustering algorithm on the following four one-dimensional data points using single linkage distance:

$$A = 2, B = 4, C = 10, D = 12$$

Perform the clustering step by step until all points are merged into one cluster. Also show the order in which clusters are formed and interpret the result.

Solution:

Step 1: Write the Initial Clusters

Initially, each point is considered as a separate cluster:

$$\{A\} = \{2\}, \{B\} = \{4\}, \{C\} = \{10\}, \{D\} = \{12\}$$

So the initial clusters are:

$$\{2\}, \{4\}, \{10\}, \{12\}$$

Step 2: Compute the Distance Matrix

Using

$$d(x_i, x_j) = |x_i - x_j|$$

we calculate the pairwise distances.

$$\text{Distance between A and B: } d(A, B) = |2 - 4| = 2$$

$$\text{Distance between A and C: } d(A, C) = |2 - 10| = 8$$

$$\text{Distance between A and D: } d(A, D) = |2 - 12| = 10$$

$$\text{Distance between B and C: } d(B, C) = |4 - 10| = 6$$

$$\text{Distance between B and D: } d(B, D) = |4 - 12| = 8$$

$$\text{Distance between C and D: } d(C, D) = |10 - 12| = 2$$

So, the distance table is:

Pair	Distance
A, B	2
A, C	8

Pair	Distance
A, D	10
B, C	6
B, D	8
C, D	2

Step 3: Find the Smallest Distance and Merge the Closest Clusters

The minimum distance is: 2

This occurs for:

- A and B
- C and D

Since both are equally close, we can merge either pair first. Let us merge A and B first.

So the first merged cluster is:

$$\{A, B\} = \{2, 4\}$$

Now the clusters become:

$$\{2, 4\}, \{10\}, \{12\}$$

Step 4: Update Distances After First Merge

Now we compute the distance between cluster $\{A, B\}$ and the remaining points using single linkage.

Distance between $\{A, B\}$ and C

$$d(\{A, B\}, C) = \min \{d(A, C), d(B, C)\}$$

$$d(\{A, B\}, C) = \min \{8, 6\} = 6$$

Distance between $\{A, B\}$ and D

$$d(\{A, B\}, D) = \min \{d(A, D), d(B, D)\}$$

$$d(\{A, B\}, D) = \min \{10, 8\} = 8$$

Distance between C and D

$$d(C, D) = 2$$

So the updated distances are:

Pair	Distance
$\{A, B\}, C$	6
$\{A, B\}, D$	8
C, D	2

Step 5: Merge the Next Closest Clusters

Again, the smallest distance is: 2

So merge C and D: $\{C, D\} = \{10, 12\}$

Now the clusters become: $\{2, 4\}, \{10, 12\}$

Step 6: Compute Distance Between the Two New Clusters

Now calculate the distance between clusters $\{A, B\}$ and $\{C, D\}$ using single linkage.

$$d(\{A, B\}, \{C, D\}) = \min \{d(A, C), d(A, D), d(B, C), d(B, D)\}$$

Substitute the values:

$$d(\{A, B\}, \{C, D\}) = \min \{8, 10, 6, 8\}$$

$$d(\{A, B\}, \{C, D\}) = 6$$

Step 7: Final Merge

Now only two clusters remain:

$$\{2, 4\}, \{10, 12\}$$

Their distance is 6, so they are merged.

$$\{A, B, C, D\} = \{2, 4, 10, 12\}$$

Now all data points belong to one cluster.

The order of Final Cluster Formation is as follows:

The order of merging is:

1. A and B merge at distance 2
2. C and D merge at distance 2
3. $\{A, B\}$ and $\{C, D\}$ merge at distance 6

Dendrogram-Level Interpretation is as follows:

If we imagine the dendrogram:

- A and B join at height 2
- C and D join at height 2
- The two clusters $\{A, B\}$ and $\{C, D\}$ join at height 6

This indicates that A and B are very similar to each other, and C and D are also very similar to each other. However, the two resulting groups are farther apart from one another.

Thus, the Final Answer involves the Cluster merging sequence as follows:

$$\{2\}, \{4\}, \{10\}, \{12\}$$

$$\{2, 4\}, \{10\}, \{12\}$$

$$\{2, 4\}, \{10, 12\}$$

$$\{2, 4, 10, 12\}$$

Merge distances:

- A and B at distance 2
- C and D at distance 2
- $\{A, B\}$ and $\{C, D\}$ at distance 6

Results of the above numerical reflects that the Agglomerative Clustering algorithm first grouped the closest points together. Since $A = 2$ and $B = 4$ are very near, they formed one cluster. Similarly, $C = 10$ and $D = 12$ formed another cluster because they are also close. Finally, these two clusters merged at a larger distance of 6. This clearly shows that the data naturally forms **two smaller groups**: one around 2 and 4, and another around 10 and 12.

From a practical point of view, if we cut the dendrogram before the final merge, we would obtain two meaningful clusters:

$$\{2, 4\} \text{ and } \{10, 12\}$$

Thus, the numerical demonstrates how Agglomerative Clustering helps in discovering the hierarchical grouping structure in data.

To solve an Agglomerative Clustering numerical, always follow this order.

- First, treat each point as an individual cluster.
- Second, calculate the distance between all pairs of clusters.
- Third, merge the two nearest clusters.
- Fourth, update the distance matrix according to the chosen linkage method.
- Fifth, repeat the process until all points are merged into one cluster.

So the learning flow is:

Individual points → Compute distances → Merge nearest clusters → Update distances
→ Repeat

The main idea is that agglomerative clustering builds the hierarchy from the bottom upward by combining the most similar objects first.

The advantages and limitations of Agglomerative clustering are as follows:

Advantages of Agglomerative Clustering

- Agglomerative clustering does not require the number of clusters to be specified in advance.
- It provides a hierarchical structure that can be visualized using dendrograms.
- It is easy to understand and interpret.
- It is also flexible, as different linkage methods can be used to define similarity.

Limitations of Agglomerative Clustering

- Agglomerative clustering is computationally expensive and not suitable for very large datasets.
- Once clusters are merged, the process cannot be reversed.
- It is sensitive to noise and outliers, and the results may vary depending on the choice of linkage method.

We learned that the agglomerative clustering can be understood as a process of gradually combining similar data points into larger groups. It starts with the most basic units (individual points) and builds up a structure by merging the closest elements. This bottom-up approach makes it intuitive and closely aligned with how humans naturally group similar items.

Also, we understood that Agglomerative clustering is a fundamental technique within hierarchical clustering that builds clusters step by step from individual data points. By comparing it with divisive clustering, we understand two opposite strategies for forming clusters—merging versus splitting. Due to its simplicity and interpretability, agglomerative clustering is widely used in exploratory data analysis and forms a strong foundation for understanding clustering techniques in machine learning.

☛ Check Your Progress 4

Q1. What is Agglomerative Clustering? Explain its basic approach.

.....

.....

Q2. Explain the step-by-step working of Agglomerative Clustering.

.....

.....

Q3. What is single linkage in Agglomerative Clustering?

.....

.....

Q4. Differentiate between Agglomerative and Divisive Clustering.

.....

.....

11.2.4 ASSOCIATION RULES

Association Rule Learning is an important unsupervised learning technique used to discover interesting relationships, patterns, or associations among variables in large datasets. Unlike clustering, which groups similar data points, association rules focus on identifying **co-occurrence relationships**, that is, how items or events are related to each other. This technique is widely used in applications such as market basket analysis, recommendation systems, and customer behaviour analysis.

The basic idea of association rules is to find rules of the form:

$$X \Rightarrow Y$$

where X and Y are sets of items. This rule means that if itemset X occurs, then itemset Y is likely to occur as well. The goal is to identify such rules that are strong and meaningful based on certain evaluation measures.

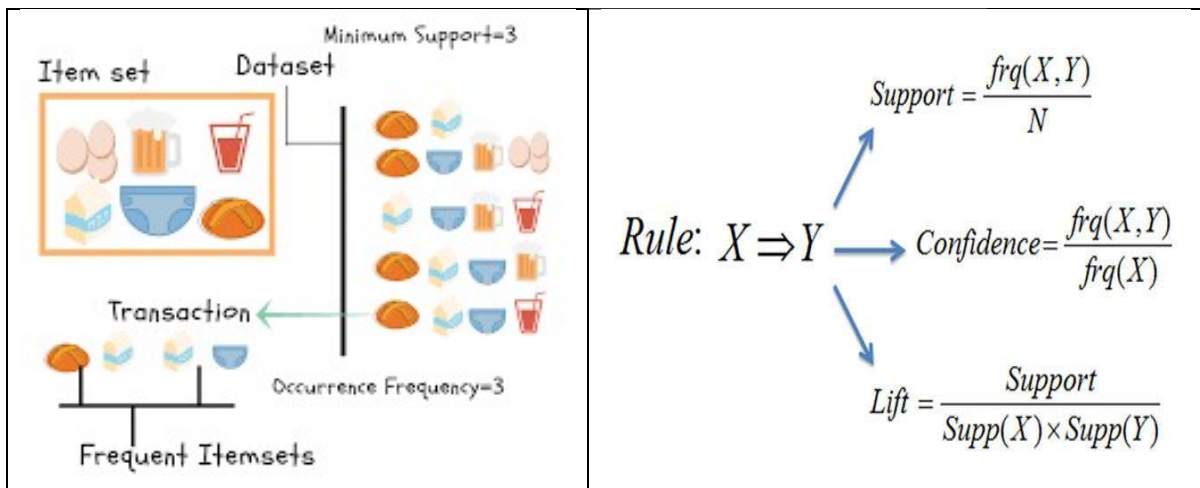


Figure: Association Rule Concept

The figure represents a typical **market basket analysis** scenario. For example, customers who buy bread may also buy butter. This relationship can be expressed as:

$$\text{Bread} \Rightarrow \text{Butter}$$

Such rules help businesses understand purchasing patterns and make better decisions such as product placement, cross-selling, and promotions.

Following are the Key Measures in Association Rules that are used to evaluate the strength of association rules, three important metrics are:

1. Support

Support measures how frequently an itemset appears in the dataset.

$$\text{Support}(X) = \frac{\text{Number of transactions containing } X}{\text{Total number of transactions}}$$

For a rule:

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Total transactions}}$$

2. Confidence

Confidence measures how often items in Y appear in transactions that contain X .

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

3. Lift

Lift measures how much more likely Y is to occur given X , compared to its general occurrence.

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

Example 6: Consider a supermarket where customers frequently buy items like bread, milk, and butter. By analysing transaction data, we may discover that customers who buy bread often buy butter as well. This insight can help the store place these items close to each other or offer bundled discounts.

Consider the following transaction dataset:

Transaction ID	Items Purchased
T1	Bread, Milk
T2	Bread, Butter
T3	Milk, Butter
T4	Bread, Milk, Butter
T5	Bread, Milk

Total transactions = 5

Solution:

Step 1: Calculate Support

Let us calculate support for:

Support(Bread)

Bread appears in T1, T2, T4, T5 → 4 transactions

$$Support(Bread) = \frac{4}{5} = 0.8$$

Support(Bread ∪ Butter)

Bread and Butter appear together in T2 and T4 → 2 transactions

$$Support(Bread \Rightarrow Butter) = \frac{2}{5} = 0.4$$

Step 2: Calculate Confidence

$$Confidence(Bread \Rightarrow Butter) = \frac{Support(Bread \cup Butter)}{Support(Bread)}$$

$$= \frac{0.4}{0.8} = 0.5$$

Step 3: Calculate Lift

First find Support(Butter):

Butter appears in T2, T3, T4 → 3 transactions

$$Support(Butter) = \frac{3}{5} = 0.6$$

Now,

$$Lift(Bread \Rightarrow Butter) = \frac{0.5}{0.6} \approx 0.83$$

Final Results

- Support = 0.4
- Confidence = 0.5
- Lift ≈ 0.83

The support value of 0.4 indicates that 40% of transactions contain both bread and butter. The confidence value of 0.5 means that 50% of customers who buy bread also buy butter. However, the lift value is less than 1, which suggests that buying bread does not strongly increase the likelihood of buying butter compared to random chance. Therefore, this rule is not very strong.

Let's explore the working of Association Rule with one more Numerical Example

Example 7: A mobile accessories store wants to analyze customer buying patterns using **Association Rule Learning**. The transaction data of 5 customers is given below:

Transaction ID	Items Purchased
T1	Mobile Cover, Earphones
T2	Mobile Cover, Charger
T3	Earphones, Charger
T4	Mobile Cover, Earphones, Charger
T5	Mobile Cover, Earphones

Using the above data, find the **Support, Confidence, and Lift** for the association rule:

$$\text{Mobile Cover} \Rightarrow \text{Earphones}$$

Also interpret whether this rule is strong or not.

Solution: In this numerical, we have to study the rule: $X \Rightarrow Y$
where:

- X = Mobile Cover
- Y = Earphones

This rule means:

If a customer buys a Mobile Cover, then the customer is likely to buy Earphones.

To evaluate this rule, we calculate three important measures:

- **Support**
- **Confidence**
- **Lift**

Equations Used are as follows:

(a) Support of the Rule

Support tells us how frequently both items occur together in the total dataset.

$$\text{Support}(X \Rightarrow Y) = \frac{\text{Number of transactions containing both } X \text{ and } Y}{\text{Total number of transactions}}$$

(b) Confidence of the Rule

Confidence tells us how often Y occurs when X has already occurred.

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

It can also be written as:

$$\text{Confidence}(X \Rightarrow Y) = \frac{\text{Number of transactions containing both } X \text{ and } Y}{\text{Number of transactions containing } X}$$

(c) Lift of the Rule

Lift tells us how much more likely Y is to occur when X occurs, compared to the normal occurrence of Y .

$$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)}$$

Meaning of Variables

- X : antecedent itemset or starting item
- Y : consequent itemset or resulting item
- $X \cup Y$: transactions containing both X and Y
- $\text{Support}(X)$: proportion of transactions containing X
- $\text{Support}(Y)$: proportion of transactions containing Y
- $\text{Confidence}(X \Rightarrow Y)$: probability of buying Y when X is bought
- $\text{Lift}(X \Rightarrow Y)$: strength of association between X and Y

Given Data

Transaction ID	Items Purchased
T1	Mobile Cover, Earphones
T2	Mobile Cover, Charger
T3	Earphones, Charger
T4	Mobile Cover, Earphones, Charger
T5	Mobile Cover, Earphones

Total number of transactions:

$$N = 5$$

Rule to analyze:

Mobile Cover $\Rightarrow$ Earphones

Step 1: Find Transactions Containing Mobile Cover

Mobile Cover appears in: Transaction T1, T2, T4, T5

So, number of transactions containing Mobile Cover = 4

$$Support(\text{Mobile Cover}) = \frac{4}{5} = 0.8$$

Step 2: Find Transactions Containing Earphones

Earphones appear in: Transaction T1, T3, T4, T5

So, number of transactions containing Earphones = 4

$$Support(\text{Earphones}) = \frac{4}{5} = 0.8$$

Step 3: Find Transactions Containing Both Mobile Cover and Earphones

Mobile Cover and Earphones appear together in: Transaction T1, T4, T5

So, number of transactions containing both = 3

$$Support(\text{Mobile Cover} \Rightarrow \text{Earphones}) = \frac{3}{5} = 0.6$$

Thus, the support of the rule is: 0.6 or 60%

Step 4: Calculate Confidence

Using the formula: $Confidence(\text{Mobile Cover} \Rightarrow \text{Earphones}) = \frac{Support(\text{Mobile Cover} \cup \text{Earphones})}{Support(\text{Mobile Cover})}$

Substitute the values:

$$Confidence(\text{Mobile Cover} \Rightarrow \text{Earphones}) = \frac{0.6}{0.8}$$

$$Confidence(\text{Mobile Cover} \Rightarrow \text{Earphones}) = 0.75$$

Thus, the confidence of the rule is: 0.75 or 75%

Step 5: Calculate Lift

Using the formula:

$$Lift(\text{Mobile Cover} \Rightarrow \text{Earphones}) = \frac{Confidence(\text{Mobile Cover} \Rightarrow \text{Earphones})}{Support(\text{Earphones})}$$

Substitute the values:

$$Lift(\text{Mobile Cover} \Rightarrow \text{Earphones}) = \frac{0.75}{0.8}$$

$$Lift(\text{Mobile Cover} \Rightarrow \text{Earphones}) = 0.9375$$

Thus, the lift of the rule is: 0.9375 or 93.75%

Final Answer: For the rule

Mobile Cover $\Rightarrow$ Earphones

the results are:

- Support = 0.6 or 60%
- Confidence=0.75 or 75%
- Lift=0.9375 or 93.75%

The support value of **0.6** means that 60% of all transactions contain both Mobile Cover and Earphones together. The confidence value of **0.75** means that 75% of customers who buy a Mobile Cover also buy Earphones. This suggests a fairly frequent co-occurrence between the two products.

However, the lift value is **0.9375**, which is less than 1. This means that although these two items appear together in several transactions, buying a Mobile Cover does not increase the probability of buying Earphones beyond their usual occurrence in the dataset. Therefore, this association rule is **not very strong**.

While solving Association Rule numerical, always follow this sequence. First, identify the rule $X \Rightarrow Y$. Then count the transactions containing X , the transactions containing Y , and the transactions containing both X and Y . After that, calculate support, confidence, and lift using the standard formulas. Finally, interpret the lift value carefully. If lift is greater than 1, the association is positive and strong; if it is equal to 1, there is no special association; and if it is less than 1, the rule is weak.

So the learning flow is:

Count transactions $\rightarrow$ Find Support $\rightarrow$ Find Confidence $\rightarrow$ Find Lift $\rightarrow$ Interpret

The Advantages and Limitations of Association Rules are as follows:

Advantages of Association Rules:

- Association rule learning is simple and easy to understand.
- It helps uncover hidden relationships in large datasets.
- It is widely used in business applications such as recommendation systems and marketing strategies.
- It can handle large volumes of data efficiently.

Limitations of Association Rules

- The algorithm can generate a large number of rules, many of which may not be useful.
- It requires careful selection of minimum support and confidence thresholds.
- It may not perform well with sparse datasets.
- Interpretation of rules can sometimes be complex when many variables are involved.

Thus, Association rules can be understood as discovering “if-then” relationships in data. The key idea is to identify patterns that frequently occur together and evaluate their strength using support, confidence, and lift. By understanding these measures, one can easily determine whether a rule is meaningful or not.

We learned that the Association Rule Learning is a powerful unsupervised technique that helps in discovering relationships between items in a dataset. By using measures like support, confidence, and lift, it allows us to evaluate the usefulness of these relationships. It plays a crucial role in real-world applications such as market basket analysis, recommendation systems, and decision-making processes.

☛ Check Your Progress 5

Q1. What is Association Rule Learning? Explain its purpose.

.....
.....

Q2. Explain the basic form of an association rule with an example.

.....
.....

Q3. What are the key measures used in Association Rule Learning?

.....
.....

Q4. List applications of Association Rule Learning in real-world scenarios..

.....
.....

11.3 SUMMARY

This unit introduces the concept of **unsupervised learning models**, which are used to analyse data without predefined labels or outcomes. Unlike supervised learning, where models learn from known input-output pairs, unsupervised learning focuses on discovering hidden patterns, structures, and relationships within the data. These models are particularly useful in scenarios such as customer segmentation, pattern recognition, and market analysis, where the goal is to extract meaningful insights from raw and unstructured data.

The unit begins with an overview of unsupervised learning models and their significance in data analysis. It then explores **K-Means Clustering**, a widely used partitioning technique that groups data into a predefined number of clusters by minimizing the distance between data

points and their respective centroids. K-Means is simple, efficient, and suitable for large datasets, making it one of the most popular clustering algorithms.

Further, the unit discusses **Hierarchical Clustering**, which builds a hierarchy of clusters in the form of a dendrogram. This approach allows data to be analyzed at different levels of granularity without requiring the number of clusters to be specified in advance. A key method within this category is **Agglomerative Clustering**, which follows a bottom-up approach by initially treating each data point as a separate cluster and then merging the closest clusters step by step. This method provides a clear understanding of how clusters are formed and related to each other.

Finally, the unit covers **Association Rules**, which focus on discovering relationships and patterns among variables in large datasets. Using measures such as support, confidence, and lift, association rule learning identifies how items are related and how frequently they occur together. This technique is widely used in applications like market basket analysis and recommendation systems.

Overall, this unit provides a comprehensive understanding of key unsupervised learning techniques, highlighting their working principles, applications, and significance in data analysis. It equips learners with the ability to uncover hidden patterns in data and apply appropriate models to solve real-world problems effectively.

11.4 ANSWERS

☛ Check Your Progress 1

Q1. What is Unsupervised Learning? How does it differ from supervised learning?

Solution: refer to Section 11.0

Q2. List the applications of Unsupervised Learning in real-world scenarios.

Solution: refer to Section 11.0

Q3. What are Unsupervised Learning Models? Explain their significance in data analysis.

Solution: refer to Section 11.2

Q4. Briefly explain different types of Unsupervised Learning techniques..

Solution: refer to Section 11.2

☛ Check Your Progress 2

Q1. What is K-Means Clustering? Explain its objective.

Solution: refer to Sub-section 11.2.1

Q2. Explain the working steps of the K-Means algorithm

Solution: refer to Sub-section 11.2.1

Q3. What is the role of centroid in K-Means Clustering?.

Solution: refer to Sub-section 11.2.1

Q4. Explain the Elbow Method for selecting optimal number of clusters (K).

Solution: refer to Sub-section 11.2.1

☛ Check Your Progress 3

Q1. What is Hierarchical Clustering? Explain its key concept.

Solution: refer to Sub-section 11.2.2

Q2. Differentiate between Agglomerative and Divisive approaches in Hierarchical Clustering.

Solution: refer to Sub-section 11.2.2

Q3. What is a dendrogram and how is it used in clustering?

Solution: refer to Sub-section 11.2.2

Q4. Explain different linkage methods used in Hierarchical Clustering.

Solution: refer to Sub-section 11.2.2

☛ Check Your Progress 4

Q1. What is Agglomerative Clustering? Explain its basic approach.

Solution: refer to Sub-section 11.2.3

Q2. Explain the step-by-step working of Agglomerative Clustering.

Solution: refer to Sub-section 11.2.3

Q3. What is single linkage in Agglomerative Clustering?

Solution: refer to Sub-section 11.2.3

Q4. Differentiate between Agglomerative and Divisive Clustering.

Solution: refer to Sub-section 11.2.3

☛ Check Your Progress 5

Q1. What is Association Rule Learning? Explain its purpose.

Solution: refer to Sub-section 11.2.4

Q2. Explain the basic form of an association rule with an example.

Solution: refer to Sub-section 11.2.4

Q3. What are the key measures used in Association Rule Learning?

Solution: refer to Sub-section 11.2.4

Q4. List applications of Association Rule Learning in real-world scenarios.

Solution: refer to Sub-section 11.2.4

11.5 REFERENCES/FURTHER READINGS

1. *Pattern Recognition and Machine Learning* by Christopher M. Bishop, published by Springer (1st Edition, 2006, ISBN: 978-0387310732).
2. *Machine Learning* by Tom M. Mitchell, published by McGraw-Hill (1st Edition, 1997, ISBN: 978-0070428072).

3. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* by Aurélien Géron, published by O'Reilly Media (2nd Edition, 2019, ISBN: 978-1492032649).
4. *Data Mining: Concepts and Techniques* by Jiawei Han, Micheline Kamber, and Jian Pei, published by Morgan Kaufmann (3rd Edition, 2011, ISBN: 978-0123814791)
5. *Practical Statistics for Data Scientists* by Peter Bruce, Andrew Bruce, and Peter Gedeck, published by O'Reilly Media (2nd Edition, 2020, ISBN: 978-1492072942).
6. *Business Analytics: Data Analysis and Decision Making* by S. Christian Albright and Wayne L. Winston, published by Cengage Learning (7th Edition, 2020, ISBN: 978-0357131787).
7. *Decision Analysis for Management Judgment* by Paul Goodwin and George Wright, published by Wiley (5th Edition, 2014, ISBN: 978-1119974017).
8. *Handbook of Statistical Analysis and Data Mining Applications* by Robert Nisbet, John Elder, and Gary Miner, published by Elsevier (2nd Edition, 2017, ISBN: 978-0124166455).
9. *Data Mining and Analysis: Fundamental Concepts and Algorithms* by Mohammed J. Zaki and Wagner Meira Jr., published by Cambridge University Press (1st Edition, 2014, ISBN: 978-0521766333).
10. *Core Concepts in Data Analysis: Summarization, Correlation and Visualization* by Boris Mirkin, published by Springer (1st Edition, 2011, ISBN: 978-0857292872).
11. *Applied Predictive Modeling*, Max Kuhn and Kjell Johnson, 1st Edition, Springer, 2013.
12. *An Introduction to Statistical Learning with Applications in R*, Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani, 1st Edition, Springer, 2013.
13. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Trevor Hastie, Robert Tibshirani and Jerome Friedman, 2nd Edition, Springer, 2009.
14. *An Introduction to Statistical Learning with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, published by Springer (2nd Edition, 2021, ISBN: 978-1071614174).
15. *Time Series Analysis: Forecasting and Control* by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung, published by Wiley (5th Edition, 2015, ISBN: 978-1118675021)