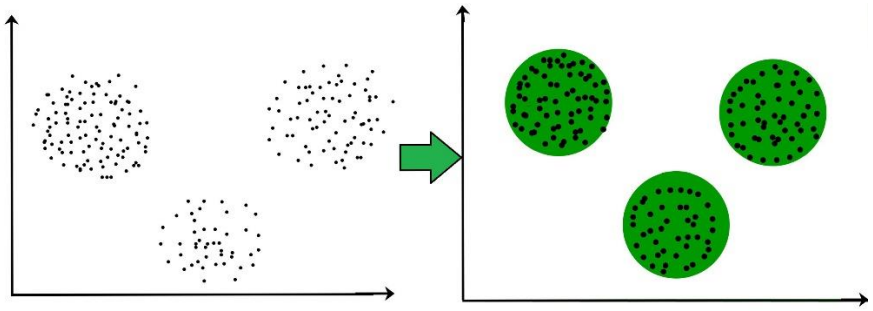
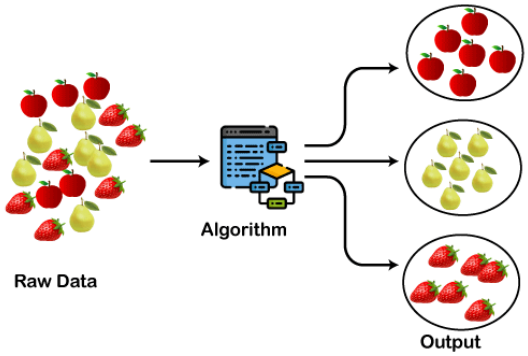
What is clustering? Explain in detail.

- The task of grouping data points based on their similarity with each other is called Clustering or Cluster Analysis.
- Clustering is the process of dividing the unlabeled data or data points into different clusters such that similar data points fall in the same cluster than those which differ from the others. In simple words, the aim of the clustering process is to segregate groups with similar traits and assign them into clusters.
- Clustering aims at forming groups of homogeneous data points from a heterogeneous dataset.
- It evaluates the similarity based on a metric like Euclidean distance, Cosine similarity, Manhattan distance, etc. and then group the points with highest similarity score together.
- For Example, In the graph given below, we can clearly see that there are 3 circular clusters forming on the basis of distance.



- The below diagram explains the working of the clustering algorithm. We can see the different fruits are divided into several groups with similar properties.



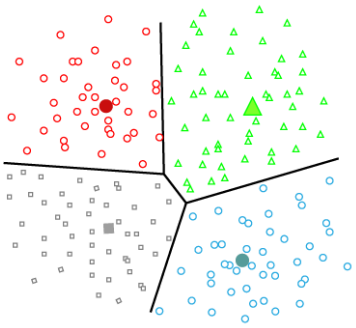
Types of Clustering Methods

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

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

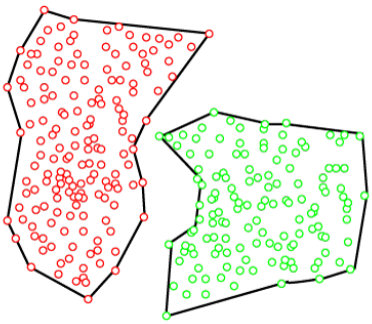
Partitioning Clustering :

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



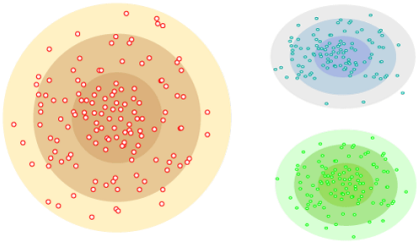
Density-Based Clustering :

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



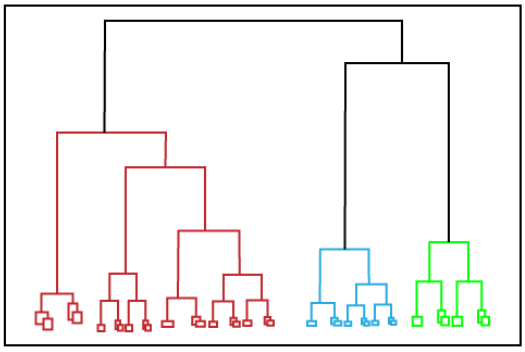
Distribution Model-Based Clustering :

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



Hierarchical Clustering :

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



Explain K Means clustering.

- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.
- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.
- Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
- It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

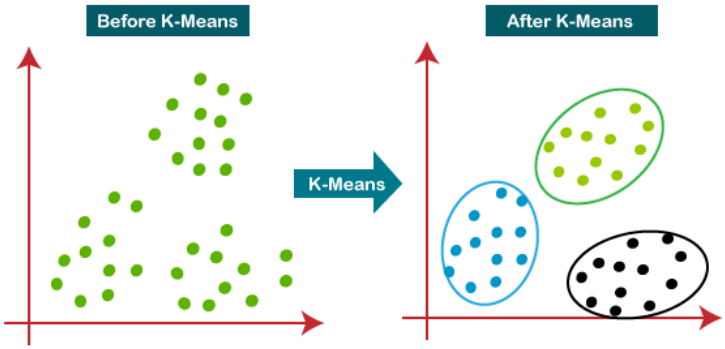
Objective of k-means clustering :

- 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.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

The below diagram explains the working of the K-means Clustering Algorithm:



Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

Explain Elbow Method in K Means clustering

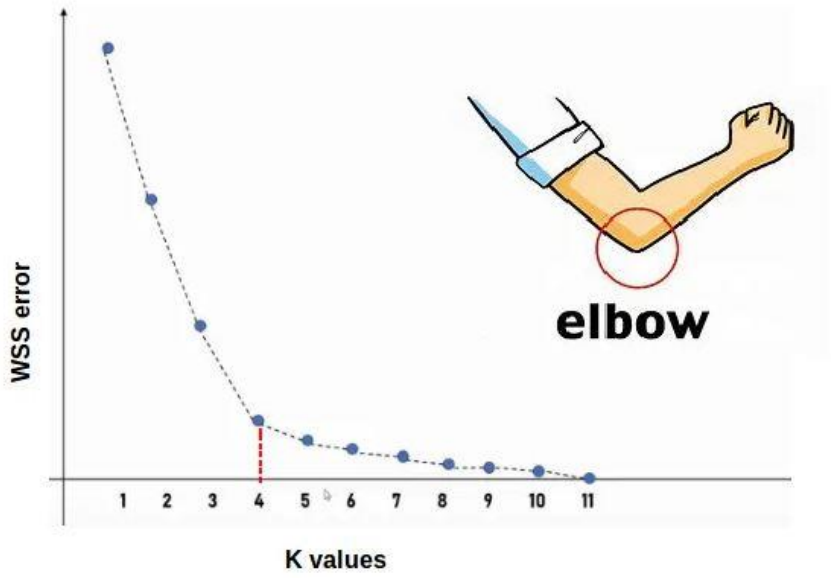
- K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science.
- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.
- The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms .Selecting a lower number of clusters will result in underfitting while specifying a higher number of clusters can result in overfitting. Unfortunately, there is no definitive way to find the optimal number.
- we don't have any method to determine the exact accurate value of clusters K but we can estimate the value using some techniques, including Elbow method.

Elbow method

- The Elbow Method is a visual approach used to determine the ideal 'K' (number of clusters) in K-means clustering.
- It operates by calculating the Within-Cluster Sum of Squares (WCSS), which is the total of the squared distances between data points and their cluster center.
- However, there is a point where increasing K no longer leads to a significant decrease in WCSS, and the rate of decrease slows down. This point is often referred to as the **elbow**.

K Means Clustering Using the Elbow Method

- In the Elbow Method, we systematically experiment with different numbers of clusters (K) ranging from 1 to 10.
- With each K value, we compute the Within-Cluster Sum of Squares (WCSS). When we plot WCSS against K, the resulting graph resembles an elbow.
- As we increase the number of clusters, the WCSS value begins to decrease. Notably, the WCSS is at its highest when K=1.



Explain Hierarchical clustering.

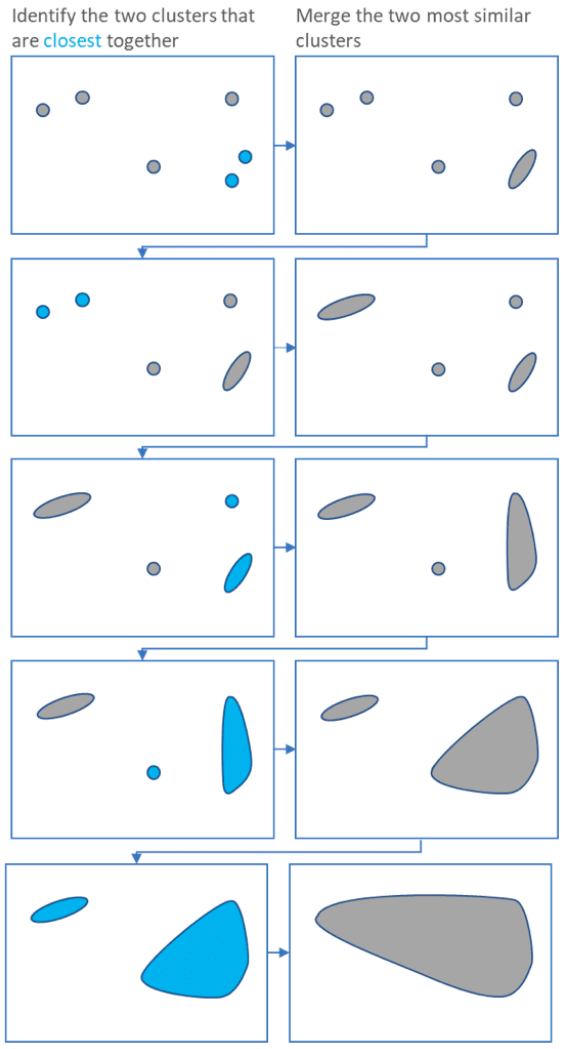
- Hierarchical clustering, also known as *hierarchical cluster analysis,* is an algorithm that groups similar objects into groups called *clusters*.
- The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.
- Hierarchical clustering is a method of cluster analysis in data mining that creates a hierarchical representation of the clusters in a dataset.
- The method starts by treating each data point as a separate cluster and then iteratively combines the closest clusters until a stopping criterion is reached.
- The result of hierarchical clustering is a tree-like structure, called a dendrogram, which illustrates the hierarchical relationships among the clusters.

How hierarchical clustering works

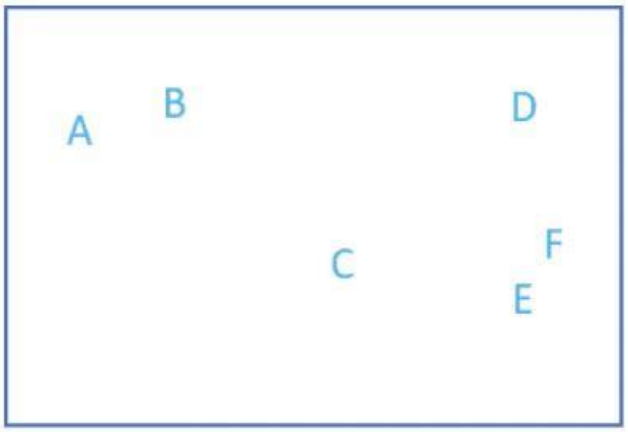
- Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

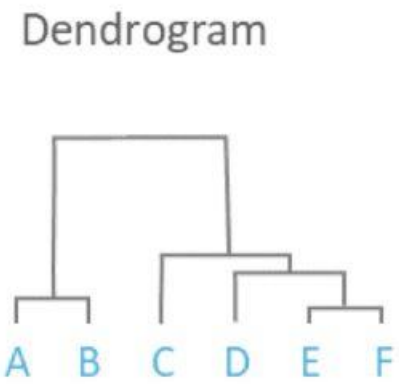
1. identify the two clusters that are closest together,
2. merge the two most similar clusters.

- This iterative process continues until all the clusters are merged together. This is illustrated in the diagrams below.



The main output of Hierarchical Clustering is a *dendrogram*, which shows the hierarchical relationship between the clusters:





Types of Hierarchical Clustering

Basically, there are two types of hierarchical Clustering:

1. Agglomerative Clustering
2. Divisive clustering

What is Agglomerative Hierarchical clustering?

- Hierarchical clustering is a method of cluster analysis in data mining that creates a hierarchical representation of the clusters in a dataset.
- The method starts by treating each data point as a separate cluster and then iteratively combines the closest clusters until a stopping criterion is reached.

Basically, there are two types of hierarchical Clustering:

- Agglomerative Clustering
- Divisive clustering

1. Agglomerative Clustering

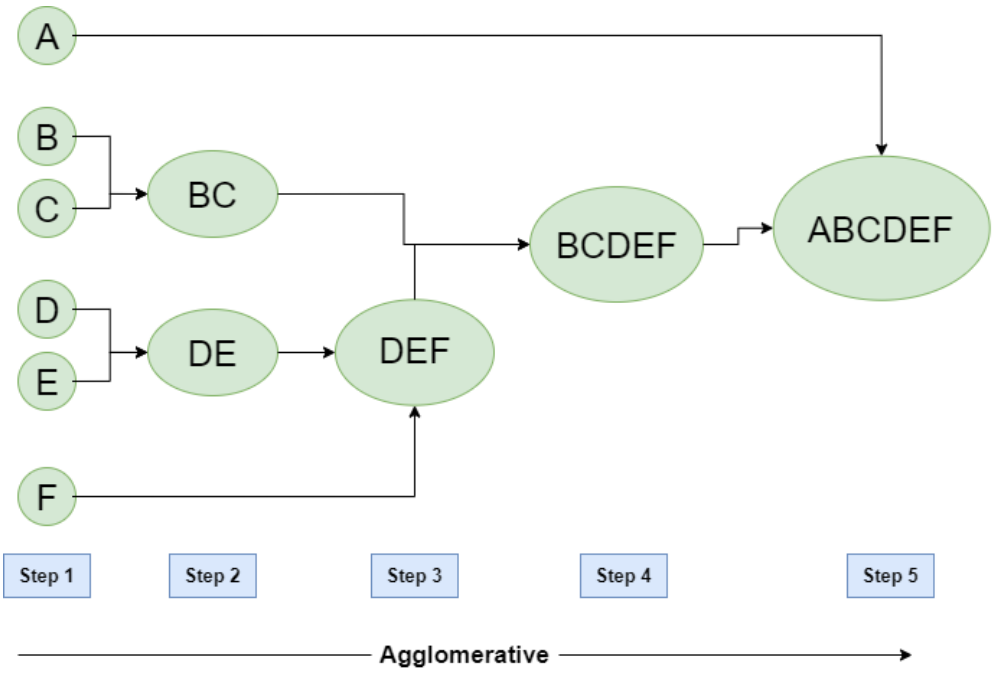
Initially consider every data point as an **individual** Cluster and at every step, merge the nearest pairs of the cluster. (It is a bottom-up method). At first, every dataset is considered an individual entity or cluster. At every iteration, the clusters merge with different clusters until one cluster is formed.

The algorithm for Agglomerative Hierarchical Clustering is:

- Calculate the similarity of one cluster with all the other clusters (calculate proximity matrix)
- Consider every data point as an individual cluster
- Merge the clusters which are highly similar or close to each other.
- Recalculate the proximity matrix for each cluster
- Repeat Steps 3 and 4 until only a single cluster remains.

graphical representation of this algorithm using a dendrogram.

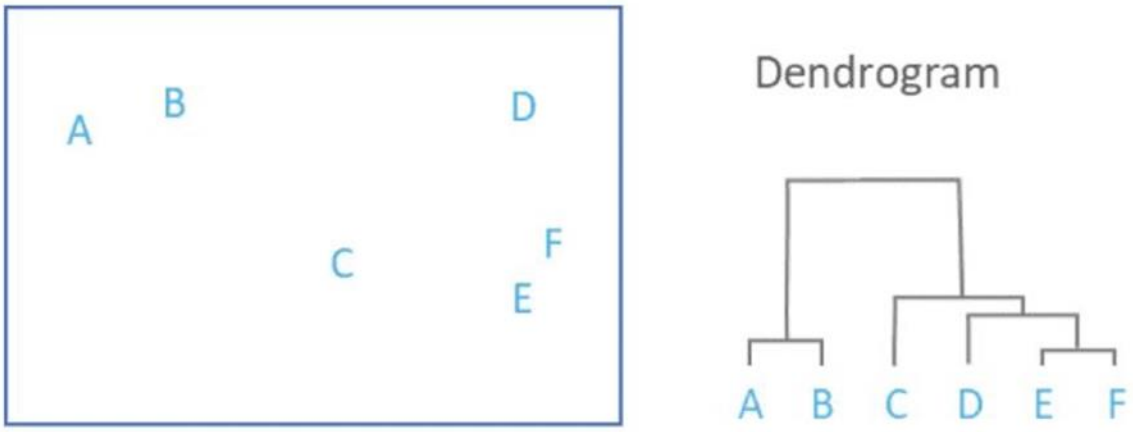
Let's say we have six data points **A, B, C, D, E, and F**.



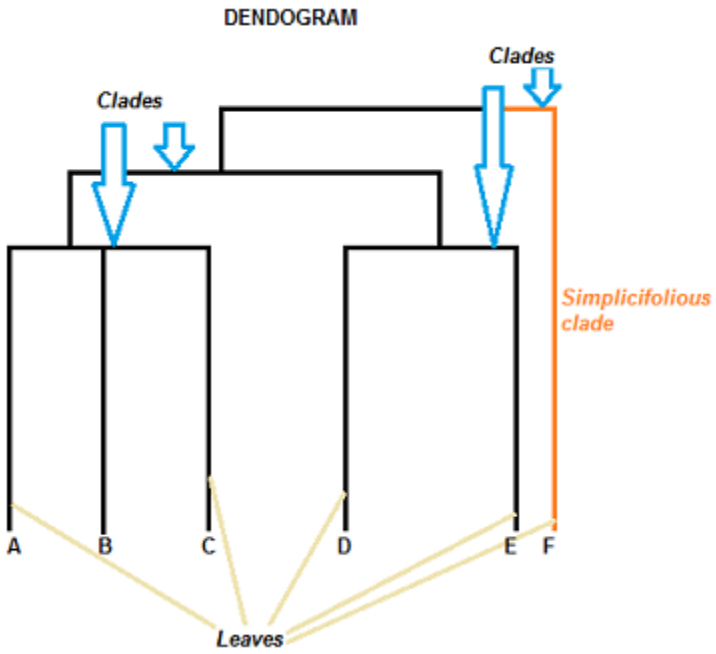
- **Step-1:** Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.
- **Step-2:** In the second step comparable clusters are merged together to form a single cluster. Let's say cluster (B) and cluster (C) are very similar to each other therefore we merge them in the second step similarly to cluster (D) and (E) and at last, we get the clusters [(A), (BC), (DE), (F)]
- **Step-3:** We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]
- **Step-4:** Repeating the same process; The clusters DEF and BC are comparable and merged together to form a new cluster. We're now left with clusters [(A), (BCDEF)].
- **Step-5:** At last, the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

Explain working of dendrogram in Hierarchical clustering

- Hierarchical clustering, also known as *hierarchical cluster analysis,* is an algorithm that groups similar objects into groups called *clusters*.
- Hierarchical clustering is a method of cluster analysis in data mining that creates a hierarchical representation of the clusters in a dataset.
- A *dendrogram* is a diagram that shows the hierarchical relationship between objects.
- It is most commonly created as an output from *hierarchical clustering*. The main use of a dendrogram is to work out the best way to allocate objects to clusters.
- The dendrogram below shows the hierarchical clustering of six *observations* shown on the *scatterplot* to the left



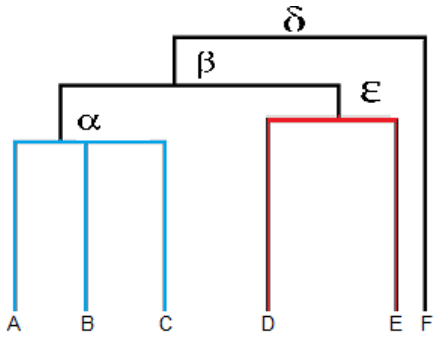
Parts of a Dendrogram:



- The *clade* is the branch. Usually labeled with Greek letters from left to right (e.g. α β, δ…).
- Each clade has one or more *leaves*. The leaves in the above image are:
    - Single (simplicifolius): F
    - Double (bifolius): D E
    - Triple (trifolious): A B C

How to Read a Dendrogram

- The clades are arranged according to how similar (or dissimilar) they are. Clades that are close to the same height are similar to each other;
- clades with different heights are dissimilar — the greater the difference in height, the more dissimilarity (you can measure similarity in many different ways;
- One of the most popular measures is Pearson's Correlation Coefficient).



- Leaves A, B, and C are more similar to each other than they are to leaves D, E, or F.
- Leaves D and E are more similar to each other than they are to leaves A, B, C, or F.
- Leaf F is substantially different from all of the other leaves.

Explain Association Rule mining.

Explain apriori algorithm.

Explain Eclat algorithm

Explain F-P Growth algorithm

- Association rule mining is a data mining technique that aims to discover interesting relationships, patterns, and correlations within large datasets.
- It focuses on identifying strong associations between different items or variables in the data. It presents these associations in the form of if-then rules, commonly known as association rules.
- An association rule consists of an antecedent (if part) and a consequent (then part). The dataset contains an antecedent, and we derive a consequent by using the antecedent.
- At a basic level, association rule mining involves the use of machine learning models to analyze data for patterns, called *co-occurrences*, in a database.
- The association rule learning is one of the very important concepts of machine learning, and it is employed in **Market Basket analysis, Web usage mining, continuous production, etc.**
- Here market basket analysis is a technique used by the various big retailer to discover the associations between items. We can understand it by taking an example of a supermarket, as in a supermarket, all products that are purchased together are put together.

- Association rule learning works on the concept of If and Else Statement, such as if A then B.

- Here the If element is called **antecedent**, and then statement is called as **Consequent**.
- These types of relationships where we can find out some association or relation between two items is known *as single cardinality*.
- It is all about creating rules, and if the number of items increases, then cardinality also increases accordingly.

So, to measure the associations between thousands of data items, there are several metrics. These metrics are given below:

1. Support
2. Confidence
3. Lift

Types of Association Rule Learning:

Association rule learning can be divided into three algorithms:

Apriori Algorithm

- This algorithm uses frequent datasets to generate association rules.
- It is designed to work on the databases that contain transactions. This algorithm uses a breadth-first search and Hash Tree to calculate the itemset efficiently.
- It is mainly used for market basket analysis and helps to understand the products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

Steps for Apriori Algorithm

Below are the steps for the apriori algorithm:

Step-1: Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

Step-2: Take all supports in the transaction with higher support value than the minimum or selected support value.

Step-3: Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

Step-4: Sort the rules as the decreasing order of lift.

Apriori Algorithm Working

We will understand the apriori algorithm using an example and mathematical calculation:

Example of Apriori: Support threshold=50%, Confidence= 60%

TABLE-1

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Solution:

Support threshold=50% => 0.5*6= 3 => min_sup=3

1. Count Of Each Item

TABLE-2

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

2. Prune Step: TABLE -2 shows that I5 item does not meet min_sup=3, thus it is deleted, only I1, I2, I3, I4 meet min_sup count.

TABLE-3

Item	Count
I1	4
I2	5
I3	4
I4	4

3. Join Step: Form 2-itemset. From TABLE-1 find out the occurrences of 2-itemset.

TABLE-4

Item	Count
I1,I2	4
I1,I3	3
I1,I4	2
I2,I3	4
I2,I4	3
I3,I4	2

4. Prune Step: TABLE -4 shows that item set {I1, I4} and {I3, I4} does not meet min_sup, thus it is deleted.

TABLE-5

Item	Count
I1,I2	4
I1,I3	3
I2,I3	4
I2,I4	3

5. Join and Prune Step: Form 3-itemset. From the TABLE- 1 find out occurrences of 3-itemset. From TABLE-5, find out the 2-itemset subsets which support min_sup.

We can see for itemset {I1, I2, I3} subsets, {I1, I2}, {I1, I3}, {I2, I3} are occurring in TABLE-5 thus {I1, I2, I3} is frequent.

We can see for itemset {I1, I2, I4} subsets, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} is not frequent, as it is not occurring in TABLE-5 thus {I1, I2, I4} is not frequent, hence it is deleted.

TABLE-6

Item
I1,I2,I3
I1,I2,I4
I1,I3,I4
I2,I3,I4

Only {I1, I2, I3} is frequent.

6. Generate Association Rules: From the frequent itemset discovered above the association could be:

{I1, I2} => {I3}

Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4)* 100 = 75%

{I1, I3} => {I2}

Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3)* 100 = 100%

{I2, I3} => {I1}

Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4)* 100 = 75%

{I1} => {I2, I3}

Confidence = support {I1, I2, I3} / support {I1} = (3/ 4)* 100 = 75%

{I2} => {I1, I3}

Confidence = support {I1, I2, I3} / support {I2 = (3/ 5)* 100 = 60%

{I3} => {I1, I2}

Confidence = support {I1, I2, I3} / support {I3} = (3/ 4)* 100 = 75%

This shows that all the above association rules are strong if minimum confidence threshold is 60%.

Eclat Algorithm

- Eclat algorithm stands for **Equivalence Class Transformation**.
- This algorithm uses a depth-first search technique to find frequent itemsets in a transaction database.
- It performs faster execution than Apriori Algorithm.
- While the Apriori algorithm works in a horizontal sense imitating the Breadth-First Search of a graph, the ECLAT algorithm works in a vertical manner just like the Depth-First Search of a graph.

Let us now understand the above stated working with an example:-

Consider the following transactions record:-

Transaction Id	Bread	Butter	Milk	Coke	Jam
T1	1	1	0	0	1
T2	0	1	0	1	0
T3	0	1	1	0	0
T4	1	1	0	1	0
T5	1	0	1	0	0
T6	0	1	1	0	0
T7	1	0	1	0	0
T8	1	1	1	0	1
T9	1	1	1	0	0

The above-given data is a boolean matrix where for each cell (i, j), the value denotes whether the j'th item is included in the i'th transaction or not. 1 means true while 0 means false.

We now call the function for the first time and arrange each item with it's tidset in a tabular fashion:-

k = 1, minimum support = 2

Item	Tidset
Bread	{T1, T4, T5, T7, T8, T9}
Butter	{T1, T2, T3, T4, T6, T8, T9}
Milk	{T3, T5, T6, T7, T8, T9}
Coke	{T2, T4}
Jam	{T1, T8}

We now recursively call the function till no more item-tidset pairs can be combined:-

k = 2

Item	Tidset
{Bread, Butter}	{T1, T4, T8, T9}
{Bread, Milk}	{T5, T7, T8, T9}

Item	Tidset
{Bread, Coke}	{T4}
{Bread, Jam}	{T1, T8}
{Butter, Milk}	{T3, T6, T8, T9}
{Butter, Coke}	{T2, T4}
{Butter, Jam}	{T1, T8}
{Milk, Jam}	{T8}

k = 3

Item	Tidset
{Bread, Butter, Milk}	{T8, T9}
{Bread, Butter, Jam}	{T1, T8}

k = 4

Item	Tidset
{Bread, Butter, Milk, Jam}	{T8}

We stop at k = 4 because there are no more item-tidset pairs to combine.

Since minimum support = 2, we conclude the following rules from the given dataset:-

Items Bought	Recommended Products
Bread	Butter
Bread	Milk
Bread	Jam
Butter	Milk
Butter	Coke
Butter	Jam
Bread and Butter	Milk

Items Bought	Recommended Products
Bread and Butter	Jam

F-P Growth Algorithm

- The F-P growth algorithm stands for **Frequent Pattern**, and it is the improved version of the Apriori Algorithm.
- It represents the database in the form of a tree structure that is known as a frequent pattern or tree. The purpose of this frequent tree is to extract the most frequent patterns.
- The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance.

This algorithm works as follows:

- First, it compresses the input database creating an FP-tree instance to represent frequent items.
- After this first step, it divides the compressed database into a set of conditional databases, each associated with one frequent pattern.
- Finally, each such database is mined separately.

Example

Support threshold=50%, Confidence= 60%

Table 1:

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Solution: Support threshold=50% => 0.5*6= 3 => min_sup=3

Table 2: Count of each item

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

Table 3: Sort the itemset in descending order.

Item	Count
I2	5
I1	4
I3	4
I4	4

Advantages of FP Growth Algorithm

Here are the following advantages of the FP growth algorithm, such as:

- This algorithm needs to scan the database twice when compared to Apriori, which scans the transactions for each iteration.
- The pairing of items is not done in this algorithm, making it faster.
- The database is stored in a compact version in memory.
- It is efficient and scalable for mining both long and short frequent patterns.

Disadvantages of FP-Growth Algorithm

This algorithm also has some disadvantages, such as:

- FP Tree is more cumbersome and difficult to build than Apriori.
- It may be expensive.
- The algorithm may not fit in the shared memory when the database is large.