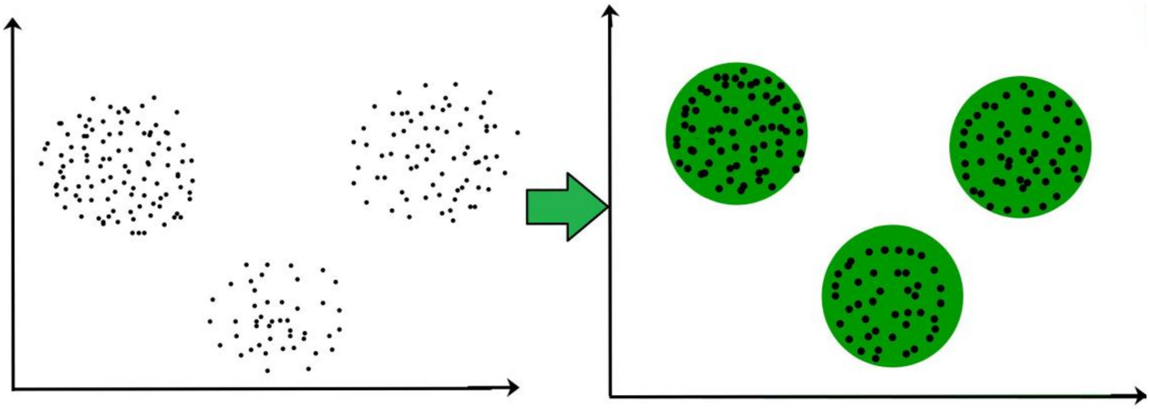
Q1.What is clustering? Explain in detail.

Ans: The task of grouping data points based on their similarity with each other is called Clustering or Cluster Analysis. This method is defined under the branch of Unsupervised Learning, which aims at gaining insights from unlabelled data points, that is, unlike supervised learning we don't have a target variable.

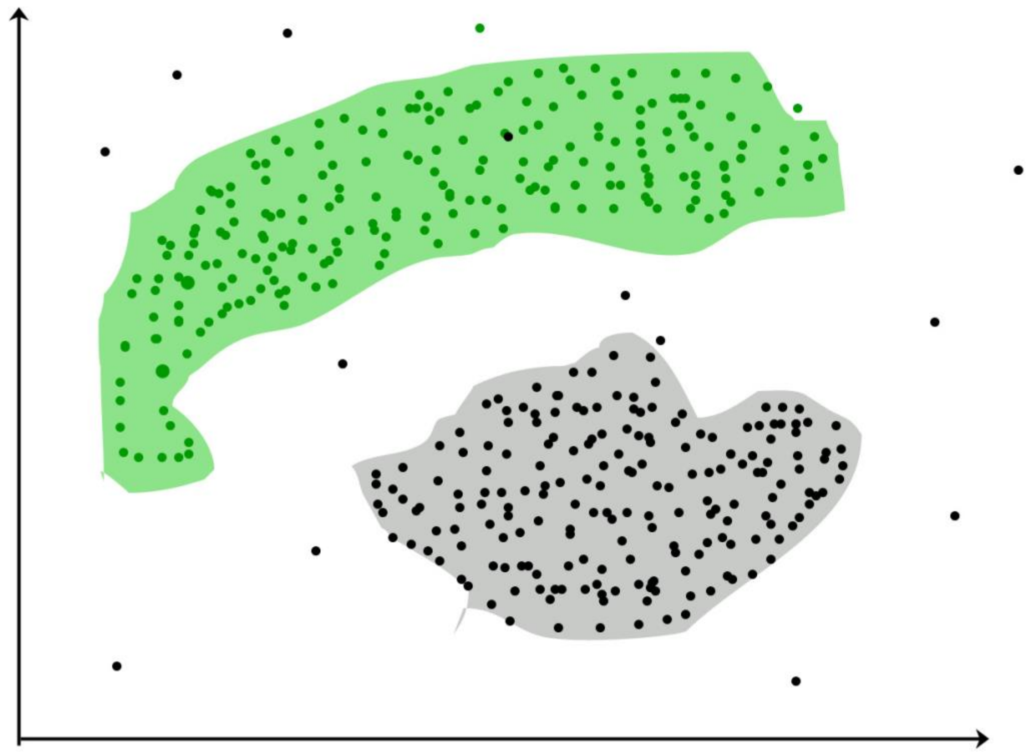
Clustering aims at forming groups of homogeneous data points from a heterogeneous dataset. It evaluates the similarity based on a metric like Euclidean distance, Cosine similarity, Manhattan distance, etc. and then group the points with highest similarity score together.

For Example, In the graph given below, we can clearly see that there are 3 circular clusters forming on the basis of distance.



Now it is not necessary that the clusters formed must be circular in shape. The shape of clusters can be arbitrary. There are many algortihms that work well with detecting arbitrary shaped clusters.

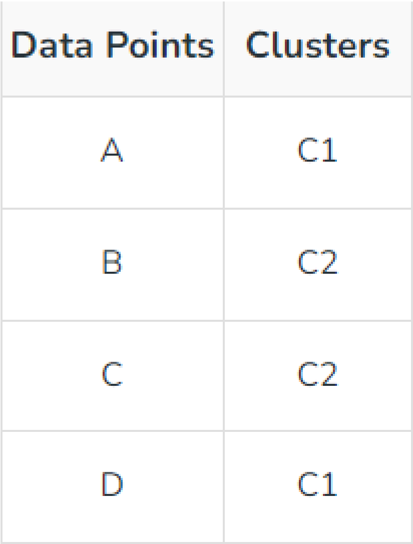
For example, In the below given graph we can see that the clusters formed are not circular in shape.



Types of Clustering

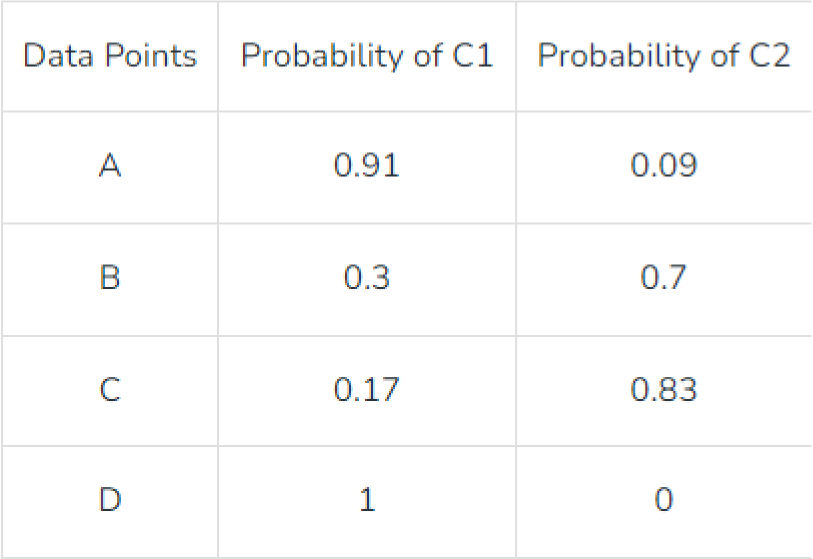
Broadly speaking, there are 2 types of clustering that can be performed to group similar data points:

- **Hard Clustering:** In this type of clustering, each data point belongs to a cluster completely or not. For example, Let’s say there are 4 data point and we have to cluster them into 2 clusters. So each data point will either belong to cluster 1 or cluster 2.



Data Points	Clusters
A	C1
B	C2
C	C2
D	C1

Soft Clustering: In this type of clustering, instead of assigning each data point into a separate cluster, a probability or likelihood of that point being that cluster is evaluated. For example, Let's say there are 4 data point and we have to cluster them into 2 clusters. So we will be evaluating a probability of a data point belonging to both clusters. This probability is calculated for all data points.



Data Points	Probability of C1	Probability of C2
A	0.91	0.09
B	0.3	0.7
C	0.17	0.83
D	1	0

Uses of Clustering

Market Segmentation – Businesses use clustering to group their customers and use targeted advertisements to attract more audience.

Market Basket Analysis – Shop owners analyze their sales and figure out which items are majorly bought together by the customers. For example, In USA, according to a study diapers and beers were usually bought together by fathers.

Social Network Analysis – Social media sites use your data to understand your browsing behaviour and provide you with targeted friend recommendations or content recommendations.

Medical Imaging – Doctors use Clustering to find out diseased areas in diagnostic images like X-rays.

Anomaly Detection – To find outliers in a stream of real-time dataset or forecasting fraudulent transactions we can use clustering to identify them.

Simplify working with large datasets – Each cluster is given a cluster ID after clustering is complete. Now, you may reduce a feature set's whole feature set into its cluster ID. Clustering is effective when it can represent a complicated case with a straightforward cluster ID. Using the same principle, clustering data can make complex datasets simpler.

Q2.Explain K Means clustering.

Ans: K-Means Clustering Algorithm

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering.

What is K-Means Algorithm?

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 is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

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.

It is a centroid-based algorithm, where each cluster is associated with a centroid.

The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

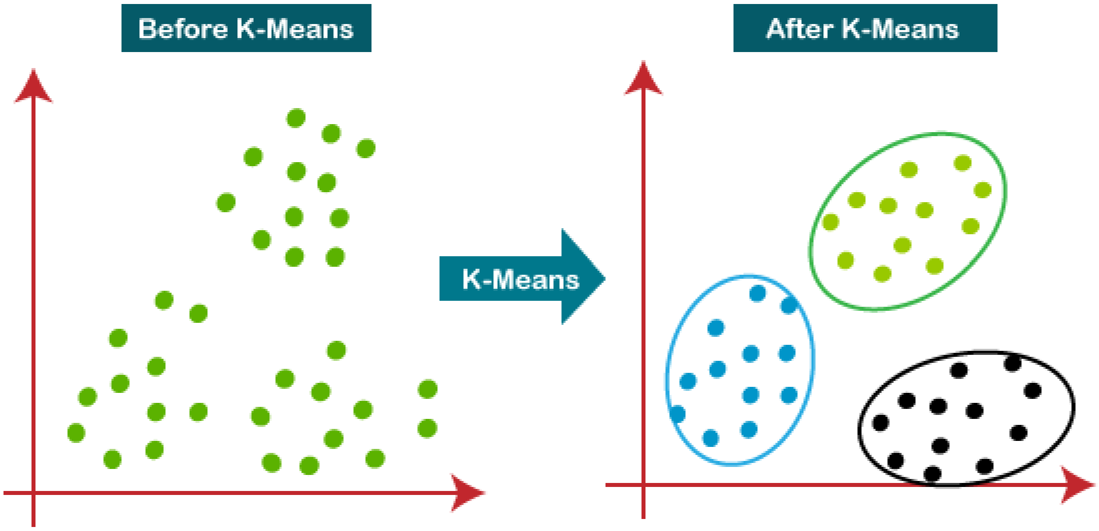
The k-means clustering algorithm mainly performs two tasks:

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.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

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



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

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.

Q3.Explain Elbow Method in K Means clustering.

Ans: A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered. Since we do not have any predefined number of clusters in unsupervised learning. We tend to use some method that can help us decide the best number of clusters. In the case of K-Means clustering, we use Elbow Method for defining the best number of clustering

As we know in the k-means clustering algorithm we randomly initialize k clusters and we iteratively adjust these k clusters till these k-centroids riches in an equilibrium state. However, the main thing we do before initializing these clusters is that determine how many clusters we have to use.

For determining K(numbers of clusters) we use Elbow method. Elbow Method is a technique that we use to determine the number of centroids(k) to use in a k-means clustering algorithm. In this method to determine the k-value we continuously iterate for k=1 to k=n (Here n is the hyperparameter that we choose as per our requirement). For every value of k, we calculate the within-cluster sum of squares (WCSS) value.

WCSS - It is defined as the sum of square distances between the centroids and each points.

Now For determining the best number of clusters(k) we plot a graph of k versus their WCSS value. Surprisingly the graph looks like an elbow (which we will see later). Also, When k=1 the WCSS has the highest value but with increasing k value WCSS value starts to decrease. We choose that value of k from where the graph starts to look like a straight line.

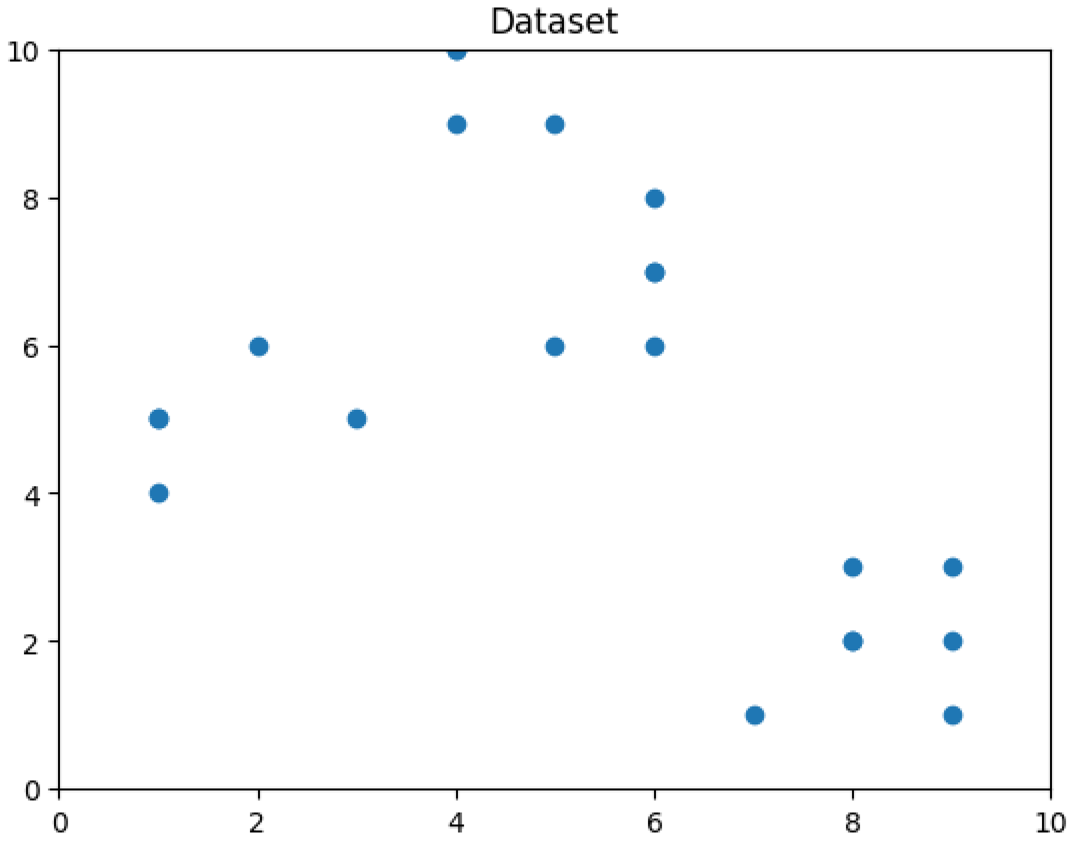
Now For determining the best number of clusters(k) we plot a graph of k versus their WCSS value. Surprisingly the graph looks like an elbow (which we will see

later). Also, When k=1 the WCSS has the highest value but with increasing k value WCSS value starts to decrease. We choose that value of k from where the graph starts to look like a straight line.

Implementation of the Elbow Method Usking Sklearn in Python:

step 1: Importing the required libraries

Step 2: Creating and Visualizing the data



From the above visualization, we can see that the optimal number of clusters should be around 3. But visualizing the data alone cannot always give the right answer. Hence we demonstrate the following steps.

We now define the following:-

Distortion: It is calculated as the average of the squared distances from the cluster centers of the respective clusters to each data point. Typically, the Euclidean distance metric is used.

Distortion = 1/n * Σ(distance(point, centroid)^2)

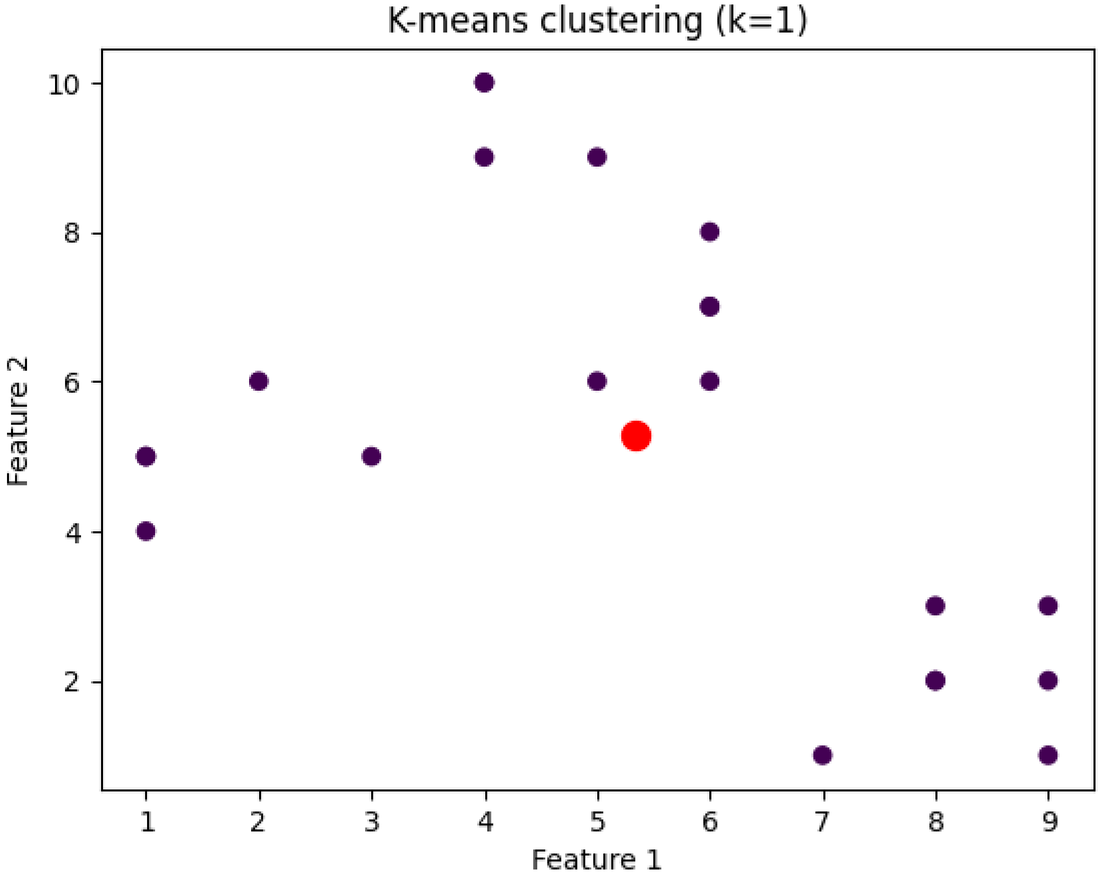
Inertia: It is the sum of the squared distances of samples to their closest cluster center.

Inertia = Σ(distance(point, centroid)^2)

We iterate the values of k from 1 to n and calculate the values of distortions for each value of k and calculate the distortion and inertia for each value of k in the given range.

step 3: Building the clustering model and calculating the values of the Distortion and Inertia

Step 4: Tabulating and Visualizing the Results



Q4.Explain Hierarchical clustering.

Ans: A Hierarchical clustering method works via grouping data into a tree of clusters. Hierarchical clustering begins by treating every data point as a separate cluster. Then, it repeatedly executes the subsequent steps:

1.Identify the 2 clusters which can be closest together, and

2.Merge the 2 maximum comparable clusters. We need to continue these steps until all the clusters are merged together.

In Hierarchical Clustering, the aim is to produce a hierarchical series of nested clusters. A diagram called Dendrogram (A Dendrogram is a tree-like diagram that statistics the sequences of merges or splits) graphically represents this hierarchy and is an inverted tree that describes the order in which factors are merged (bottom-up view) or clusters are broken up (top-down view).

What is 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. The result of hierarchical clustering is a tree-like structure, called a dendrogram, which illustrates the hierarchical relationships among the clusters.

Hierarchical clustering has several advantages over other clustering methods:

The ability to handle non-convex clusters and clusters of different sizes and densities.

The ability to handle missing data and noisy data.

The ability to reveal the hierarchical structure of the data, which can be useful for understanding the relationships among the clusters.

Drawbacks of Hierarchical Clustering:

The need for a criterion to stop the clustering process and determine the final number of clusters.

The computational cost and memory requirements of the method can be high, especially for large datasets.

The results can be sensitive to the initial conditions, linkage criterion, and distance metric used.

In summary, Hierarchical clustering is a method of data mining that groups similar data points into clusters by creating a hierarchical structure of the clusters.

This method can handle different types of data and reveal the relationships among the clusters. However, it can have high computational cost and results can be sensitive to some conditions.

Types of Hierarchical Clustering

1.Agglomerative Clustering

2.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)

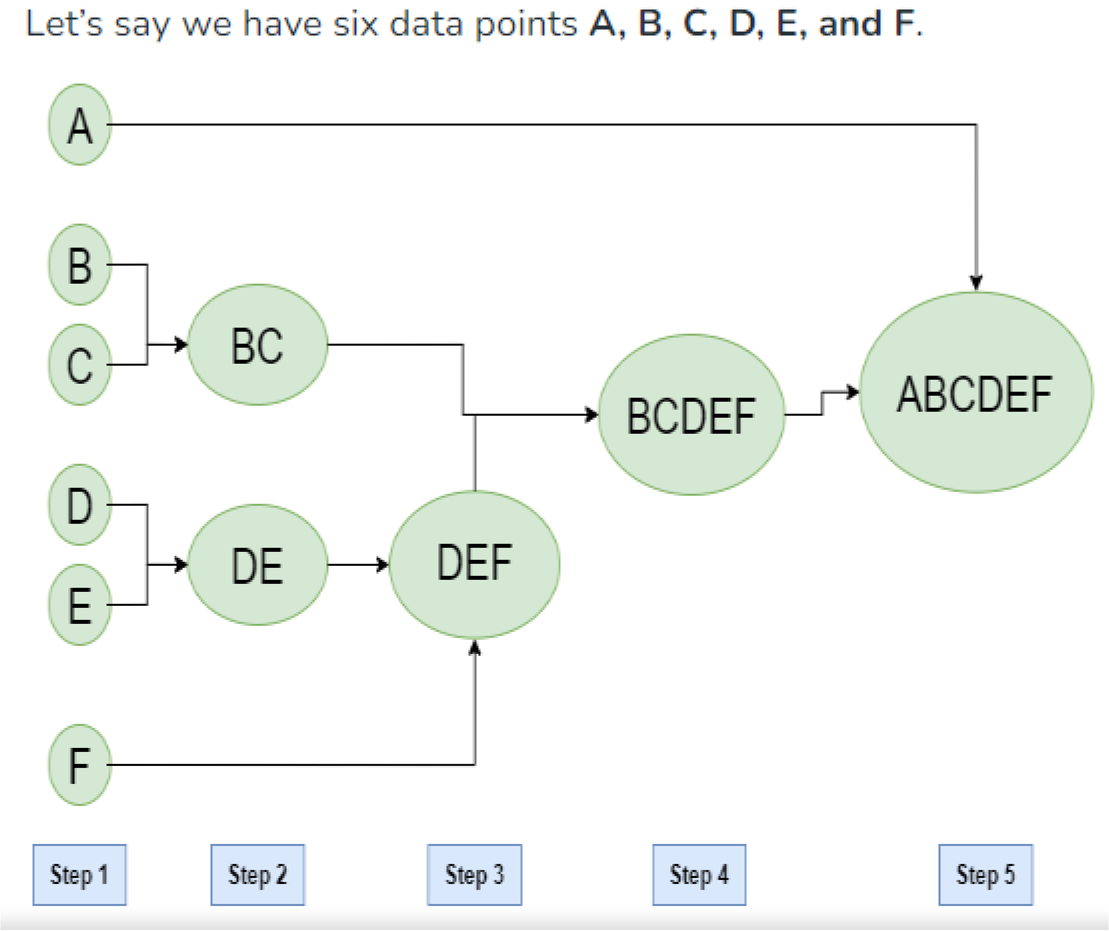
Consider every data point as an individual cluster

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.

Example:



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)]

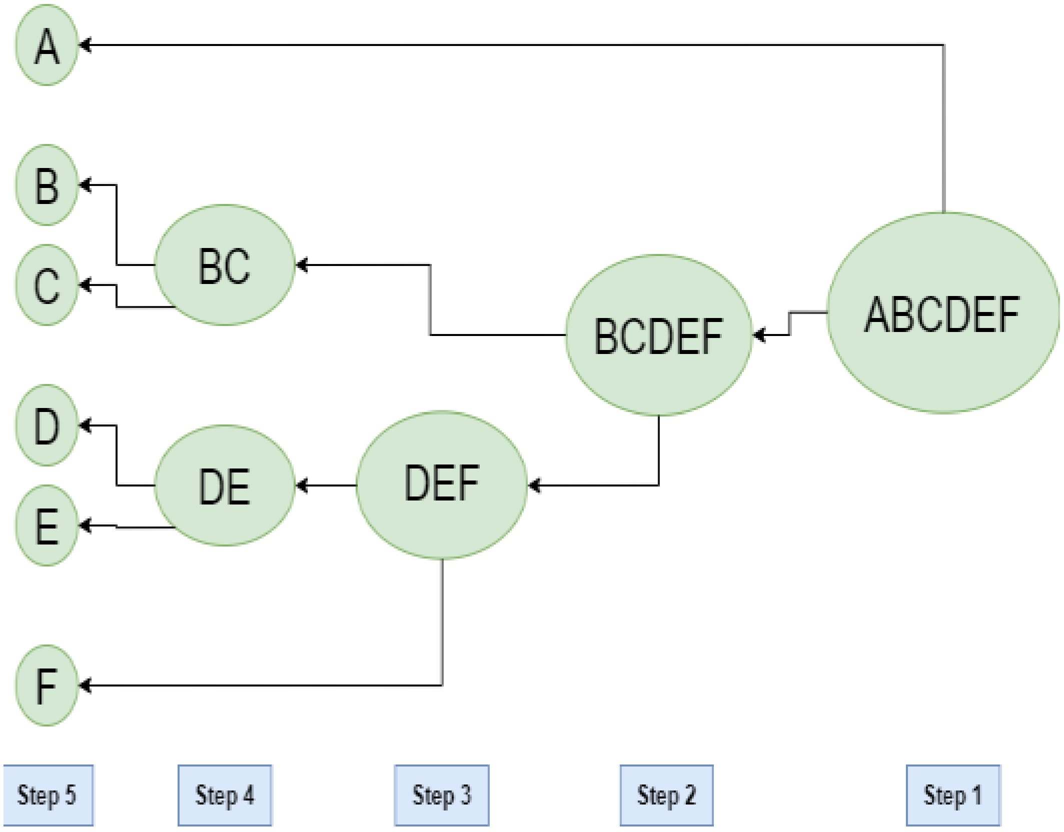
Step-3: We recalculate the proximity according to the algorithm and merge the two nearest clusters([(DE), (F)]) together to form new clusters as [(A), (BC), (DEF)]

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)].

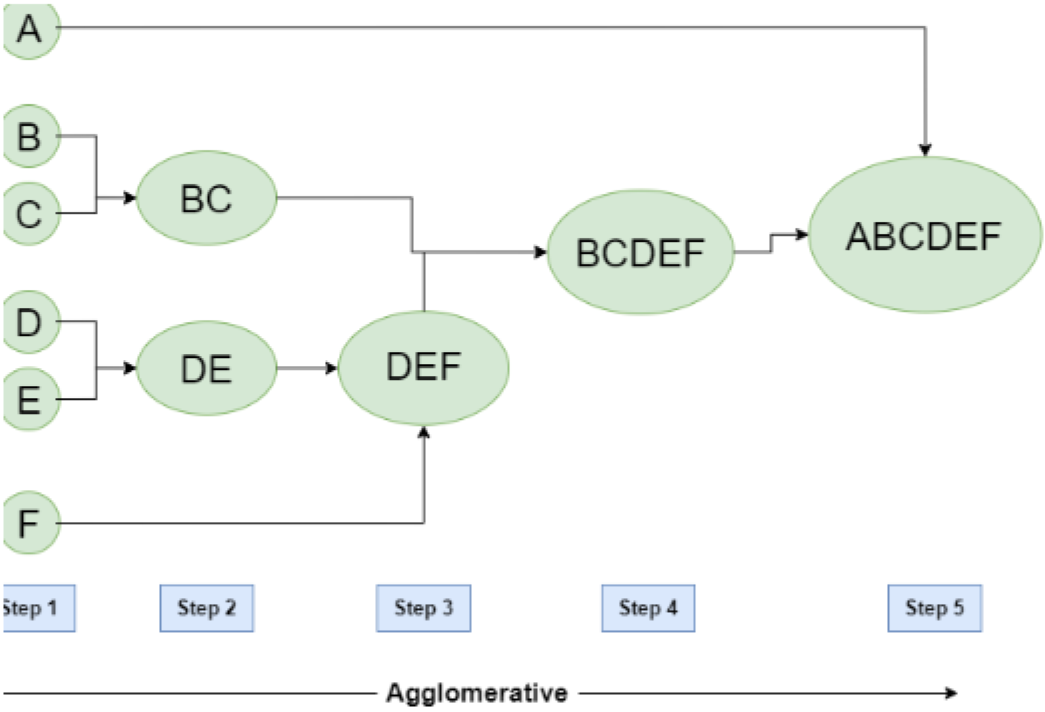
2. Divisive Hierarchical clustering:

We can say that Divisive Hierarchical clustering is precisely the opposite of Agglomerative Hierarchical clustering. In Divisive Hierarchical clustering, we take into account all of the data points as a single cluster and in every iteration, we separate the data points from the clusters which aren't comparable. In the end, we are left with N clusters.



Q5.What is Agglomerative Hierarchical clustering?

Ans: It is also known as the bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. 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.



Steps:

Consider each alphabet as a single cluster and calculate the distance of one cluster from all the other clusters.

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)]

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)]

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)].

At last, the two remaining clusters are merged together to form a single cluster [(ABCDEF)].

Q6.Explain working of dendrogram in Hierarchical clustering.

Ans: A dendrogram is a tree-like diagram used in hierarchical clustering to illustrate the arrangement of the clusters produced by the algorithm. Hierarchical clustering is a method of cluster analysis which builds a hierarchy of clusters. A dendrogram represents the relationships between the data points and the clusters they form as they are merged together.

Here's how a dendrogram works in hierarchical clustering:

1. **Distance Matrix Calculation**: Initially, the algorithm calculates the distance between each pair of data points. This distance could be based on various metrics such as Euclidean distance, Manhattan distance, etc. The distances are typically stored in a distance matrix.

2. **Initial Clusters**: Each data point starts as its own cluster.

3. **Cluster Fusion**: The algorithm proceeds by iteratively merging the closest clusters based on the distance between them. At each step, the two closest clusters are combined into a single cluster.

4. **Dendrogram Construction**: As clusters merge, the dendrogram is constructed vertically. The height at which two clusters are merged represents the distance between them. The longer the vertical line connecting two clusters, the larger the dissimilarity (or smaller the similarity) between them.

5. **Interpretation**: The dendrogram allows analysts to visually inspect the hierarchical structure of the clusters. Depending on the application, they can choose to cut the dendrogram at a certain height to obtain a specific number of clusters. This is known as "cutting the dendrogram" at a particular height.

6. **Cluster Identification**: By examining the dendrogram, analysts can also identify the similarity or dissimilarity between different clusters and individual data points.

7. **Termination**: The process continues until all data points have been merged into a single cluster or until a stopping criterion is met (e.g., a predetermined number of clusters).

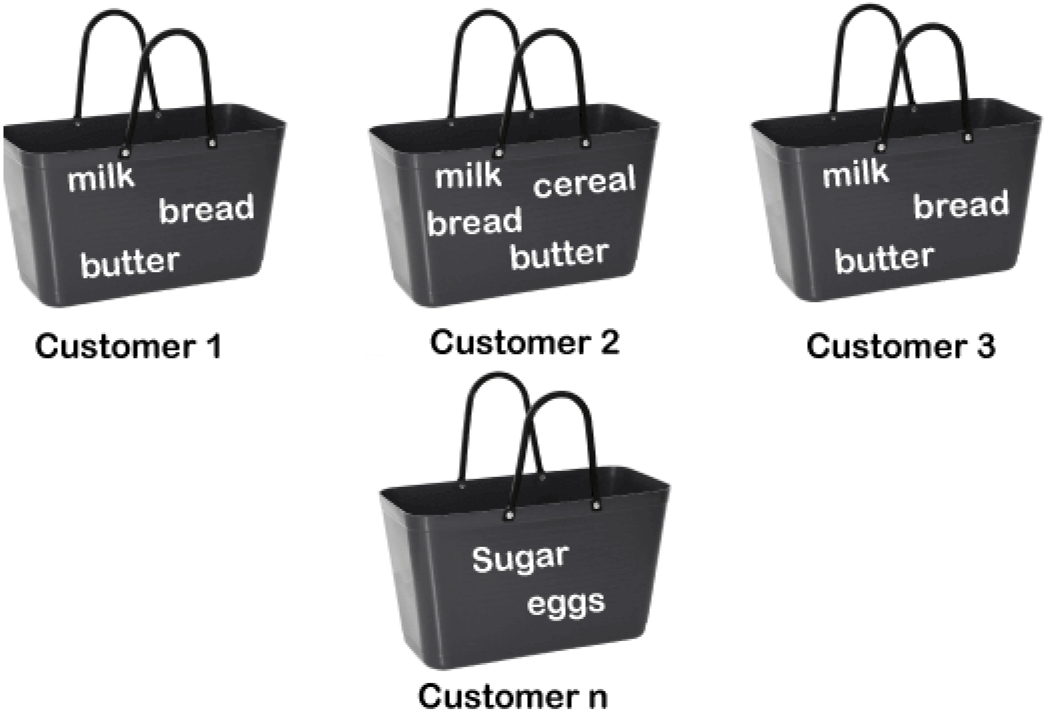
Overall, dendrograms provide a helpful visual representation of the hierarchical clustering process, enabling analysts to explore the structure of the data and make informed decisions about the number and composition of clusters.

Q7.Explain Association Rule mining.

Ans: Association rule learning is a type of unsupervised learning technique that checks for the dependency of one data item on another data item and maps accordingly so that it can be more profitable. It tries to find some interesting relations or associations among the variables of dataset. It is based on different rules to discover the interesting relations between variables in the 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.

For example, if a customer buys bread, he most likely can also buy butter, eggs, or milk, so these products are stored within a shelf or mostly nearby. Consider the below diagram:



Association rule learning can be divided into three types of algorithms:

Apriori

Eclat

F-P Growth Algorithm

How does Association Rule Learning work?

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

Association Rule Learning

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:

Support

Confidence

Lift

Support

Support is the frequency of A or how frequently an item appears in the dataset. It is defined as the fraction of the transaction T that contains the itemset X. If there are X datasets, then for transactions T, it can be written as:

$$\text{Supp(X)=}\frac{Freq(X)}{T}$$

Confidence

Confidence indicates how often the rule has been found to be true. Or how often the items X and Y occur together in the dataset when the occurrence of X is already given. It is the ratio of the transaction that contains X and Y to the number of records that contain X.

$$\text{Confidence=}\frac{Freq(X,Y)}{Freq(X)}$$

Lift

It is the strength of any rule, which can be defined as below formula:

$$\text{Lift=}\frac{Supp(X,Y)}{Supp(X)\times Supp(Y)}$$

Types of Association Rule Lerning

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.

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.

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.

Applications of Association Rule Learning

Market Basket Analysis: It is one of the popular examples and applications of association rule mining. This technique is commonly used by big retailers to determine the association between items.

Medical Diagnosis: With the help of association rules, patients can be cured easily, as it helps in identifying the probability of illness for a particular disease.

Protein Sequence: The association rules help in determining the synthesis of artificial Proteins.

It is also used for the Catalog Design and Loss-leader Analysis and many more other applications.

Q8.Explain apriori algorithm.

Ans: Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.

The primary objective of the apriori algorithm is to create the association rule between different objects. The association rule describes how two or more objects are related to one another. Apriori algorithm is also called frequent pattern mining. Generally, you operate the Apriori algorithm on a database that consists of a huge number of transactions. Let's understand the apriori algorithm with the help of an example; suppose you go to Big Bazar and buy different products. It helps the customers buy their products with ease and increases the sales performance of the Big Bazar. In this tutorial, we will discuss the apriori algorithm with examples.

We take an example to understand the concept better. You must have noticed that the Pizza shop seller makes a pizza, soft drink, and breadstick combo together. He also offers a discount to their customers who buy these combos. Do you ever think why does he do so? He thinks that customers who buy pizza also buy soft drinks and breadsticks. However, by making combos, he makes it easy for the customers. At the same time, he also increases his sales performance.

Similarly, you go to Big Bazar, and you will find biscuits, chips, and Chocolate bundled together. It shows that the shopkeeper makes it comfortable for the customers to buy these products in the same place.

What is Apriori Algorithm?

Apriori algorithm refers to an algorithm that is used in mining frequent products sets and relevant association rules. Generally, the apriori algorithm operates on a database containing a huge number of transactions. For example, the items customers but at a Big Bazar.

Apriori algorithm helps the customers to buy their products with ease and increases the sales performance of the particular store.

Components of Apriori algorithm

The given three components comprise the apriori algorithm.

Support

Confidence

Lift

Support

Support refers to the default popularity of any product. You find the support as a quotient of the division of the number of transactions comprising that product by the total number of transactions. Hence, we get

Support (Biscuits) = (Transactions relating biscuits) / (Total transactions)

= 400/4000 = 10 percent.

Confidence

Confidence refers to the possibility that the customers bought both biscuits and chocolates together. So, you need to divide the number of transactions that comprise both biscuits and chocolates by the total number of transactions to get the confidence.

Hence,

Confidence = (Transactions relating both biscuits and Chocolate) / (Total transactions involving Biscuits)

= 200/400

= 50 percent.

It means that 50 percent of customers who bought biscuits bought chocolates also.

Lift

Consider the above example; lift refers to the increase in the ratio of the sale of chocolates when you sell biscuits. The mathematical equations of lift are given below.

Lift = (Confidence (Biscuits - chocolates)/ (Support (Biscuits)

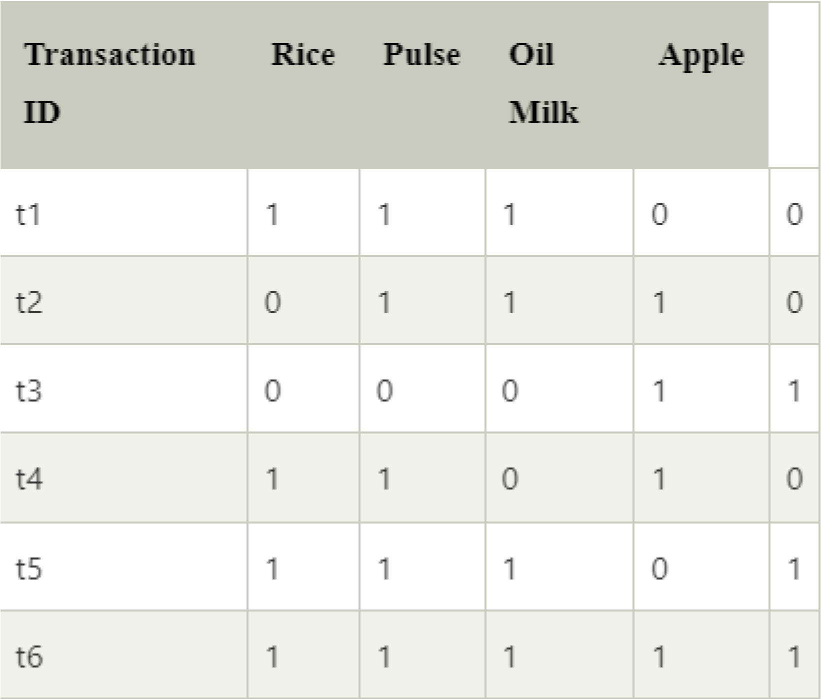
= 50/10 = 5

It means that the probability of people buying both biscuits and chocolates together is five times more than that of purchasing the biscuits alone. If the lift value is below one, it requires that the people are unlikely to buy both the items together. Larger the value, the better is the combination.

How does the Apriori Algorithm work in Data Mining?

We will understand this algorithm with the help of an example

Consider a Big Bazar scenario where the product set is P = {Rice, Pulse, Oil, Milk, Apple}. The database comprises six transactions where 1 represents the presence of the product and 0 represents the absence of the product.



Transaction ID	Rice	Pulse	Oil Milk	Apple	
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1
t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

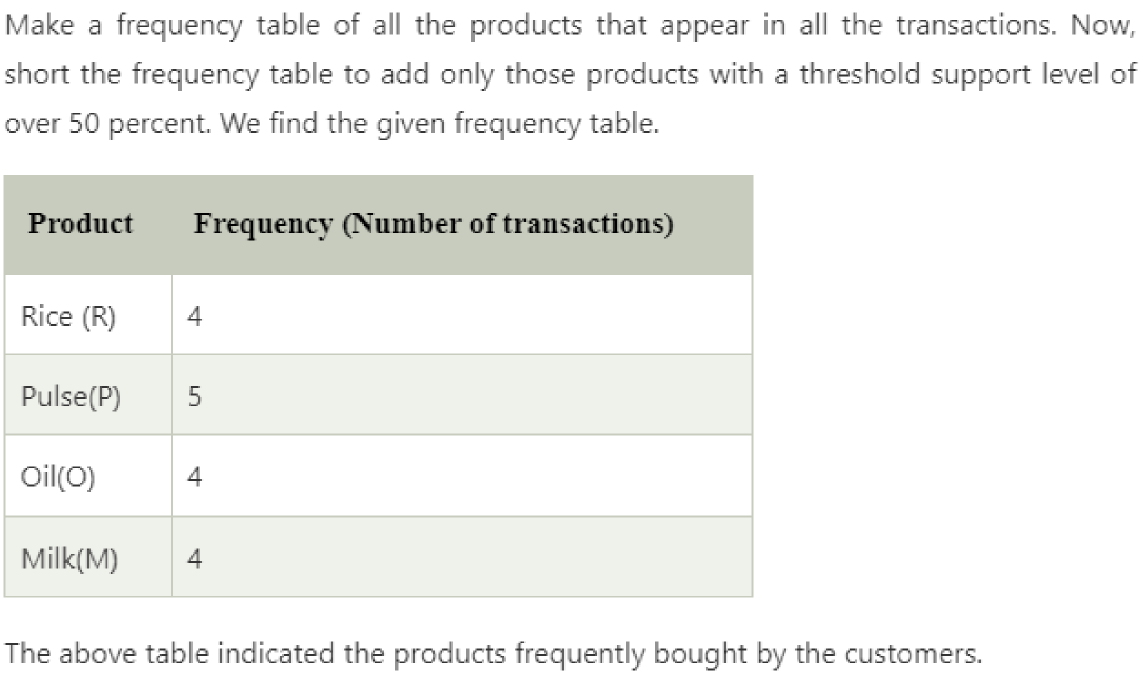
he Apriori Algorithm makes the given assumptions

All subsets of a frequent itemset must be frequent.

The subsets of an infrequent item set must be infrequent.

Fix a threshold support level. In our case, we have fixed it at 50 percent.

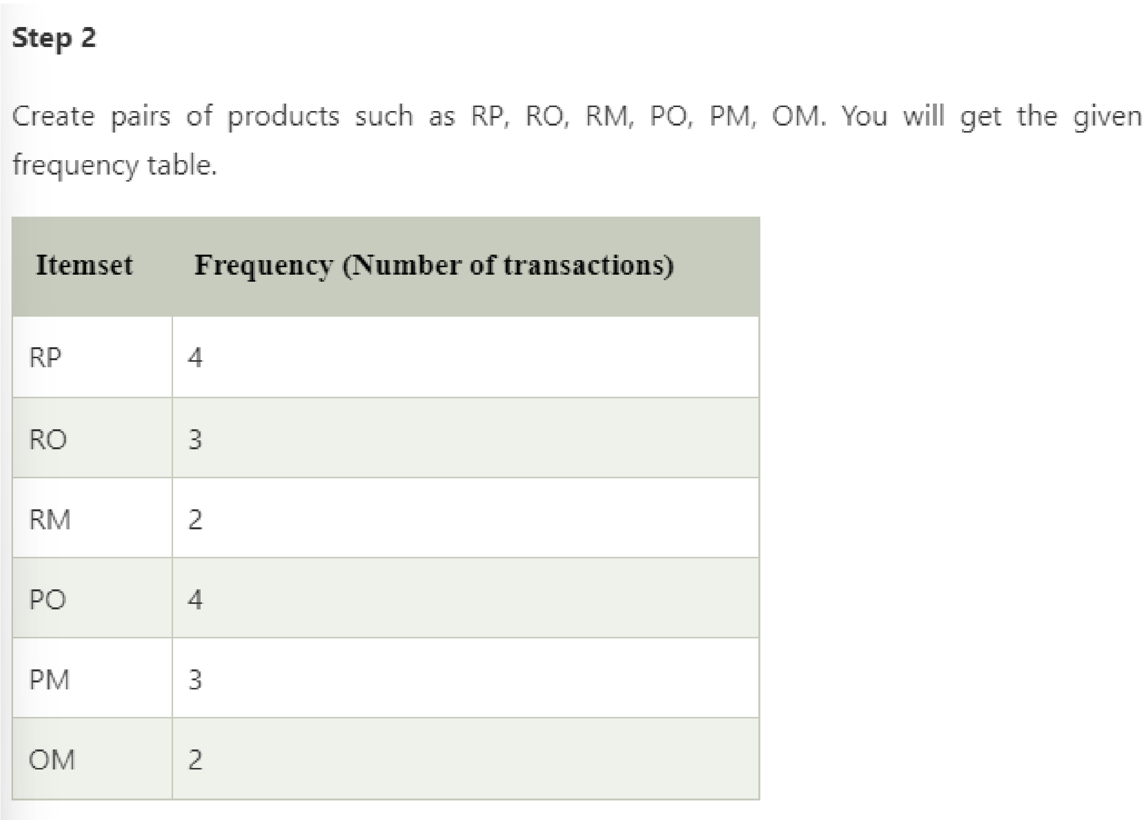
Step 1



Make a frequency table of all the products that appear in all the transactions. Now, short the frequency table to add only those products with a threshold support level of over 50 percent. We find the given frequency table.

Product	Frequency (Number of transactions)
Rice (R)	4
Pulse(P)	5
Oil(O)	4
Milk(M)	4

The above table indicated the products frequently bought by the customers.



Step 2

Create pairs of products such as RP, RO, RM, PO, PM, OM. You will get the given frequency table.

Itemset	Frequency (Number of transactions)
RP	4
RO	3
RM	2
PO	4
PM	3
OM	2

Step 3

Implementing the same threshold support of 50 percent and consider the products that are more than 50 percent. In our case, it is more than 3

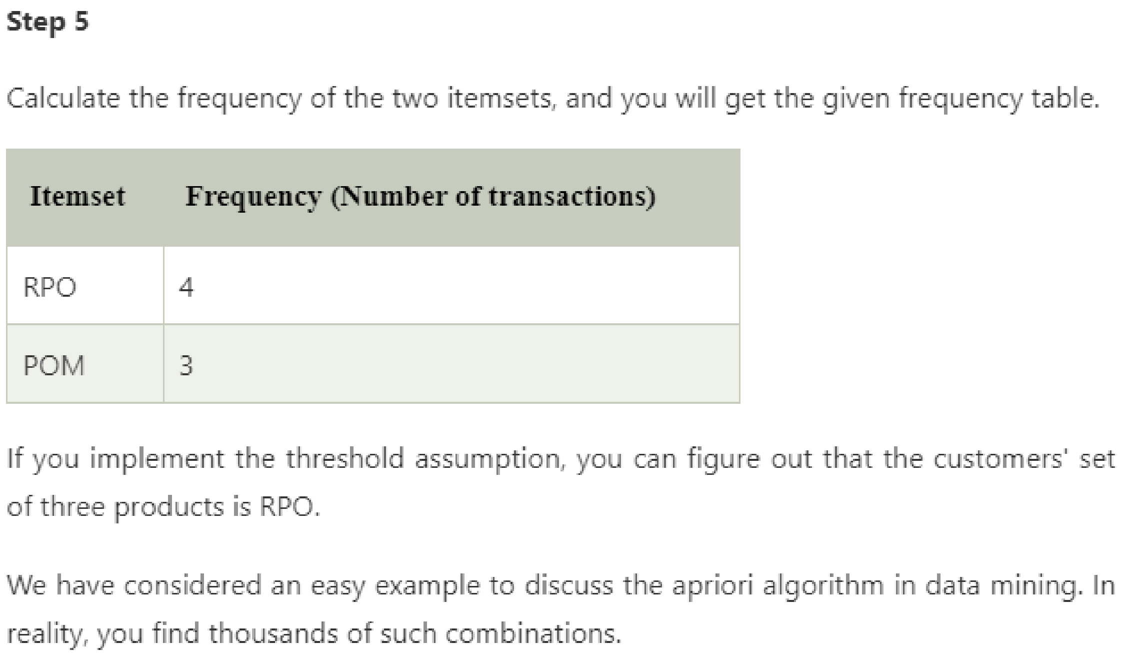
Thus, we get RP, RO, PO, and PM

Step 4

Now, look for a set of three products that the customers buy together. We get the given combination.

1.RP and RO give RPO

2.PO and PM give POM



Step 5

Calculate the frequency of the two itemsets, and you will get the given frequency table.

Itemset	Frequency (Number of transactions)
RPO	4
POM	3

If you implement the threshold assumption, you can figure out that the customers' set of three products is RPO.

We have considered an easy example to discuss the apriori algorithm in data mining. In reality, you find thousands of such combinations.

Advantages of Apriori Algorithm

It is used to calculate large itemsets.

Simple to understand and apply.

Disadvantages of Apriori Algorithms

Apriori algorithm is an expensive method to find support since the calculation has to pass through the whole database.

Sometimes, you need a huge number of candidate rules, so it becomes computationally more expensive.

Q9.Explain Eclat algorithm.

Ans: The Eclat algorithm is a frequent itemset mining technique used in machine learning and data mining to discover patterns in transactional datasets. It is particularly useful for market basket analysis, where you want to find associations between items frequently bought together.

How Eclat Algorithm Works:

The Eclat algorithm, short for “Equivalence Class Clustering and Bottom-Up Lattice Traversal” is a depth-first search-based approach to find frequent itemsets. It relies on the concept of an “equivalence class” to reduce the search space efficiently.

Here’s how it works step by step:

step1:

Transaction Database: Eclat starts with a transaction database, where each row represents a transaction, and each column represents an item. Each cell contains either a 1 (indicating the presence of an item in a transaction) or 0 (indicating absence).

step2:

Itemset Generation: Initially, Eclat creates a list of single items as 1-itemsets. It counts the support (frequency) of each item in the database by scanning it once.

step3:

Building Equivalence Classes: Eclat constructs equivalence classes by grouping transactions that share common items in their 1-itemsets. Equivalence classes reduce the number of potential itemset combinations to consider.

step4:

Recursive Search: Eclat recursively explores larger itemsets by combining smaller ones. It does this by taking the intersection of equivalence classes of items. This step is similar to the join operation in the Apriori algorithm.

step5:

Pruning: Eclat prunes infrequent itemsets at each step to reduce the search space, just like Apriori. If an itemset’s support falls below a predefined minimum support threshold, it is eliminated.

step6:

Repeat: Steps 4 and 5 are repeated iteratively to find all frequent itemsets in the dataset.

///

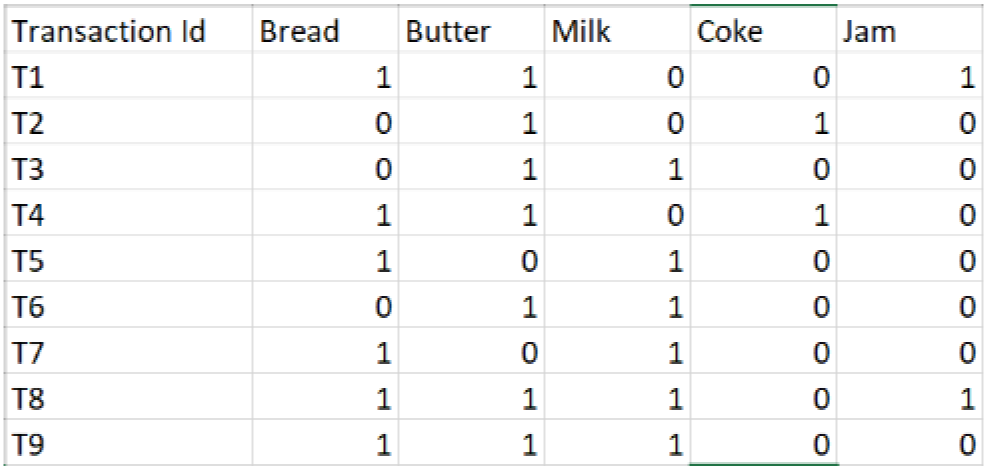
The ECLAT algorithm stands for Equivalence Class Clustering and bottom-up Lattice Traversal. It is one of the popular methods of Association Rule mining. It is a more efficient and scalable version of the Apriori algorithm. While the Apriori algorithm works in a horizontal sense imitating the Breadth-First Search of a graph, the ECLAT algorithm works in a vertical manner just like the Depth-First Search of a graph. This vertical approach of the ECLAT algorithm makes it a faster algorithm than the Apriori algorithm.

How the algorithm work? :

The basic idea is to use Transaction Id Sets(tidsets) intersections to compute the support value of a candidate and avoiding the generation of subsets which do not exist in the prefix tree. In the first call of the function, all single items are used along with their tidsets. Then the function is called recursively and in each recursive call, each item-tidset pair is verified and combined with other item-tidset pairs. This process is continued until no candidate item-tidset pairs can be combined.

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

Consider the following transactions record:-

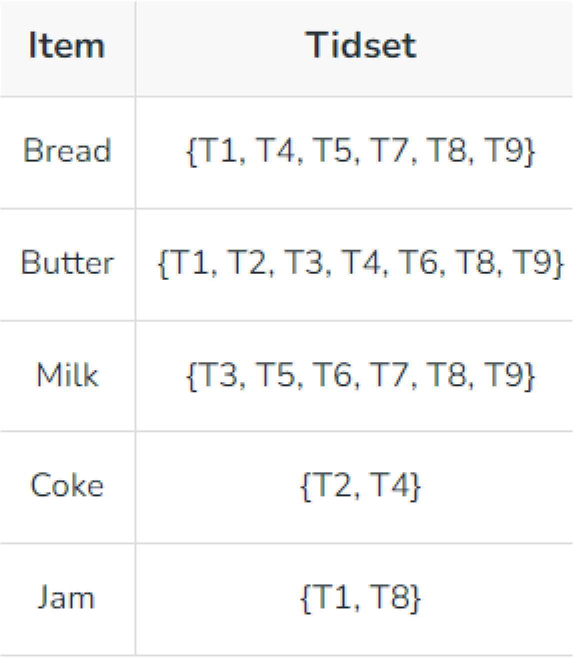


Transaction Id	Bread	Butter	Milk	Coke	Jam
T1	1	1	0	0	1
T2	0	1	0	1	0
T3	0	1	1	0	0
T4	1	1	0	1	0
T5	1	0	1	0	0
T6	0	1	1	0	0
T7	1	0	1	0	0
T8	1	1	1	0	1
T9	1	1	1	0	0

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:-

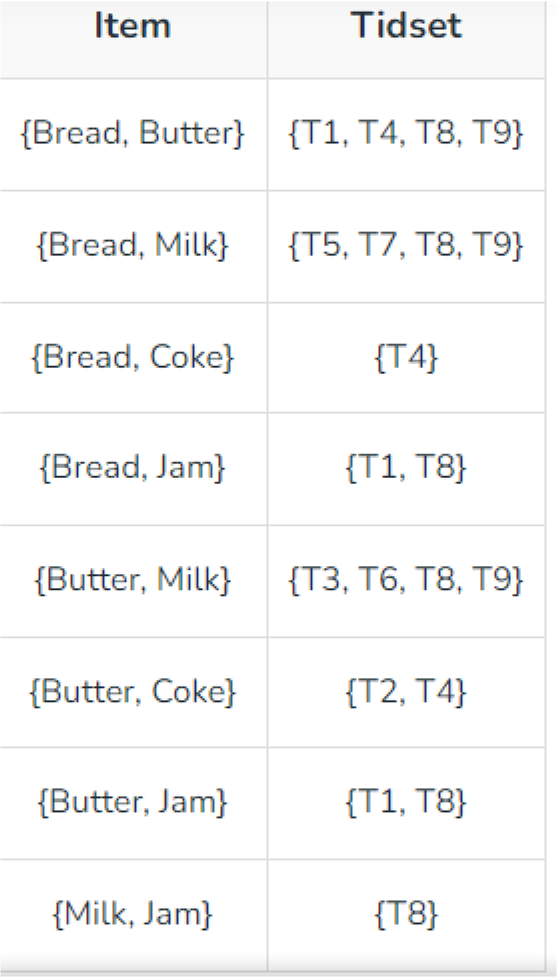
k = 1, minimum support = 2



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}

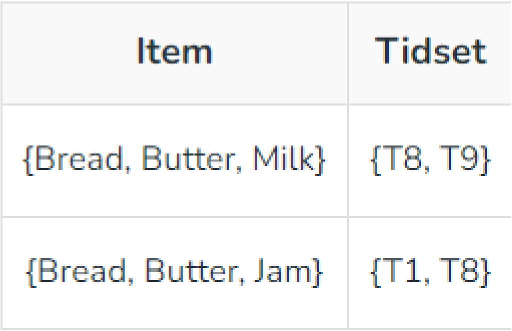
We now recursively call the function till no more item-tidset pairs can be combined:-

k = 2



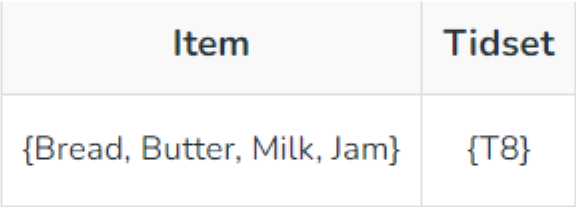
Item	Tidset
{Bread, Butter}	{T1, T4, T8, T9}
{Bread, Milk}	{T5, T7, T8, T9}
{Bread, Coke}	{T4}
{Bread, Jam}	{T1, T8}
{Butter, Milk}	{T3, T6, T8, T9}
{Butter, Coke}	{T2, T4}
{Butter, Jam}	{T1, T8}
{Milk, Jam}	{T8}

k = 3



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

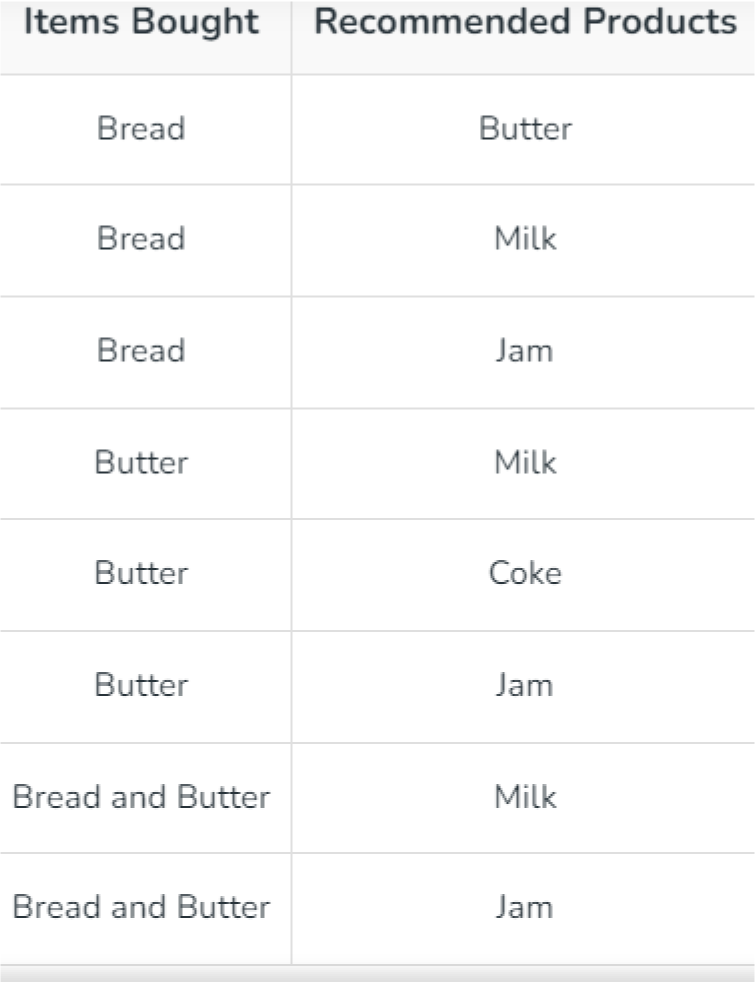
k = 4



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
Bread and Butter	Jam

Advantages over Apriori algorithm:-

Memory Requirements: Since the ECLAT algorithm uses a Depth-First Search approach, it uses less memory than Apriori algorithm.

Speed: The ECLAT algorithm is typically faster than the Apriori algorithm.

Number of Computations: The ECLAT algorithm does not involve the repeated scanning of the data to compute the individual support values.

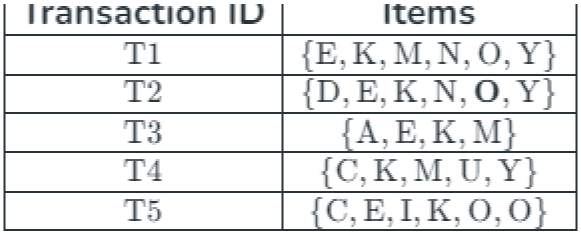
Q10.10. Explain F-P Growth algorithm.

Ans: The two primary drawbacks of the Apriori Algorithm are:

At each step, candidate sets have to be built.

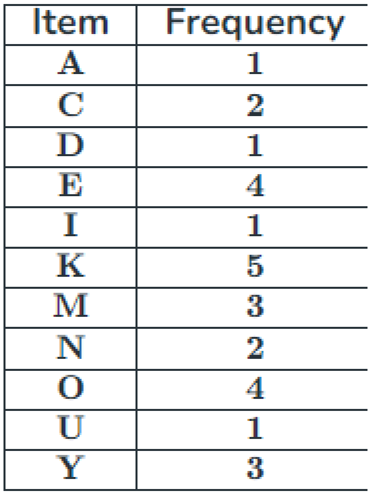
To build the candidate sets, the algorithm has to repeatedly scan the database.

These two properties inevitably make the algorithm slower. To overcome these redundant steps, a new association-rule mining algorithm was developed named Frequent Pattern Growth Algorithm. It overcomes the disadvantages of the Apriori algorithm by storing all the transactions in a Trie Data Structure. Consider the following data:-



Transaction ID	Items
T1	{E, K, M, N, O, Y}
T2	{D, E, K, N, O, Y}
T3	{A, E, K, M}
T4	{C, K, M, U, Y}
T5	{C, E, I, K, O, O}

The above-given data is a hypothetical dataset of transactions with each letter representing an item. The frequency of each individual item is computed:-

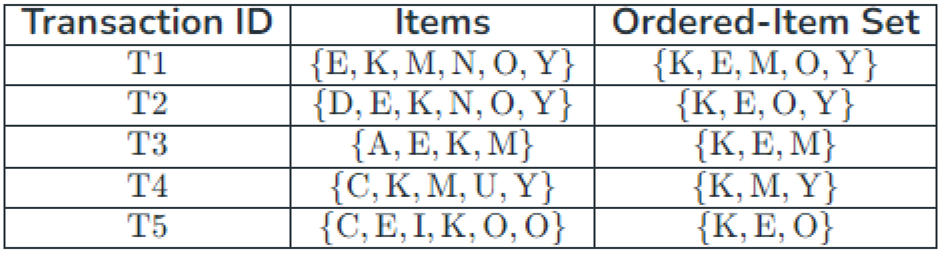


Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	4
U	1
Y	3

Let the minimum support be 3. A Frequent Pattern set is built which will contain all the elements whose frequency is greater than or equal to the minimum support. These elements are stored in descending order of their respective frequencies. After insertion of the relevant items, the set L looks like this:-

L = {K : 5, E : 4, M : 3, O : 4, Y : 3}

Now, for each transaction, the respective Ordered-Item set is built. It is done by iterating the Frequent Pattern set and checking if the current item is contained in the transaction in question. If the current item is contained, the item is inserted in the Ordered-Item set for the current transaction. The following table is built for all the transactions:

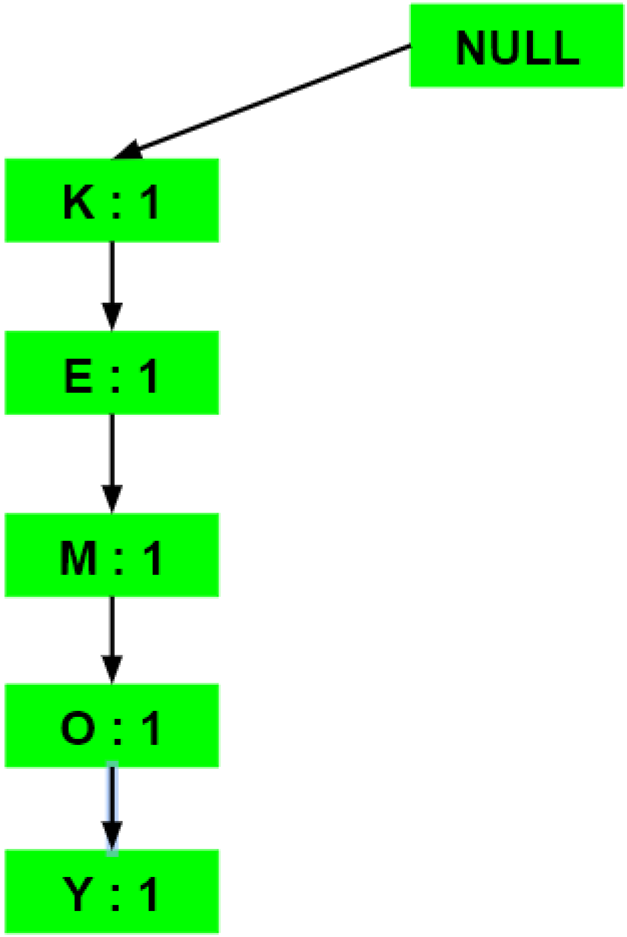


Transaction ID	Items	Ordered-Item Set
T1	{E, K, M, N, O, Y}	{K, E, M, O, Y}
T2	{D, E, K, N, O, Y}	{K, E, O, Y}
T3	{A, E, K, M}	{K, E, M}
T4	{C, K, M, U, Y}	{K, M, Y}
T5	{C, E, I, K, O, O}	{K, E, O}

Now, all the Ordered-Item sets are inserted into a Trie Data Structure.

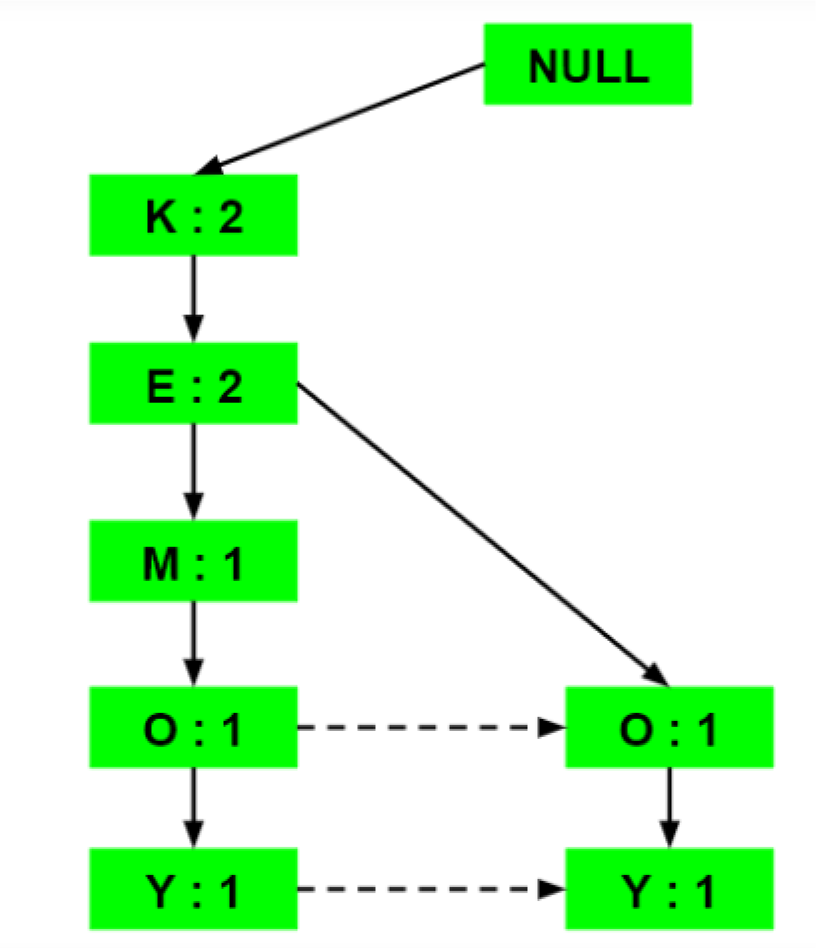
a) Inserting the set {K, E, M, O, Y}:

Here, all the items are simply linked one after the other in the order of occurrence in the set and initialize the support count for each item as 1.



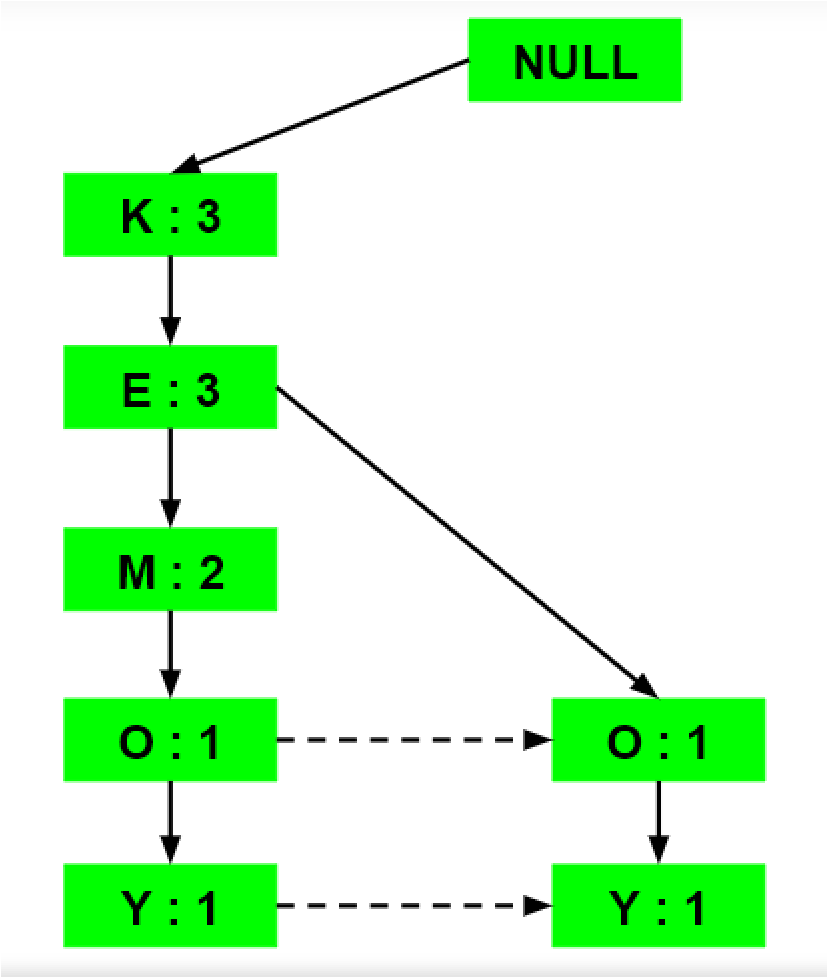
b) Inserting the set {K, E, O, Y}:

Till the insertion of the elements K and E, simply the support count is increased by 1. On inserting O we can see that there is no direct link between E and O, therefore a new node for the item O is initialized with the support count as 1 and item E is linked to this new node. On inserting Y, we first initialize a new node for the item Y with support count as 1 and link the new node of O with the new node of Y.



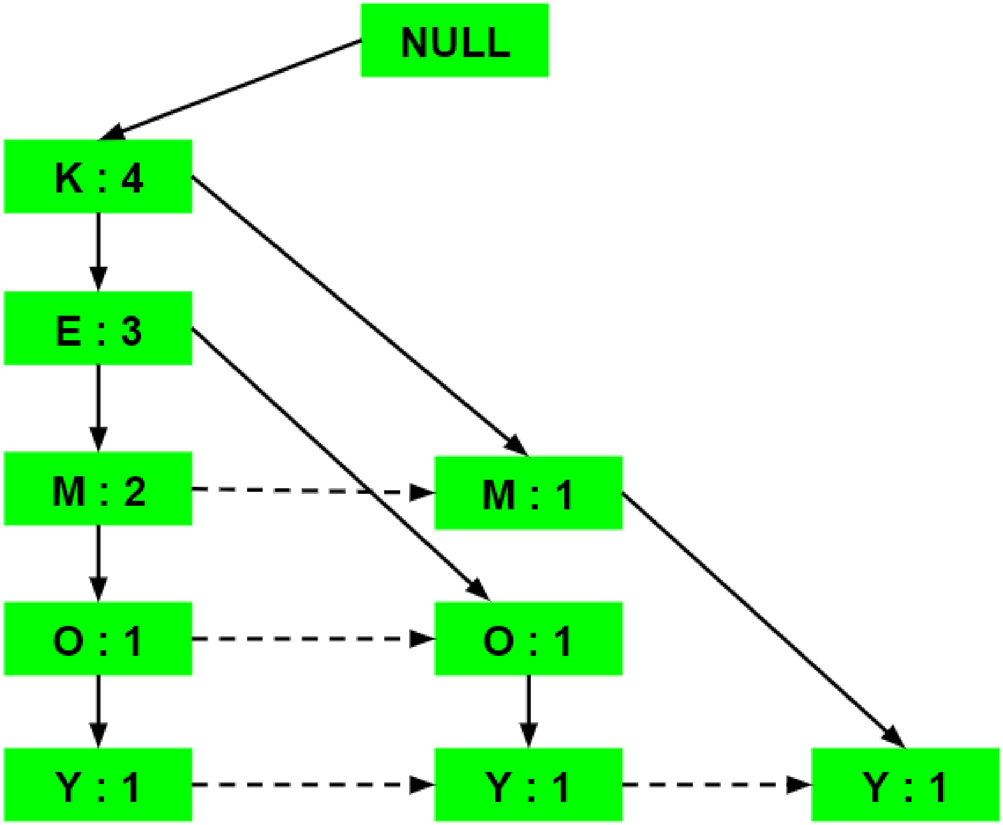
c) Inserting the set {K, E, M}:

Here simply the support count of each element is increased by 1.



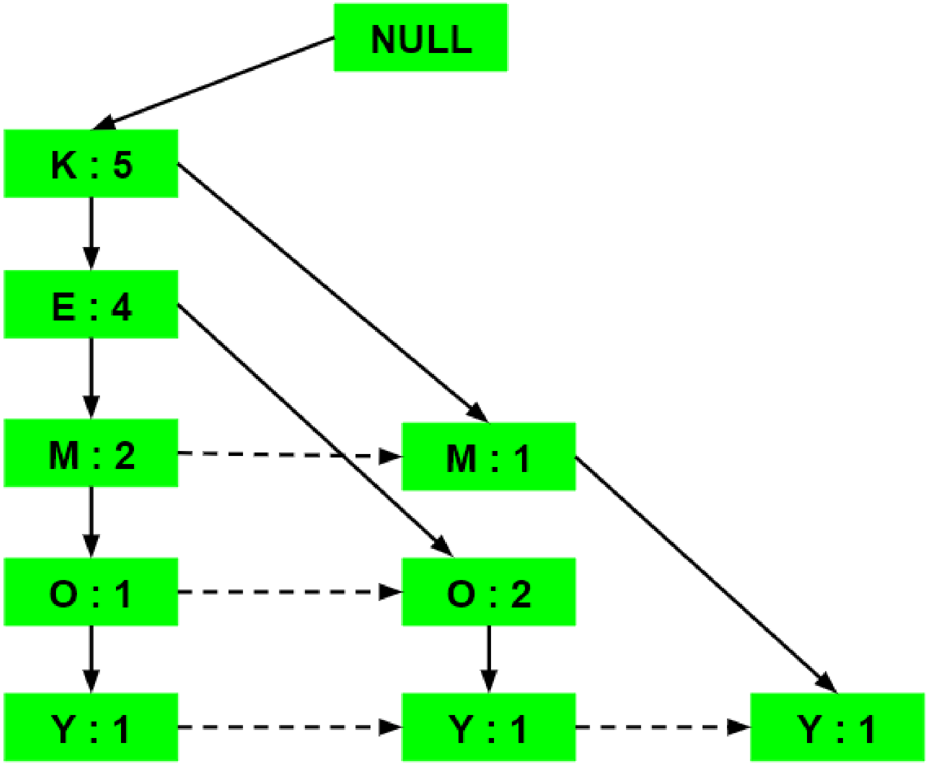
d) Inserting the set {K, M, Y}:

Similar to step b), first the support count of K is increased, then new nodes for M and Y are initialized and linked accordingly.



e) Inserting the set {K, E, O}:

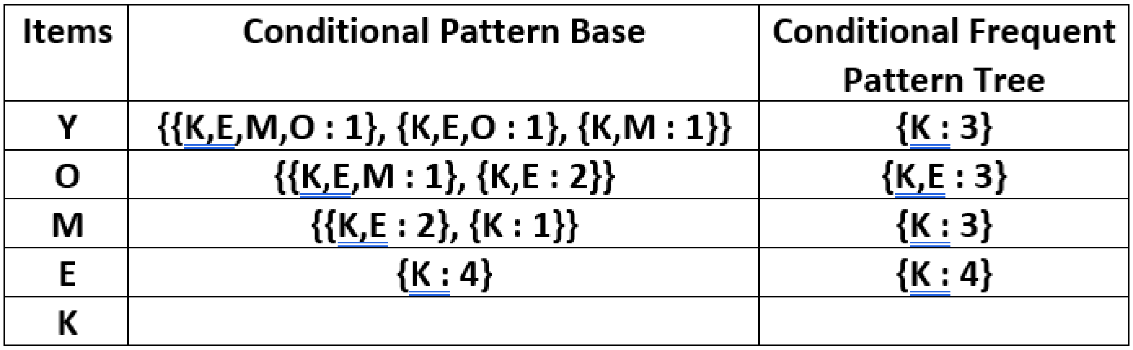
Here simply the support counts of the respective elements are increased. Note that the support count of the new node of item O is increased.



Now, for each item, the Conditional Pattern Base is computed which is path labels of all the paths which lead to any node of the given item in the frequent-pattern tree. Note that the items in the below table are arranged in the ascending order of their frequencies.

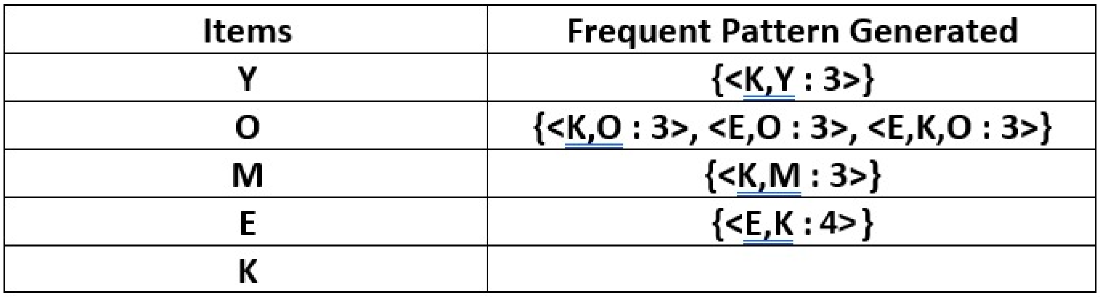
Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Now for each item, the Conditional Frequent Pattern Tree is built. It is done by taking the set of elements that is common in all the paths in the Conditional Pattern Base of that item and calculating its support count by summing the support counts of all the paths in the Conditional Pattern Base.



Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

From the Conditional Frequent Pattern tree, the Frequent Pattern rules are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding to the item as given in the below table.



Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 4>}
K	

For each row, two types of association rules can be inferred for example for the first row which contains the element, the rules K -> Y and Y -> K can be inferred. To determine the valid rule, the confidence of both the rules is calculated and the one with confidence greater than or equal to the minimum confidence value is retained.