Apriori Algorithm

The Apriori Algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contains transactions. With the help of these association rules, it determines how strongly or how weakly two objects are connected.

The algorithm uses breadth-first search and Hash Tree to calculate the itemset associations efficiently. It is iterative process for finding the frequent itemsets from the large dataset.

The algorithm was given by the R. Agrawal and Srikant in 1994. It is mainly used for Market Basket Analysis and helps to find those products that can be bought together.

Steps :-

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

Step 2 :- Take all the subsets in transactions having higher support than minimum support.

Step 3 :- Generate association rules, 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.

Example :- Find frequent itemset and generate association rules using Apriori Algorithm.

TID	ITEMSETS
T_1	A, B
T_2	B, D
T_3	B, C
T_4	A, B, D
T_5	A, C
T_6	B, C
T_7	A, C
T_8	A, B, C, E
T_9	A, B, C

Given that

Minimum Support = 2

Minimum Confidence = 50%.

Step 1:- Calculating C1 and L1:

- In the first step, we will create a table that contains (the frequency of each itemset individually in dataset) of each itemset in the given dataset. This table is called the Candidate Set or C1.

C1	
Itemset	Support_Count
A	6
B	7
C	5
D	2
E	1

- Now, we will discard all elements that have smaller support count than minimum support (i.e. 2 in our example). It will give us the table for frequent itemset L1. In our example, E will be removed as it has only 1 support count.

L1	
Itemset	Support_Count
A	6
B	7
C	5
D	2

Step 2:- Candidate Generation C2, and L2:-

- In this step, we will generate C2, with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.

- After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e. how many times these pairs have occured together in the given dataset. So, we will get the table C2.

C2	
Itemset	Support_Count
{A, B}	4
{A, C}	4
{A, D}	1
{B, C}	4
{B, D}	2
{C, D}	0

- Again, we need to compare the C2 support count with the minimum support count and discard those itemsets which have less support count than the threshold. It will give us table L2.

L2	
Itemset	Support_Count
{A, B}	4
{A, C}	4
{B, C}	4
{B, D}	2

Step 3 :- Candidate Generation C3, and L3 :-

- For table C3, we will repeate the same process, but now we will form the table with three itemsets together.

C3	
Itemset	Support_Count
{A, B, C}	2
{B, C, D}	1
{A, C, D}	0
{A, B, D}	0

- Now for L3 table, as we can see, only one combination of itemset left that has support count equal to the minimum support.

L3	
Itemset	Support_Count
{A, B, C}	2

Step 4 :- Finding the association rules for the subsets :-