

# Apriori Algorithm: Finding Frequent Itemsets by Confined Candidate Generation

- Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules .
- The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.
- Apriori employs an iterative approach known as a level-wise search, where  $k$ -itemsets are used to explore  $(k + 1)$ -itemsets.

- Apriori algorithm was the first algorithm that was proposed for frequent itemset mining.
- It was later improved by R Agarwal and R Srikant and came to be known as Apriori.
- This algorithm uses two steps “join” and “prune” to reduce the search space.
- It is an iterative approach to discover the most frequent itemsets.

The probability that item  $I$  is not frequent is if:

- $P(I) < \text{minimum support threshold}$ , then  $I$  is not frequent.
- $P(I+A) < \text{minimum support threshold}$ , then  $I+A$  is not frequent, where  $A$  also belongs to itemset.
- If an itemset set has value less than minimum support then all of its supersets will also fall below min support, and thus can be ignored. This property is called the **Antimonotone** property.

# The steps followed in the Apriori Algorithm of data mining are:

- **Join Step:** This step generates  $(K+1)$  itemset from  $K$ -itemsets by joining each item with itself.
- **Prune Step:** This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

# Steps In Apriori

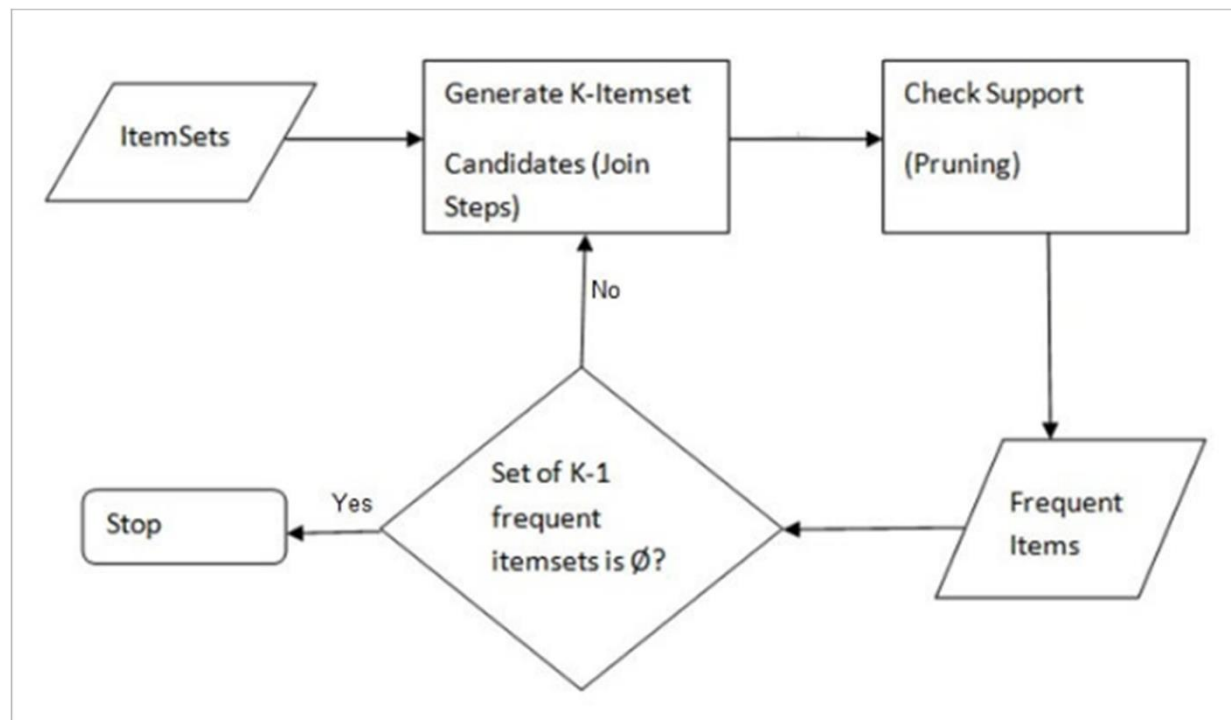
- Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database.
- This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved.
- A minimum support threshold is given in the problem or it is assumed by the user.

1. In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.
2. Let there be some minimum support,  $\text{min\_sup}$  (eg 2). The set of 1 – itemsets whose occurrence is satisfying the min sup are determined. Only those candidates which count more than or equal to  $\text{min\_sup}$ , are taken ahead for the next iteration and the others are pruned.

3. Next, 2-itemset frequent items with min\_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.
4. The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2 –itemsets with min-sup only.



5. The next iteration will form 3 –itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2 –itemset subsets of each group fall in min\_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.
6. Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min\_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.



# 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%  $\Rightarrow 0.5 * 6 = 3 \Rightarrow$

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  $\text{min\_sup}=3$ , thus it is deleted, only I1, I2, I3, I4 meet  $\text{min\_sup}$  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.

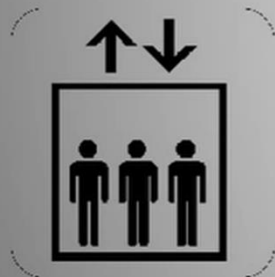
**TABLE-4**

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



# LIFT

$$\text{lift}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) * \text{supp}(Y)}$$



$> 1$

Y is likely to be bought with X



# CONVICTION



$$\text{conv}(X \rightarrow Y) = \frac{1 - \text{supp}(Y)}{1 - \text{conf}(X \rightarrow Y)}$$

{onion, potato} $\Rightarrow$ {burger}



32%