Assignment No. 5 Apriori

Name : Sahil Shendkar

PRN : 122B1F118

**AIM:**

Assignment of understanding and applying the **Apriori algorithm** for discovering frequent itemsets and learning association rules in datasets using machine learning techniques.

**PREREQUISITE:**

* Basic understanding of **Probability and Set Theory**
* Knowledge of **Python programming** and **Data Mining concepts**

**THEORY:**

**Introduction to Association Rule Learning**

**Association rule learning** is a rule-based machine learning method for discovering interesting relations between variables in large datasets. It is commonly used in **market basket analysis**, where the goal is to find associations between items bought together.

**Apriori Algorithm – Overview**

The **Apriori algorithm** is a classic algorithm in data mining for learning **association rules**. It is named “Apriori” because it uses a prior knowledge of frequent itemset properties. The algorithm works by identifying the frequent individual items in the dataset and extending them to larger and larger itemsets as long as those itemsets appear sufficiently often in the dataset.

It is a **bottom-up approach**, where:

* It starts from single items (1-itemsets),
* Builds larger itemsets (k-itemsets),
* And prunes those that don’t satisfy the **minimum support threshold**.

**Key Concepts**

**1. Itemset**

An **itemset** is simply a collection of one or more items. For example, in a supermarket: {Milk, Bread} is an itemset.

**2. Support**

The **support** of an itemset is the proportion of transactions in the dataset that contain the itemset.  
**Support(A)** = (Number of transactions containing A) / (Total transactions)

**3. Confidence**

The **confidence** of a rule A → B indicates how often B appears in transactions that contain A.  
**Confidence(A→B)** = Support(A ∪ B) / Support(A)

**4. Lift**

The **lift** of a rule compares the observed support with what would be expected if A and B were independent.  
**Lift(A→B)** = Confidence(A→B) / Support(B)

* Lift > 1: Positive correlation
* Lift = 1: No correlation
* Lift < 1: Negative correlation

**Working of Apriori Algorithm**

* **Generate 1-itemsets** from the transaction database that meet the minimum support.
* **Generate candidate 2-itemsets** using the frequent 1-itemsets.
* **Prune** itemsets that have infrequent subsets (Apriori Property).
* **Repeat** until no more frequent itemsets can be generated.
* **Generate association rules** from the frequent itemsets that meet the confidence threshold.

**Professional Example: Market Basket Analysis**

Suppose we analyze a grocery store’s transaction log:

|  |  |
| --- | --- |
| **Transaction ID** | **Items** |
| T1 | Milk, Bread, Butter |
| T2 | Bread, Butter |
| T3 | Milk, Bread |
| T4 | Milk, Bread, Butter, Eggs |
| T5 | Bread, Butter |

* **Step 1:** Count itemsets like {Milk}, {Bread}, etc.
* **Step 2:** Build combinations like {Milk, Bread}, {Bread, Butter}, etc.
* **Step 3:** Apply support and confidence thresholds to extract rules like:  
  **“If Bread, then Butter”** with a confidence of 80%.

**Advantages:**

* Easy to implement and interpret
* Effective for datasets with large transactions and discrete values
* Generates all possible frequent itemsets and rules

**Limitations:**

* May generate **too many rules**, some of which may be redundant
* High **computational cost** with large datasets
* Requires **setting thresholds** (support/confidence) carefully

**CONCLUSION:**

The **Apriori algorithm** remains one of the foundational algorithms in data mining, widely used in practical scenarios like market basket analysis, recommendation systems, and inventory planning. It helps uncover hidden patterns and associations in large datasets. Understanding Apriori provides a strong base for learning more advanced machine learning concepts like frequent pattern trees (FP-Growth) and sequential pattern mining.