**Name** - Omkar Keny

**Div** - TY9 - B

**Roll no** - 26

### *DWM EXPERIMENT NO: 5*

**Aim** :- Implementation of Association Rule Mining algorithm (Apriori)

**Introduction** :-

- Association Rule Mining: Discovers relationships between items in a dataset.
- Apriori Algorithm: A popular algorithm for association rule mining based on frequent itemsets.
- Frequent Itemsets: Sets of items frequently occurring together in a dataset.

Two-Step Process:

- Frequent Itemset Generation: Finds itemsets meeting a minimum support threshold.
- Rule Generation: Creates association rules from frequent itemsets (e.g., "if milk, then bread").

**Procedure** :-

- Import the necessary libraries:
- Define a function to get frequent itemsets
- Define a function to generate candidate itemsets
- Define the Apriori algorithm
- Use the Apriori algorithm to find frequent itemsets

```
from itertools import combinations

def get_frequent_itemsets(transactions, min_support):
    itemsets = {}
    for transaction in transactions:
        for item in transaction:
            if item in itemsets:
                itemsets[item] += 1
            else:
                itemsets[item] = 1

    frequent_itemsets = {item: support for item, support in itemsets.items() if support >=
    return frequent_itemsets
```

```
def get_candidate_itemsets(frequent_itemsets, k):
    candidates = []
    frequent_items = list(frequent_itemsets.keys())
    for combination in combinations(frequent_items, k):
        candidates.append(combination)
    return candidates

def apriori(transactions, min_support):
    k = 1
    frequent_itemsets = get_frequent_itemsets(transactions, min_support)
    all_frequent_itemsets = [frequent_itemsets]

    while frequent_itemsets:
        k += 1
        candidates = get_candidate_itemsets(frequent_itemsets, k)
        candidate_supports = {candidate: 0 for candidate in candidates}

        for transaction in transactions:
            for candidate in candidates:
                if set(candidate).issubset(set(transaction)):
                    candidate_supports[candidate] += 1

        frequent_itemsets = {itemset: support for itemset, support in candidate_supports.i
        if frequent_itemsets:
            all_frequent_itemsets.append(frequent_itemsets)

    return all_frequent_itemsets

transactions = [
    ['milk', 'bread', 'butter'],
    ['bread', 'butter','jam'],
    ['milk', 'eggs'],
    ['milk', 'butter'],
    ['bread', 'butter'],
    ['milk', 'bread', 'butter']
]

min_support = 2
frequent_itemsets = apriori(transactions, min_support)
```

```
[{'milk': 4, 'bread': 4, 'butter': 5}, {('milk', 'bread'): 2, ('milk', 'butter'): 3, ('b
```

**1) What is the Apriori algorithm in Association Rule Mining?** Apriori is a classic algorithm used in Association Rule Mining, which is a technique used in data mining to identify interesting relationships (or associations) among a set of items in large datasets, such as in market basket analysis. The goal is to find frequent itemsets in a dataset and then derive association rules from these itemsets.

Frequent itemsets: These are groups of items that appear together in transactions with a frequency greater than or equal to a predefined threshold (called minimum support). Association rules: These are rules of the form A → B, which means "if A occurs, then B is likely to occur as well." These rules are generated from frequent itemsets. The Apriori algorithm works in two main steps:

Find frequent itemsets: The algorithm scans the dataset to find all itemsets that meet the minimum support threshold. It starts with single items (1-itemsets) and gradually combines them into larger itemsets (2-itemsets, 3-itemsets, etc.) while pruning those that do not meet the support requirement. Generate association rules: Once the frequent itemsets are found, Apriori generates rules based on confidence and lift, which help to evaluate the strength and significance of the rules. Apriori uses a bottom-up approach, iterating over itemsets of increasing size and ensuring that any subset of a frequent itemset is also frequent. This helps reduce the number of candidate itemsets and makes the algorithm efficient.

**2) What is the significance of support, confidence, and lift in Apriori?** These three metrics are critical for evaluating the strength and relevance of the association rules generated by the Apriori algorithm. Let's elaborate on each:

Definition: Support refers to the frequency of an itemset appearing in the dataset. It’s calculated as the ratio of transactions that contain the itemset to the total number of transactions.

Significance:

Support helps identify the most popular items or combinations of items in the dataset. High support means the itemset occurs frequently, but it doesn't necessarily indicate that the items have a strong relationship. Support = (Number of transactions containing itemset) / (Total number of transactions) Confidence: Definition: Confidence is a measure of the likelihood that an item Y appears in a transaction given that item X is already present. It’s essentially the conditional probability of Y occurring given that X occurs.

Significance:

Confidence tells us how reliable a rule is. A high confidence value means that the rule has a high probability of being true. It helps in filtering out weak rules that don’t show a strong relationship between items. Confidence = (Number of transactions containing both X and Y) / (Number of transactions containing X) Lift: Definition: Lift is a measure of the strength of an association rule relative to how frequently the items X and Y occur independently. It compares the observed frequency of X and Y appearing together with the expected frequency if X and Y were independent.

Significance:

Lift helps to evaluate whether the occurrence of X and Y together is more significant than random chance. If the lift is greater than 1, it means X and Y appear together more often than expected (indicating a strong association). If the lift is less than 1, the items tend to appear independently of each other. Lift = Confidence(X → Y) / (Support(Y))

**Conclusion** :- The Apriori algorithm effectively identifies frequent itemsets and generates association rules from transactional data. By using a minimum support threshold, it efficiently finds significant relationships between items. Implementing it in Python allows for a structured approach, uncovering hidden patterns and leading to valuable insights for decision-making in various domains. The Apriori algorithm is a fundamental and practical technique for association rule mining, offering a powerful solution for discovering knowledge from large datasets.

github: https://github.com/omk279/DWM