IMPLEMENTATION OF APRIORI

Batch-1, Module-1, Target-3.

*B.Durga Reddy N.Laxmi Prathyusha*

*VFSTR University VFSTR University*

*Department of CSE Department of CSE*

211FA04377211FA04442

*G.Lasya Priya B.Chaitanya Sai*

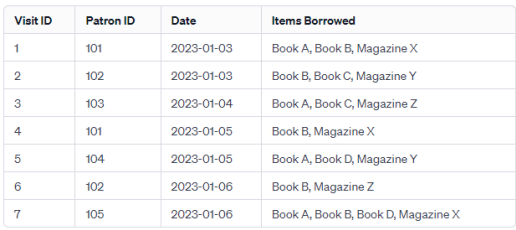
*VFSTR University VFSTR University*

*Department of CSE Department of CSE*

211FA04477 211FA04496

**Problem Statement:**

Consider the following dataset which represents a library checkout history, where each row corresponds to a library patron's visit and the items they borrowed during that visit.



1.What is the Apriori algorithm, and how does it work in the context of mining frequent item sets?

2. Use a support threshold of 30% to identify frequent item sets in the dataset. Provide examples of these frequent item sets.

3. Apply the Apriori algorithm again with a lower support threshold of 20%. Compare the results with the previous analysis and discuss the impact of changing the threshold.

4. Explain the concept of association rules and how they are generated from frequent item sets.

5. Generate association rules with a minimum confidence of 70% from the frequent item sets obtained with a 30% support threshold. Provide examples of these rules.

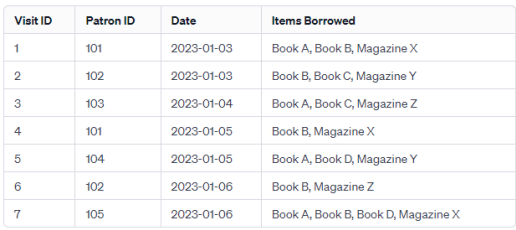
6.Discuss the practical implications of the association rules you've generated for the library, including potential strategies for recommending books and magazines to patrons.

**1.Apriori algorithm:** It is used for finding frequent itemsets in a dataset for boolean association rule.

We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets. To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called Apriori property which helps by reducing the search space.

**Principle of Apriori Algorithm:**

The core idea is based on the Apriori property, which states that "all subsets of a frequent itemset must also be frequent". This property is used to reduce the search space.



Association Rules:  
x->y:  
Support= number of items containing a and b/total number of items  
Confidence=number of items containing a and b/number of items containing a

2.Support threshold = 30%

|  |  |
| --- | --- |
| **1-item set** | **Frequency** |
| Book A | 4 |
| Book B | 5 |
| Book C | 2 |
| Book D | 2 |
| Magazine X | 3 |
| Magazine Y | 2 |
| Magazine Z | 2 |

|  |  |
| --- | --- |
| **Frequent 1-item set** | **Frequency** |
| Book A | 4 |
| Book B | 5 |
| Magazine X | 3 |

|  |  |
| --- | --- |
| **2-item set** | **Frequency** |
| Book A, Book B | 2 |
| Book A, Magazine X | 2 |
| Book B, Magazine X | 3 |

|  |  |
| --- | --- |
| **Frequent 2-item set** | **Frequency** |
| Book B,Magazine X | 3 |

3.Support threshold = 20%

|  |  |
| --- | --- |
| **1-item set** | **Frequency** |
| Book A | 4 |
| Book B | 5 |
| Book C | 2 |
| Book D | 2 |
| Magazine X | 3 |
| Magazine Y | 2 |
| Magazine Z | 2 |

|  |  |
| --- | --- |
| **Frequent 1-item set** | **Frequency** |
| Book A | 4 |
| Book B | 5 |
| Book C | 2 |
| Book D | 2 |
| Magazine X | 3 |
| Magazine Y | 2 |
| Magazine Z | 2 |

|  |  |
| --- | --- |
| **2-item set** | **Frequency** |
| Book A, Book B | 2 |
| Book A, Book C | 1 |
| Book A, Book D | 2 |
| Book A, Magazine X | 2 |
| Book A, Magazine Y | 1 |
| Book A, Magazine Z | 1 |
| Book B, Book C | 1 |
| Book B, Book D | 1 |
| Book B, Magazine X | 3 |
| Book B, Magazine Y | 1 |
| Book B, Mgazine Z | 1 |
| Book C, Book D | 0 |
| Book C, Magazine X | 0 |
| Book C, Magazine Y | 1 |
| Book C, Magazine Z | 1 |
| Book D, Magazine X | 1 |
| Book D, Magazine Y | 1 |
| Book D, Magazine Z | 0 |
| Magazine X, Magazine Y | 0 |
| Magazine X, Magazine Z | 0 |
| Magazine Y, Magazine Z | 0 |

|  |  |
| --- | --- |
| **Frequent 2-itemset** | **Frequency** |
| Book A, Book B | 2 |
| Book A, Book D | 2 |
| Book A, Magazine X | 2 |
| Book B, Magazine X | 3 |

|  |  |
| --- | --- |
| **3-itemset** | **Frequency** |
| Book A, Book B, Book D | 1 |
| Book A, Book B, Magazine X | 2 |
| Book B, Book D, Magazine X | 1 |

|  |  |
| --- | --- |
| **Frequent 3-itemset** | **Frequency** |
| Book A, Book B, Magazine X | 2 |

4.Lowering the support threshold to 20% will likely result in more item sets being identified as frequent because the algorithm will be less strict about what constitutes a frequent item set. This could lead to a larger number of frequent item sets being discovered compared to the analysis with a 30% support threshold. The impact of changing the threshold would include potentially identifying more diverse patterns in the dataset, capturing less frequent but still significant associations among items.

Association rules are rules that describe the relationships between items in a dataset. They are generated from frequent item sets. An association rule typically has the form *X*→*Y*, where *X* is a set of items (called antecedent) and *Y* is another item (called consequent). These rules are generated by finding all possible combinations of antecedent and consequent from the frequent item sets and calculating their confidence.

**5) Association Rules:**

**Non-empty Subsets:**

{{ Book B} , {Magazine X} , {Book B, Magazine X}}

Minimum confidence = 70% , Support threshold = 30%

Rule-1: {Book B} -> { Magazine X}

Support(Book A)=4/7=57%>30%

Confidence=Support(Book B, Magazine X)/Support(Book A) = (3/7)/(4/7) =42% <70%

Not Valid

Rule-2: {Magazine X} -> {Book B}

Support(Magazine X)=3/7=42%>30%

Confidence = Support(Magazine X, Book B)/Support(Magazine X)=(3/7)/(3/7)=100%>70%

Valid

6)

Association rules from library data can be used to:

1. Offer personalized recommendations based on patrons' reading habits.

2. Identify cross-selling opportunities between different types of materials.

3. Create bundle deals and promotions to encourage borrowing multiple items.

4. Diversify recommendations to introduce patrons to new genres.

5. Tailor recommendations to seasonal trends.

6. Continuously improve recommendations through a feedback loop.

**Algorithm:**

1.Input Transactions: Input the number of transactions and the items in each transaction.

2.We create a dictionary to represent the transactions, where the keys are the items and the values are lists representing each transaction. If an item is present in a transaction, its corresponding value in the list is 1; otherwise, it's 0.

3.Create DataFrame: We convert the dictionary into a pandas DataFrame where each row represents a transaction and each column represents an item.

4.Set Minimum Support: You input the minimum support value, which is a threshold used to determine the minimum frequency at which an itemset must occur in the transactions to be considered frequent.

5.Generate Itemsets: We use the Apriori algorithm to find all the frequent itemsets that meet the minimum support threshold.

6.The frequent itemsets are stored in a DataFrame along with their support values.

7.Set Minimum Confidence: Input the minimum confidence value, which is a threshold used to filter association rules based on the strength of the relationship between antecedents and consequents.

8.Generate Association Rules: use the frequent itemsets generated in the previous step to generate association rules.

9.The association rules are filtered based on the minimum confidence threshold and stored in a DataFrame.

10.The resulting DataFrame contains information about the antecedents, consequents, support, and confidence of each association rule.

11.Display Results: We display the frequent itemsets and association rules that meet the specified support and confidence thresholds.

**Source Code:**

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

import pandas as pd

transactions = int(input("Enter the number of transactions: "))

data = {}

for i in range(transactions):

items = input(f"Enter the items for transaction {i+1} (separated by spaces): ").split()

for item in items:

if item not in data:

data[item] = [0]\*transactions

data[item][i] = 1

df = pd.DataFrame(data)

while True:

min\_support = float(input("Enter the minimum support value: "))

itemset = apriori(df, min\_support=min\_support, use\_colnames=True)

itemset['length'] = itemset['itemsets'].apply(lambda x: len(x))

print(itemset)

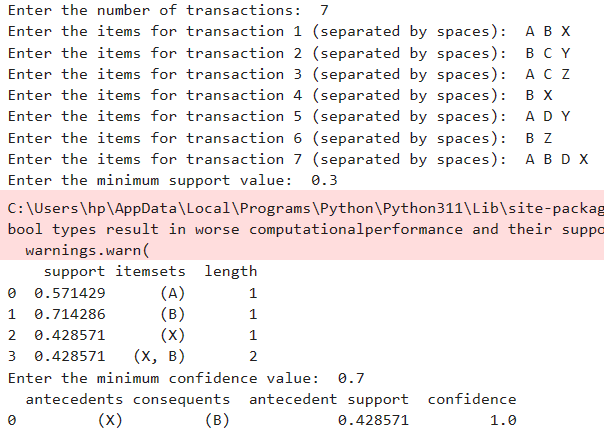
confidence = float(input("Enter the minimum confidence value: "))

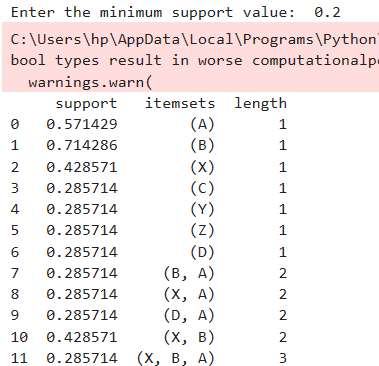
rules = association\_rules(itemset, metric="confidence", min\_threshold=confidence)

rules = rules[['antecedents', 'consequents', 'antecedent support', 'confidence']]

print(rules)

**Test Cases:**

****

****

**Conclusion:**

The Apriori algorithm is a fundamental technique in data mining used to identify frequent item sets in transactional databases. It operates by iteratively generating candidate item sets and pruning those that do not meet a minimum support threshold. By setting a support threshold of 30%, we can extract frequent item sets that occur in at least 30% of the transactions. Lowering the support threshold to 20% may reveal additional frequent item sets previously overlooked. Association rules, generated from frequent item sets, describe relationships between items and are characterized by metrics like support, confidence, and lift. With a minimum confidence of 70%, association rules provide actionable insights for libraries to optimize book and magazine recommendations, enhancing patron experience and circulation.

References:

[Apriori Algorithm – GeeksforGeeks](https://www.geeksforgeeks.org/apriori-algorithm/)

[Apriori Algorithm – Javatpoint](https://www.javatpoint.com/apriori-algorithm)