Experiment No. 7
Implement frequent pattern mining algorithm(Apriori)
Date of Performance: 10/09/25
Date of Submission:17/09/25
Name: Sumit Metkari
Div/Roll no.: 2 / 32

Aim: To implement Apriori algorithm

Objective: Develop a program to implement Apriori Algorithm on the given dataset

Theory:

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Procedure or algorithm description:

Level-wise algorithm:

a. Let k = 1

b. Generate frequent itemsets of length 1

c. Repeat until no new frequent itemsets are identified

1. Generate length (k+1) candidate itemsets from length k frequent itemsets

2. Prune candidate itemsets containing subsets of length k that are infrequent

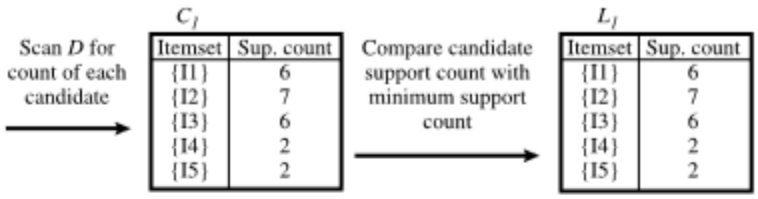
3. Count the support of each candidate by scanning the DB

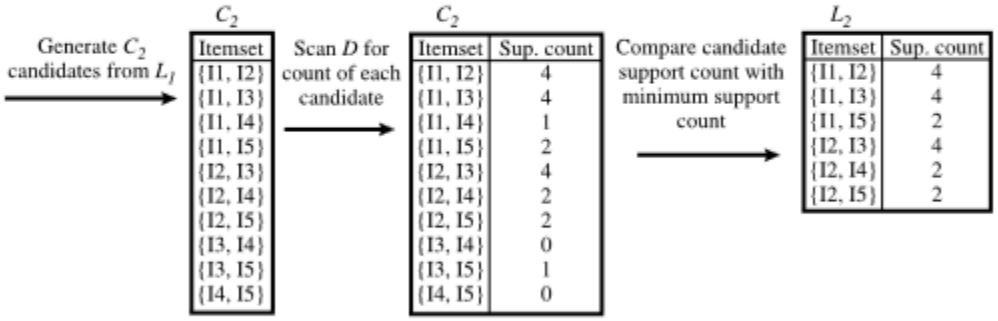
4. Eliminate candidates that are infrequent, leaving only those that are frequent

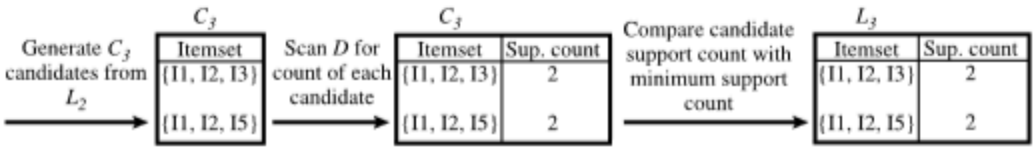
Apriori Algorithm:

Transactional data for an *AllElectronics* branch.

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3







Code and Output:

```
[1]
0s
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
[2]
0s
try:
    dataset = pd.read_csv('IRIS.csv')

    if 'Id' in dataset.columns:
        dataset = dataset.drop('Id', axis=1)
    print("Dataset loaded successfully!")
    print("Original data preview:")
    print(dataset.head())
except FileNotFoundError:
    print("Error: 'Iris.csv' not found. Please upload the dataset to your Colab environment.")
    exit()
```

```
Dataset loaded successfully!
Original data preview:
   sepal_length  sepal_width  petal_length  petal_width      species
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa
```

```
[3]
0s
discretized_dataset = dataset.copy()


if all(col in discretized_dataset.columns for col in ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']):
    numerical_cols = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
else:

    discretized_dataset.columns = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm', 'Species']
    numerical_cols = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']

labels = ['Low', 'Medium', 'High']

for col in numerical_cols:

    discretized_dataset[col] = pd.qcut(discretized_dataset[col], q=3, labels=labels)

print("\nDiscretized data preview:")
print(discretized_dataset.head())
```

```
Discretized data preview:
  SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm      Species
0           Low         High           Low          Low  Iris-setosa
1           Low       Medium           Low          Low  Iris-setosa
2           Low       Medium           Low          Low  Iris-setosa
3           Low       Medium           Low          Low  Iris-setosa
4           Low         High           Low          Low  Iris-setosa
```

```python
transactions = []
for i in range(len(discretized_dataset)):

    row = discretized_dataset.iloc[i]
    transaction = [
        f"SepalLength_{row.SepalLengthCm}",
        f"SepalWidth_{row.SepalWidthCm}",
        f"PetalLength_{row.PetalLengthCm}",
        f"PetalWidth_{row.PetalWidthCm}",
        f"Species_{row.Species}"
    ]
    transactions.append(transaction)

print("\nExample of a transaction:")
print(transactions[0])
```

```
Example of a transaction:
['SepalLength_Low', 'SepalWidth_High', 'PetalLength_Low', 'PetalWidth_Low', 'Species_Iris-setosa']
```

```
!pip install apyori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5954 sha256=260b90d4b6f9d594815948f698483cbb07e82dcf62dd1b48185adf18ac85e8a2
  Stored in directory: /root/.cache/pip/wheels/7f/49/e3/42c73b19a264de37129fadaa0c52f26cf50e87de08fb9804af
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

```python
from apyori import apriori


rules = apriori(transactions=transactions,
                min_support=0.1,
                min_confidence=0.7,
                min_lift=1.2,
                min_length=2)
results = list(rules)
print(f"\nFound {len(results)} association rules.")
```

```
Found 81 association rules.
```

```python
def inspect_results(results):
    """Formats the Apriori results into a readable DataFrame."""
    rule_list = []
    for result in results:
        for ordered_stat in result.ordered_statistics:
            rule = {
                'Antecedent (If)': ', '.join(ordered_stat.items_base),
                'Consequent (Then)': ', '.join(ordered_stat.items_add),
                'Support': result.support,
                'Confidence': ordered_stat.confidence,
                'Lift': ordered_stat.lift
            }
            rule_list.append(rule)
    return pd.DataFrame(rule_list)

if results:
    results_df = inspect_results(results)
    print("\nDiscovered Association Rules:")

    print(results_df.sort_values(by='Lift', ascending=False))
else:
    print("\nNo rules found. Try lowering the min_support or min_confidence values.")
```

```
Discovered Association Rules:
                                     Antecedent (If)  \
166                PetalLength_High, SepalWidth_Medium
145                PetalLength_High, SepalWidth_Medium
276  PetalLength_High, SepalWidth_Medium, Species_I...
274  PetalLength_High, SepalWidth_Medium, PetalWidt...
271                PetalLength_High, SepalWidth_Medium
..                                                 ...
96               SepalLength_Medium, PetalLength_Medium
24                                    PetalWidth_Medium
135         PetalWidth_Medium, Species_Iris-versicolor
87               PetalWidth_Medium, PetalLength_Medium
238  PetalWidth_Medium, Species_Iris-versicolor, Pe...

                                    Consequent (Then)   Support  Confidence  \
166           SepalLength_High, Species_Iris-virginica  0.120000    0.900000
145                 SepalLength_High, PetalWidth_High  0.120000    0.900000
276                 SepalLength_High, PetalWidth_High  0.113333    0.894737
274           SepalLength_High, Species_Iris-virginica  0.113333    0.894737
271  SepalLength_High, Species_Iris-virginica, Peta...  0.113333    0.850000
..                                                 ...       ...         ...
96                                   SepalWidth_Low  0.180000    0.729730
24                                   SepalWidth_Low  0.246667    0.711538
135                                  SepalWidth_Low  0.226667    0.708333
87                                   SepalWidth_Low  0.220000    0.702128
238                                  SepalWidth_Low  0.220000    0.702128

         Lift
166  4.354839
145  4.354839
276  4.329372
274  4.329372
271  4.250000
..        ...
96   1.920341
24   1.872470
135  1.864035
87   1.847704
238  1.847704

[328 rows x 5 columns]
```

Conclusion:

The Apriori algorithm successfully identified frequent itemsets and generated meaningful association rules from the given dataset. By iteratively exploring itemsets and pruning infrequent candidates, the algorithm efficiently narrowed down patterns that occur frequently together. The rules generated provide valuable insights into relationships among items, which can be applied to areas such as market basket analysis to understand customer purchasing behavior. Overall, the implementation demonstrated how Apriori effectively uncovers hidden correlations in transactional data, enabling informed decision-making based on data-driven trends.