Week – 11

Write a Python program to generate frequent item sets / association rules using FP-growth Tree algorithm.

AIM: To write a Python program to generate frequent item sets / association rules using FP-growth Tree algorithm.

DECRIPTION:

Frequent Itemset: The itemset that occurs frequently is called frequent itemset.
The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations. For so much, it uses a divide-and-conquer strategy. The core of this method is the usage of a special data structure named frequent-pattern tree (FP-tree), which retains the item set association information.
FP-Tree: The frequent-pattern tree (FP-tree) is a compact data structure that stores quantitative information about frequent patterns in a database. Each transaction is read and then mapped onto a path in the FP-tree. This is done until all transactions have been read.

PROGRAM:

```
pip install pyfpgrowth
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyfpgrowth
  Downloading pyfpgrowth-1.0.tar.gz (1.6 MB)
     |████████████████████████████████| 1.6 MB 7.6 MB/s
Building wheels for collected packages: pyfpgrowth
  Building wheel for pyfpgrowth (setup.py) ... done
  Created wheel for pyfpgrowth: filename=pyfpgrowth-1.0-py2.py3-none-any.whl size=5503 sha256=28a93758a505c3977be0eeb4b3888429f1842534807a82a849bd2bb402ac7fa6
  Stored in directory: /root/.cache/pip/wheels/73/97/4b/f12ac994f6bbb99597396255435824c73ad3916be1e678be55
Successfully built pyfpgrowth
Installing collected packages: pyfpgrowth
Successfully installed pyfpgrowth-1.0
```

```
#sample code to do FP- growth in python
import pyfpgrowth
#creating Sample Transactions
transactions = [
    ['Milk', 'Bread','Saffron'],
    ['Peanuts','Milk'],
    ['Honey','Coconut','Water'],
    ['Orange','Jam']
]
```

```
#finding the frequent patterns with min support threshold=0.5
FrequentPatterns = pyfpgrowth.find_frequent_patterns(transactions = transactions,support_threshold = 0.5)
print(FrequentPatterns)
```

OUTPUT:

{('Bread',): 1, ('Bread', 'Milk'): 1, ('Saffron',): 1, ('Bread', 'Saffron'): 1, ('Milk', 'Saffron'): 1, ('Bread', 'Milk', 'Saffron'): 1, ('Peanuts',): 1, ('Milk', 'Peanuts'): 1, ('Honey',): 1, ('Coconut',): 1, ('Coconut', 'Honey'): 1, ('Water',): 1, ('Coconut', 'Water'): 1, ('Honey', 'Water'): 1, ('Coconut', 'Honey', 'Water'): 1, ('Orange',): 1, ('Jam',): 1, ('Jam', 'Orange'): 1, ('Milk',): 2}

```
#generating rules with min confidence threshold=0.5
Rules = pyfpgrowth.generate_association_rules(patterns = FrequentPatterns,confidence_threshold=0.5)
print(Rules)
```

OUTPUT:

{('Bread',): (('Milk', 'Saffron'), 1.0), ('Milk',): (('Peanuts',), 0.5), ('Saffron',): (('Bread', 'Milk'), 1.0), ('Bread', 'Milk'): (('Saffron',), 1.0), ('Bread', 'Saffron'): (('Milk',), 1.0), ('Milk', 'Saffron'): (('Bread',), 1.0), ('Peanuts',): (('Milk',), 1.0), ('Coconut',): (('Honey', 'Water'), 1.0), ('Honey',): (('Coconut', 'Water'), 1.0), ('Water',): (('Coconut', 'Honey'), 1.0), ('Coconut', 'Honey'): (('Water',), 1.0), ('Coconut', 'Water'): (('Honey',), 1.0), ('Honey', 'Water'): (('Coconut',), 1.0), ('Jam',): (('Orange',), 1.0), ('Orange',): (('Jam',), 1.0)}