Experiment No. 7
Implement frequent pattern mining algorithm(Apriori)
Date of Performance:
Date of Submission:

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

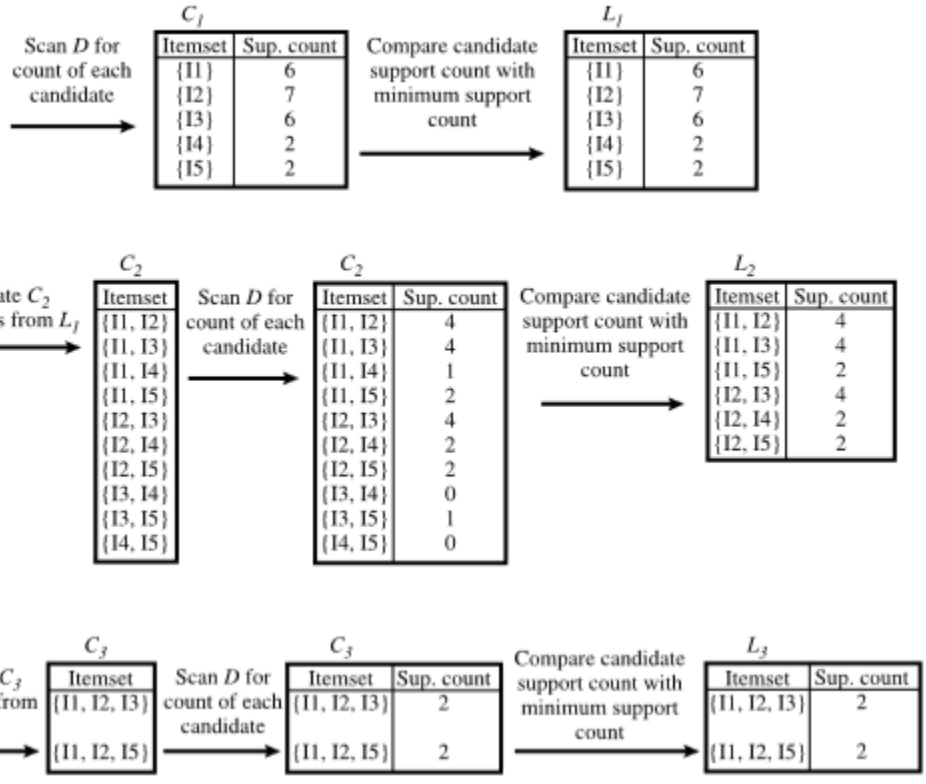
3. Count the support of each candidate by scanning the DB

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

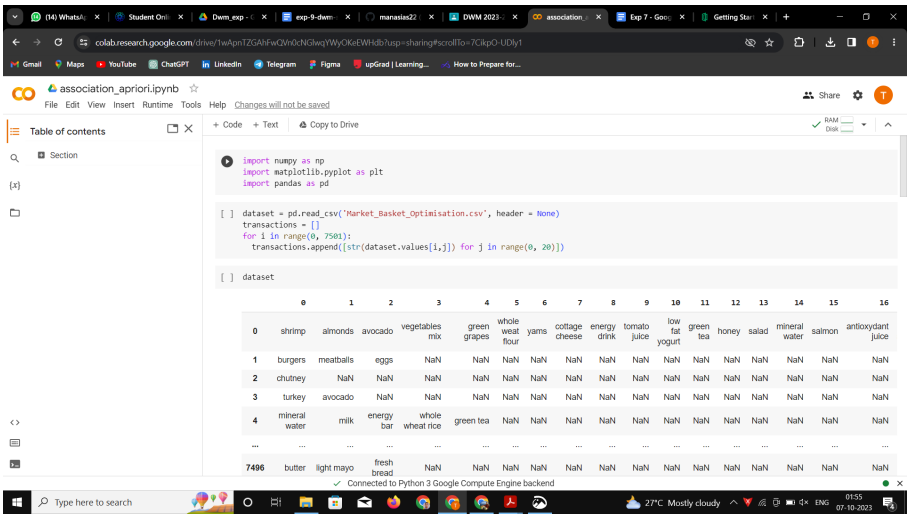
Apriori Algorithm:

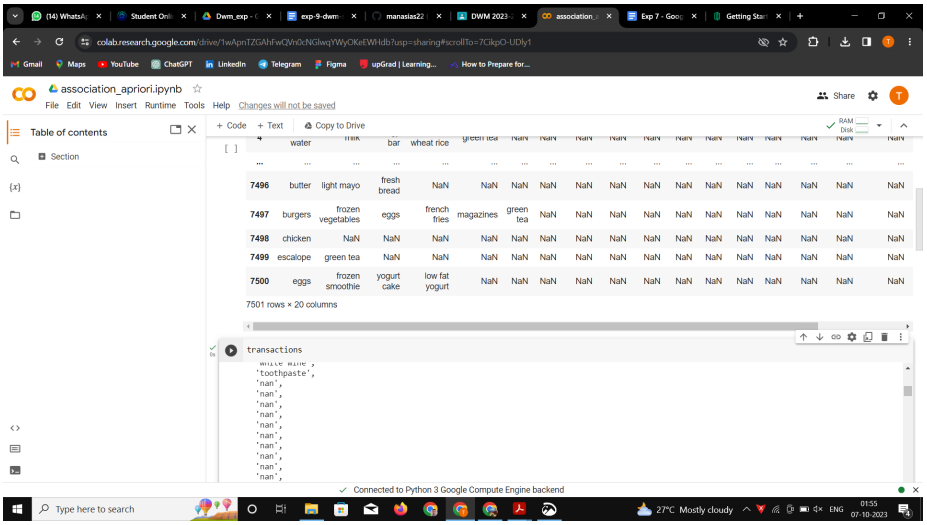
Transactional data for an *AllElectronics* branch.

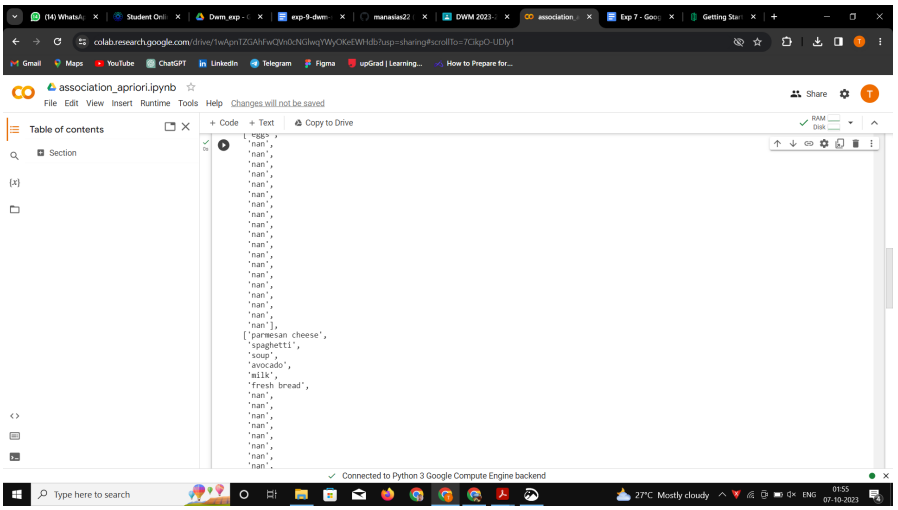
TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

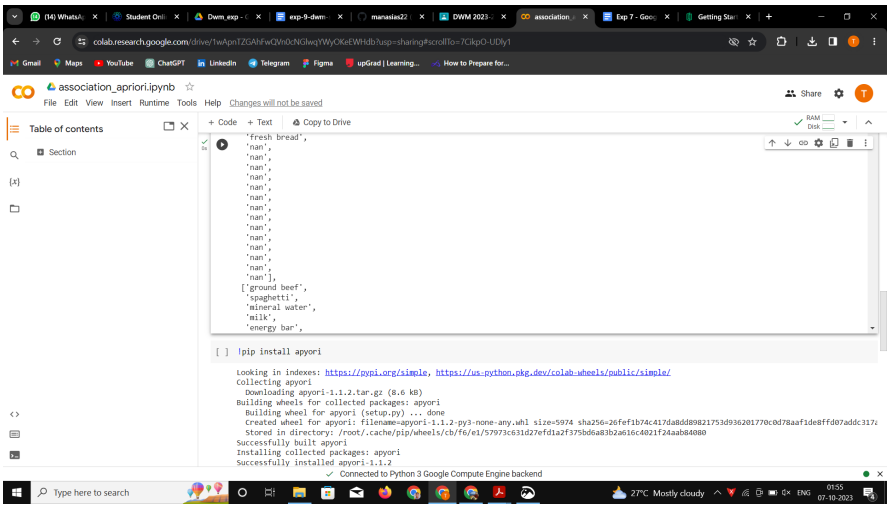


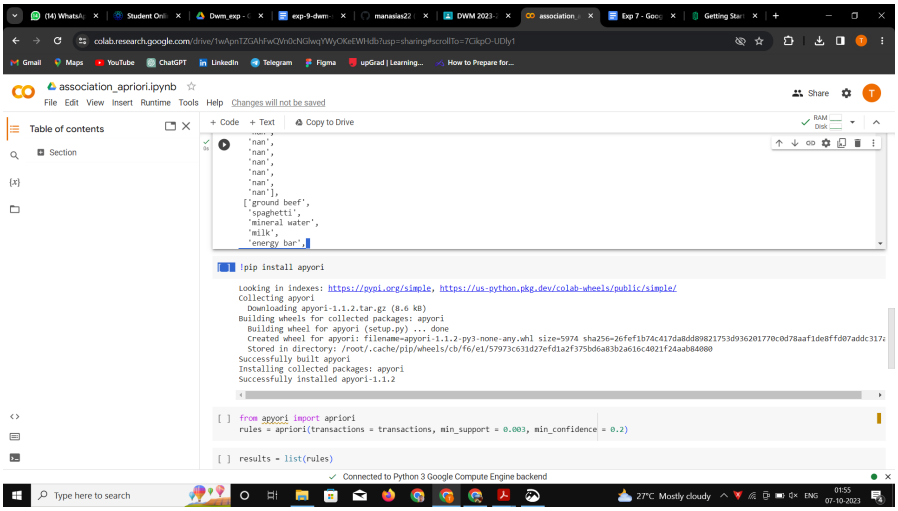
Code:


association_apriori.ipynb
Table of contents
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv('Market_Basket_Optimisation.csv', header = None)
transactions = []
for i in range(0, 7501):
transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])
dataset
Connected to Python 3 Google Compute Engine backend

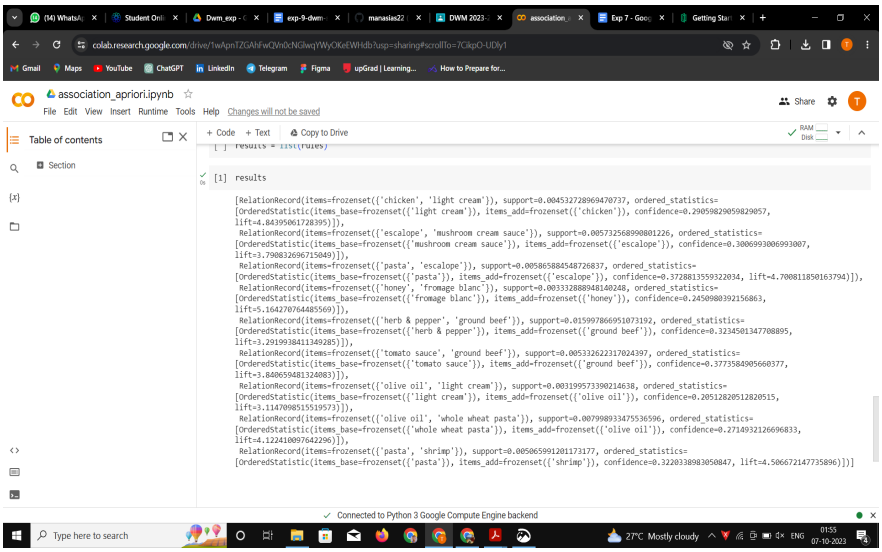

association_apriori.ipynb
Table of contents
7501 rows × 20 columns
transactions
Connected to Python 3 Google Compute Engine backend


association_apriori.ipynb
Table of contents
Section
['parmesan cheese',
'spaghetti',
'soup',
'avocado',
'milk',
'fresh bread',
Connected to Python 3 Google Compute Engine backend


association_apriori.ipynb
Table of contents
Section
['ground beef',
'spaghetti',
'mineral water',
'milk',
'energy bar',
[ ] !pip install apyori
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting apyori
Downloading apyori-1.1.2.tar.gz (8.6 kB)
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=5974 sha256=26fef1b74c417da8dd89821753d936201770c0d78aaf1de8ffd07addc317
Stored in directory: /root/.cache/pip/wheels/cb/f6/e1/57973c631d27efd1a2f375bd6a83b2a616c4021f24aab84080
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
Connected to Python 3 Google Compute Engine backend


association_apriori.ipynb
Table of contents
Section
['ground beef',
'spaghetti',
'mineral water',
'milk',
'energy bar',
!pip install apyori
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting apyori
Downloading apyori-1.1.2.tar.gz (8.6 kB)
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=5974 sha256=26fef1b74c417da8dd89821753d936201770c0d78aaf1de8ffd07addc317
Stored in directory: /root/.cache/pip/wheels/cb/f6/e1/57973c631d27efd1a2f375bd6a83b2a616c4021f24aab84080
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
[ ] from apyori import apriori
rules = apriori(transactions = transactions, min_support = 0.003, min_confidence = 0.2)
[ ] results = list(rules)
Connected to Python 3 Google Compute Engine backend

Output:



Conclusion: Thus, we successfully implemented Association Rule Mining algorithm (Apriori) .

Conclusion: Comment on the rules generated by the algorithm