TY B.Tech. (CSE) – II [2020-21]
4CS462: PE-2 - Data Mining Lab.
Assignment No. [08]

Date: 15/11/2021

GroupID:	DM21G07	
Group Members:	2018BTECS00063	Aryan Mali
	2018BTECS00094	Shreya Singh
	2018BTECS00099	Ganesh Kasar

Title: Apriori algorithm

Aim: Implement the Apriori algorithm for generating Association Rules

Introduction:

Apriori algorithm is given by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a dataset for boolean association rule. Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. 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.

Theory:

Apriori Property –

All non-empty subsets of a frequent itemset must be frequent. The key concept of the Apriori algorithm is its anti-monotonicity of support measure. Apriori assumes that

All subsets of a frequent itemset must be frequent(Apriori property).

If an item set is infrequent, all its supersets will be infrequent.

Before we start understanding the algorithm, go through some definitions which are explained in my previous post.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

minimum support count is 2

minimum confidence is 60%

Step-1: K=1

(I) Create a table containing the support count of each item present in the dataset – Called C1(candidate set)

(II) compare candidate set item's support count with minimum support count(here min_support=2 if support_count of candidate set items is less than min_support then remove those items). This gives us itemset L1.

Step-2: K=2

Generate candidate set C2 using L1 (this is called join step). The condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common.

Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)

Now find the support count of these itemsets by searching in the dataset.

(II) compare candidate (C2) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L2.

Step-3:

Generate candidate set C3 using L2 (join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common. So here, for L2, first element should match.

So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5}

Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)

find support count of these remaining itemset by searching in dataset.

(II) Compare candidate (C3) support count with minimum support count(here min_support=2 if support_count of candidate set item is less than min_support then remove those items) this gives us itemset L3.

Step-4:

Generate candidate set C4 using L3 (join step). Condition of joining Lk-1 and Lk-1 (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.

Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4

We stop here because no frequent itemsets are found further

Procedure:

Use / extend the data analysis tool (menu driven GUI) developed in Assignment No. 2

to perform the following task :

1. Implement the Apriori algorithm for generating Association Rules

2. Experiment with different values of support, confidence, and maximum rule length.

3. Tabulate the results containing frequent item sets, total number of rules generated

for different support and confidence.

4. Find the interesting rules from above obtained rules using following

metrics/measures

a. Lift

b. Chi-Square Test χ

2

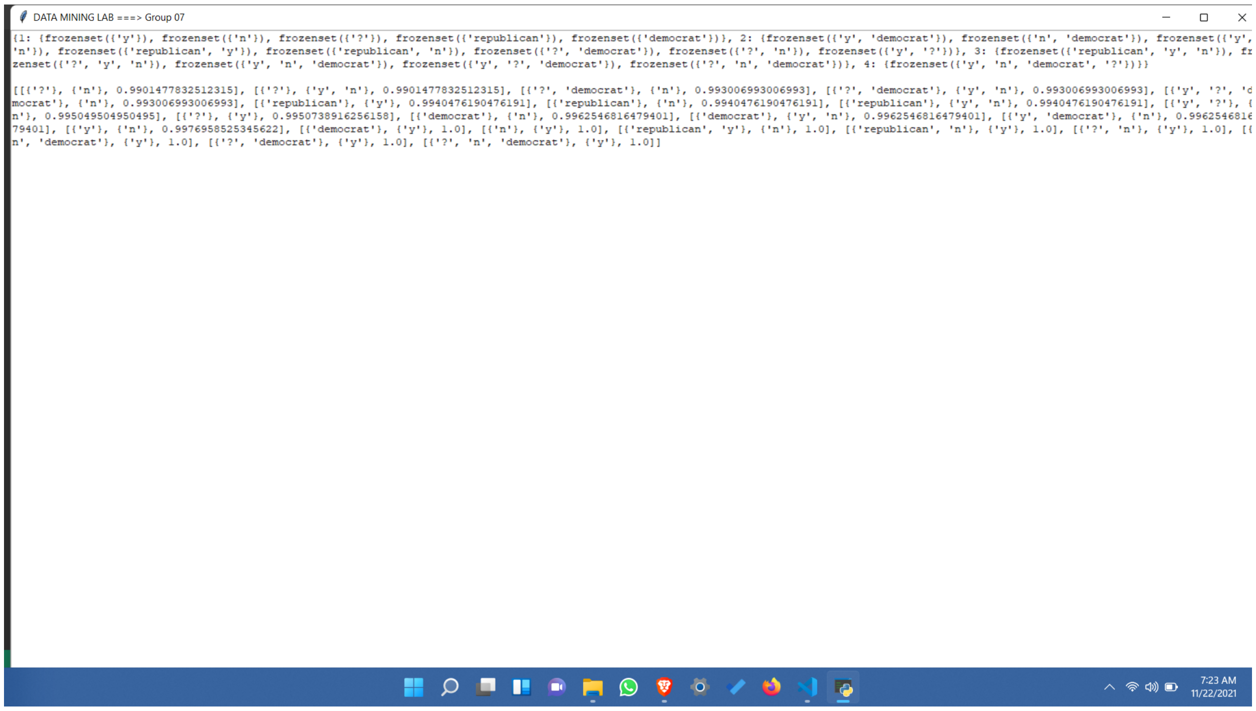
c. All_confidence measure

d. Max_confidence measure

e. Kulczynski measure

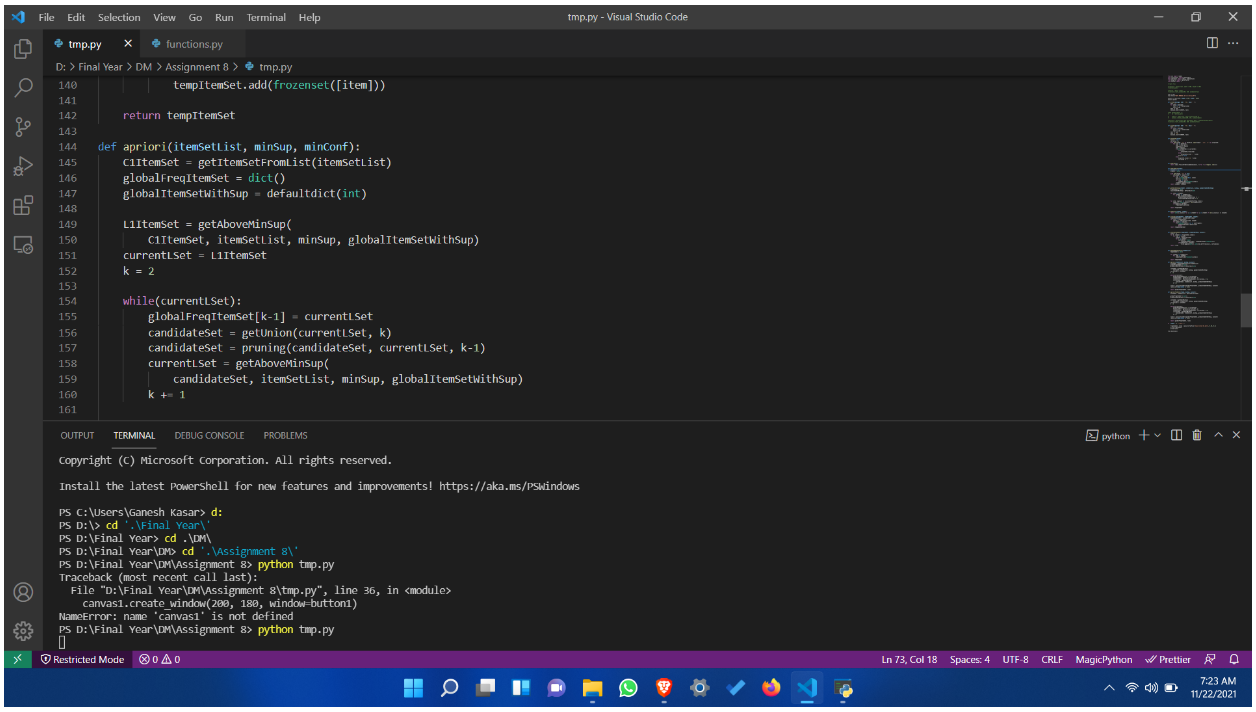
f. Cosine measure

Results:


DATA MINING LAB ===> Group 07

```
{1: {frozenset({'y'}), frozenset({'n'}), frozenset({'?'}), frozenset({'republican'}), frozenset({'democrat'})}, 2: {frozenset({'y', 'democrat'}), frozenset({'n', 'democrat'}), frozenset({'y',
'n'}), frozenset({'republican', 'y'}), frozenset({'republican', 'n'}), frozenset({'?', 'democrat'}), frozenset({'?', 'n'}), frozenset({'y', '?'})}, 3: {frozenset({'republican', 'y', 'n'}), fr
zenset({'?', 'y', 'n'}), frozenset({'y', 'n', 'democrat'}), frozenset({'y', '?', 'democrat'}), frozenset({'?', 'n', 'democrat'})}, 4: {frozenset({'y', 'n', 'democrat', '?'})}}

[[{'?'}, {'n'}, 0.9901477832512315], [{'?'}, {'y', 'n'}, 0.9901477832512315], [{'?', 'democrat'}, {'n'}, 0.993006993006993], [{'?', 'democrat'}, {'y', 'n'}, 0.993006993006993], [{'y', '?', 'd
mocrat'}, {'n'}, 0.993006993006993], [{'republican'}, {'y'}, 0.9940476190476191], [{'republican'}, {'n'}, 0.9940476190476191], [{'republican'}, {'y', 'n'}, 0.9940476190476191], [{'y', '?'}, {
n'}, 0.995049504950495], [{'?'}, {'y'}, 0.9950738916256158], [{'democrat'}, {'n'}, 0.9962546816479401], [{'democrat'}, {'y', 'n'}, 0.9962546816479401], [{'y', 'democrat'}, {'n'}, 0.996254681
79401], [{'y'}, {'n'}, 0.9976958525345622], [{'democrat'}, {'y'}, 1.0], [{'n'}, {'y'}, 1.0], [{'republican', 'y'}, {'n'}, 1.0], [{'republican', 'n'}, {'y'}, 1.0], [{'?', 'n'}, {'y'}, 1.0], [{
n', 'democrat'}, {'y'}, 1.0], [{'?', 'democrat'}, {'y'}, 1.0], [{'?', 'n', 'democrat'}, {'y'}, 1.0]]
```


tmp.py - Visual Studio Code

D: > Final Year > DM > Assignment 8 > tmp.py

```python
            tempItemSet.add(frozenset([item]))

    return tempItemSet

def apriori(itemSetList, minSup, minConf):
    C1ItemSet = getItemSetFromList(itemSetList)
    globalFreqItemSet = dict()
    globalItemSetWithSup = defaultdict(int)

    L1ItemSet = getAboveMinSup(
        C1ItemSet, itemSetList, minSup, globalItemSetWithSup)
    currentLSet = L1ItemSet
    k = 2

    while(currentLSet):
        globalFreqItemSet[k-1] = currentLSet
        candidateSet = getUnion(currentLSet, k)
        candidateSet = pruning(candidateSet, currentLSet, k-1)
        currentLSet = getAboveMinSup(
            candidateSet, itemSetList, minSup, globalItemSetWithSup)
        k += 1
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ganesh Kasar> d:
PS D:\> cd '.\Final Year\'
PS D:\Final Year> cd .\DM\
PS D:\Final Year\DM> cd '.\Assignment 8\'
PS D:\Final Year\DM\Assignment 8> python tmp.py
Traceback (most recent call last):
  File "D:\Final Year\DM\Assignment 8\tmp.py", line 36, in <module>
    canvas1.create_window(200, 180, window=button1)
NameError: name 'canvas1' is not defined
PS D:\Final Year\DM\Assignment 8> python tmp.py
```

File Edit Selection View Go Run Terminal Help
tmp.py
functions.py
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS
python
Restricted Mode
Ln 73, Col 18 Spaces: 4 UTF-8 CRLF MagicPython Prettier
7:23 AM
11/22/2021