CS 634

Data Mining Midterm Project

Report

Apriori Algorithm implementation in Python

https://github.com/NeeharikaPisipati/CS634

**Introduction:**

The Apriori algorithm is used to find out frequently occurring item sets in a given transactional database and since it relies on prior knowledge about common itemset qualities, the algorithm is called Apriori. The Apriori property states that:

1. A frequent item set’s subsets must all be frequent (Apriori property).
2. If an itemset is rare, all of its supersets are rare as well.

**Stages of Apriori algorithm implementation:**

1. Organizing the data: the data must be structured in such a way that each line contains a set of products whose collections contains items purchased in a single transaction.
2. Find out all the possible subsets for the item sets in the dataset.
3. Compute the support for each individual item in the itemset.
4. Select a minimum support value as the threshold.
5. Choose the frequently occurring datasets.
6. Repeat finding the support for these frequent sets.
7. Increase the selection of items in the frequent set and find the support.
8. For all these item sets, generate the association rules by giving the minimum support value and then, find the confidence.
9. Once the rules are obtained, we need to find the lift which is calculated as the improvement of a generated association rule over the entire dataset.

**Accuracy metrics of the Apriori Algorithm:**

The support confidence and lift determines the accuracy of the algorithm implementation. So, for every dataset, the metrics need to be determined for us to improve the accuracy.

**Limitations of Apriori Algorithm:**

The Apriori Algorithm is slow. The main constraint is the amount of time it takes to keep track of a large number of candidate sets with frequent itemsets, low minimum support, or huge itemsets.

**Implementation in Python:**

**HARDWARE CONFIGURATION:**

Operating System: Windows 10

Processor: Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz

RAM: 12.0 GB (11.8 GB usable)

**DATASET:**

Transaction1.csv

Transaction2.csv

Transaction3.csv

Transaction4.csv

Transaction5.csv

**SOFTWARES USED:**

Python 3.8 version installed in the system

Google Colaboratory to execute the Python Code

**LIBRARIES USED:**

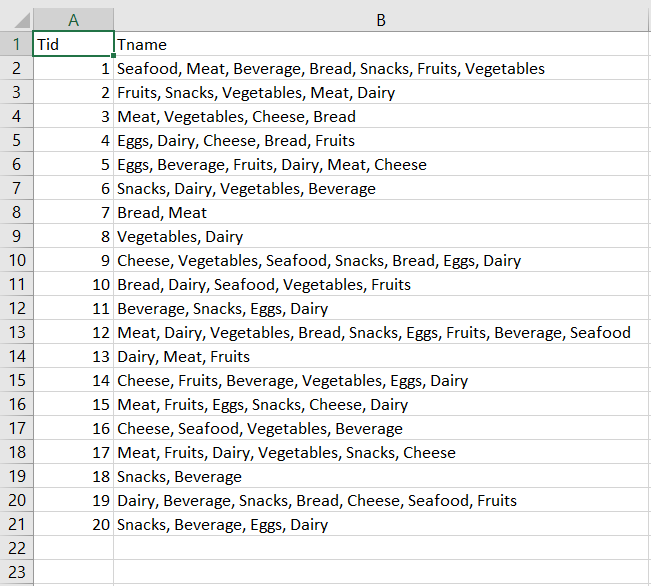
Pandas

Numpy

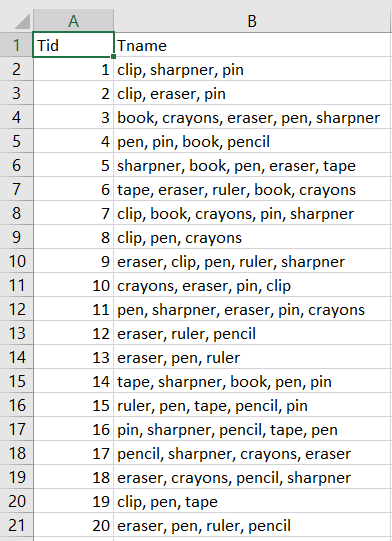
Csv

**Datasets:**

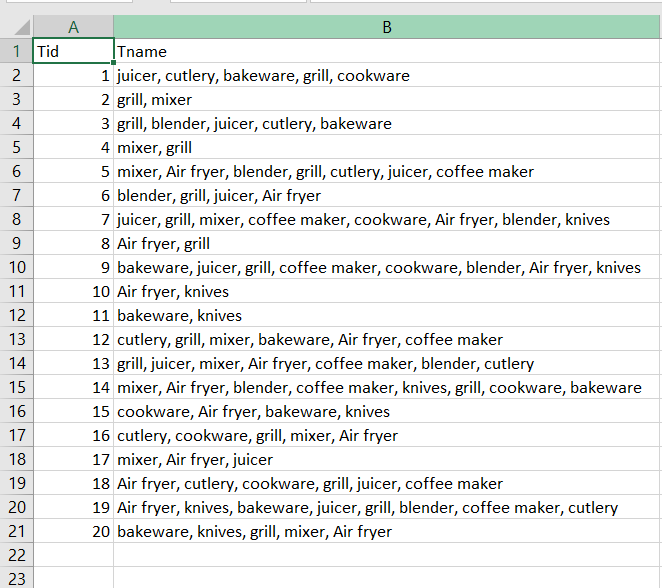
Transaction1.csv



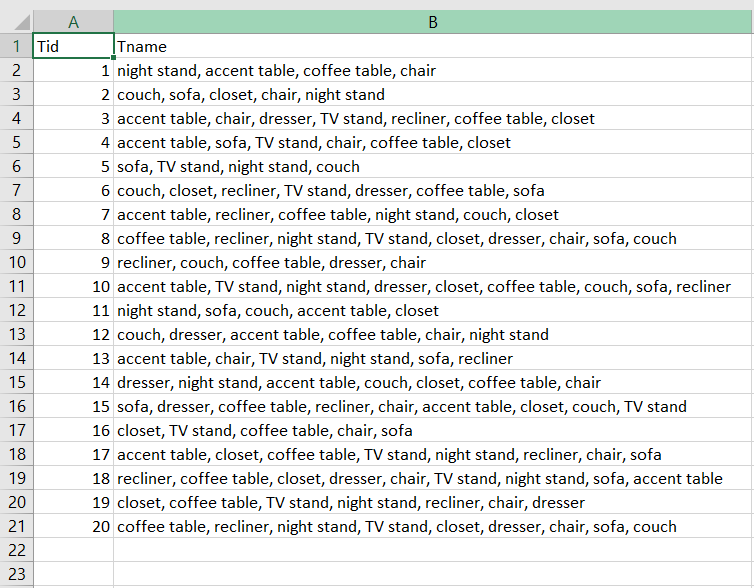
Transaction2.csv:



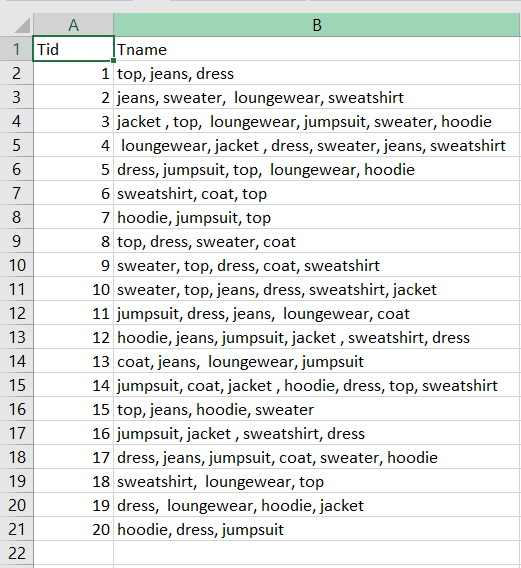
Transaction3.csv:



Transaction4.csv:



Transaction5.csv:

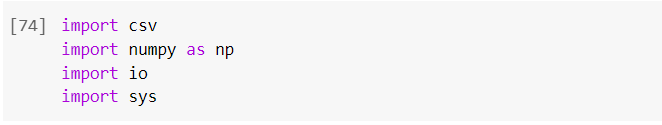


I have created 5 databases of 20 transactions each containing some of these items saved in a CSV file. Each of these datasets contains transactions of items that we see in everyday usage.

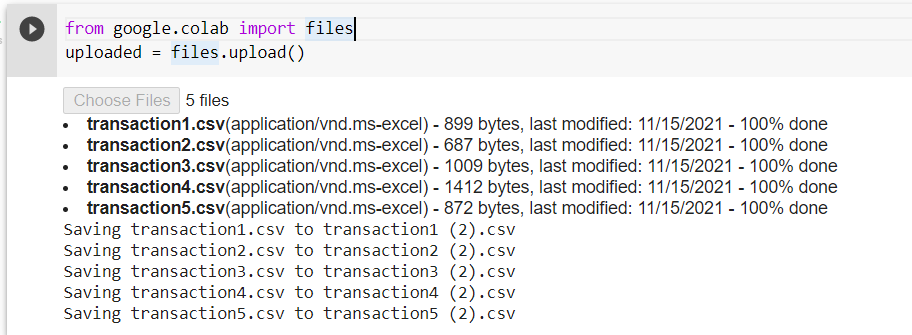
Code Implementation:

I have used Google Colaboratory to write and execute my python code.

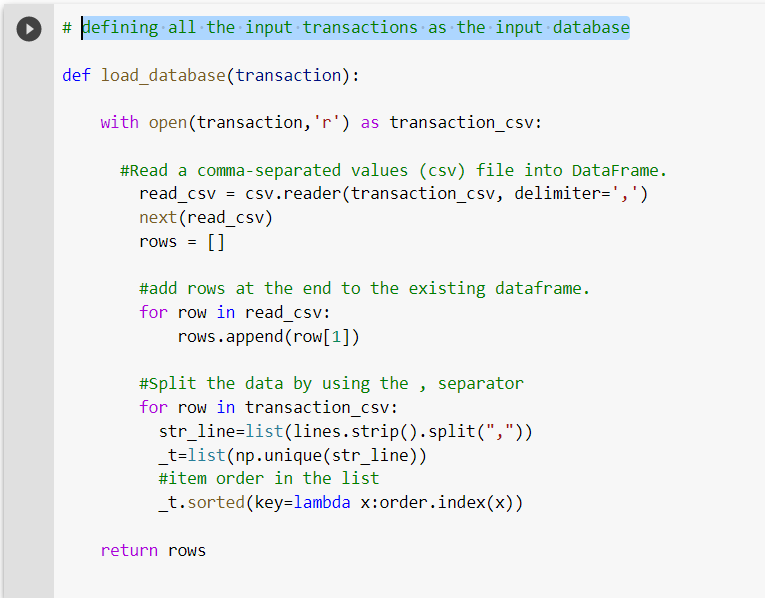
1. Import the necessary libraries.



1. Import the 5 transaction csv files from the local machine.



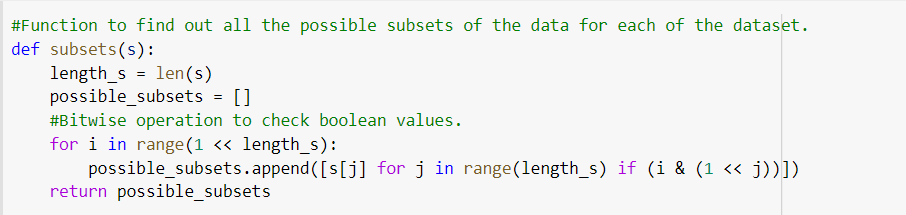
1. Define all the input transactions as the input database.



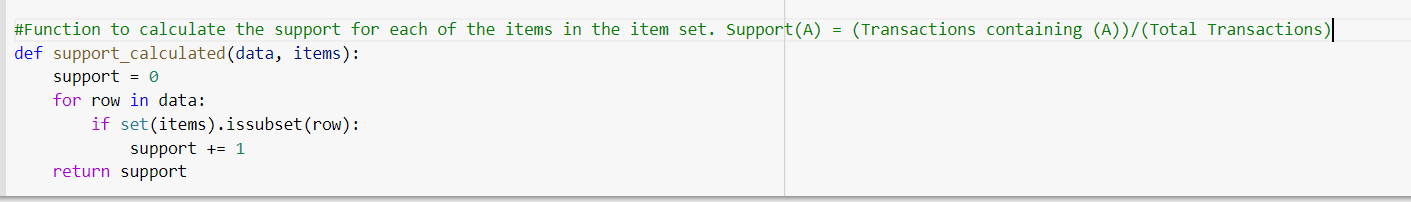
1. Preparing the data:

Split the data using the ‘,’ as the separator and sort the items in the transaction using a lambda function.

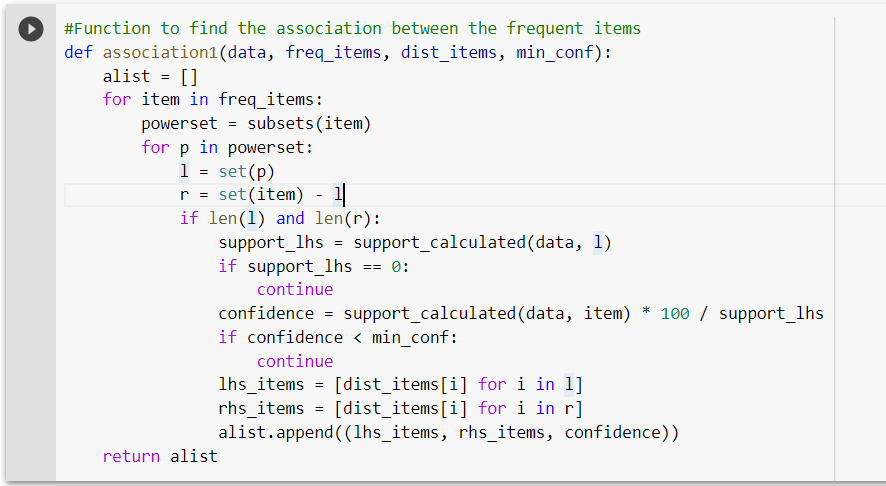
1. Find all the possible subsets of the item sets from the transactional database. Create individual functions so that they can be reused for each of the 5 datasets.



1. Create a function to find the support of all the items in the transactional database.



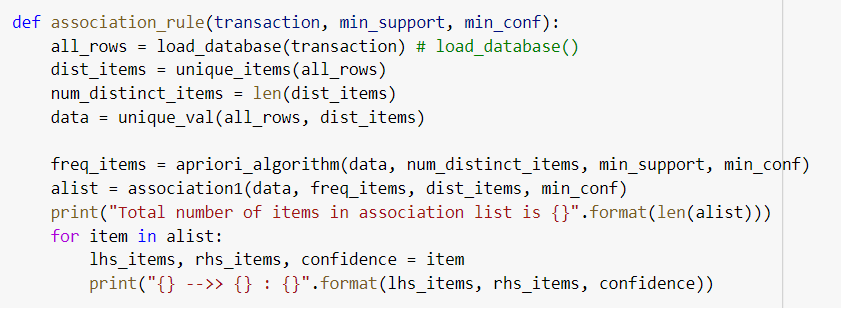
1. Find out the association from items

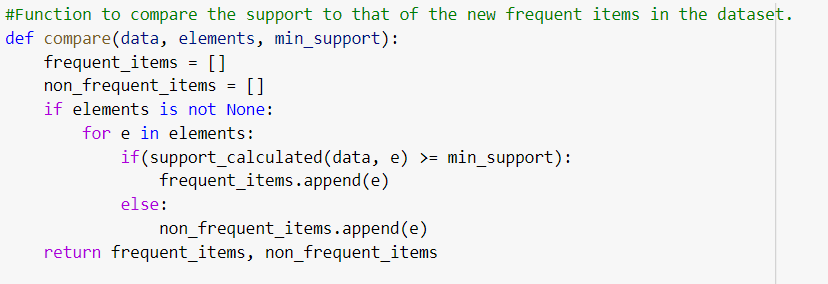


1. Find the new frequent items in the dataset.

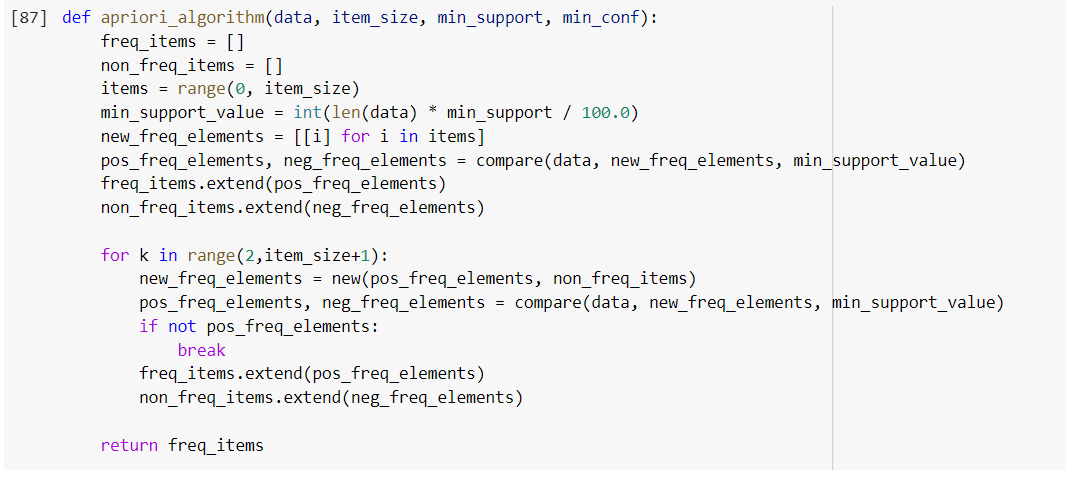


1. Calculate their support.
2. Compare it with the existing support
3. Association rules:





1. Check the non-frequent items
2. Increase the number of items and trace the frequent patterns again.
3. Finally, run the apriori algorithm with all the created functions against each individual dataset.



Outputs:

