CS 634(Section 104) Data Mining - Spring2023
Date:3/9/2024
Professor Yasser Abduallah

Midterm Project Report
Implementation of Apriori Algorithm and Brute force algorithm

Aakash Siricilla (as4592)

NJIT ID : 233370

Contents

Introduction:

What Is An Itemset?

It refers to a collection of items grouped together. When an itemset comprises k items, it's termed a k-itemset. Typically, an itemset consists of at least two items. A frequently occurring itemset earns the designation of a "frequent itemset." Consequently, frequent itemset mining serves as a data mining method aimed at recognizing items that commonly appear together. Examples include combinations like bread and butter, or laptop and antivirus software.

What Is A Frequent Itemset?

A set of items qualifies as "frequent" when it meets a predefined threshold for support and confidence. Support indicates the occurrence of items purchased together within a single transaction, while confidence signifies transactions where items are bought sequentially. In the process of frequent itemset mining, only transactions meeting the minimum support and confidence criteria are considered. Extracting insights through these mining techniques yields numerous advantages such as cost reduction and enhanced competitive edge.

Frequent Pattern Mining (FPM)

The frequent pattern mining algorithm is one of the most important techniques of data mining to discover relationships between different items in a dataset. These relationships are represented in the form of association rules. It helps to find the irregularities in data.

FPM has many applications in the field of data analysis, software bugs, cross-marketing, sale campaign analysis, market basket analysis, etc.

Frequent itemsets discovered through Apriori have many applications in data mining tasks. Tasks such as finding interesting patterns in the database, finding sequence and Mining of association rules are the most important of them.

Association rules apply to supermarket transaction data, that is, to examine the customer behavior in terms of the purchased products. Association rules describe how often the items are purchased together.

Association Rules

Association Rule Mining is defined as:
"Let I= { …} be a set of 'n' binary attributes called items. Let D= { ….} be a set of transactions called databases. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of form X->Y where X, Y? I and X?Y=?. The set of items X and Y are called antecedent and consequent of the rule respectively."
Learning of Association rules is used to find relationships between attributes in large databases. An association rule, A=> B, will be of the form" for a set of transactions, some value of itemset A determines the values of itemset B under the condition in which minimum support and confidence are met".

Support and Confidence can be represented by the following example:
Bread=> butter [support=2%, confidence-60%]

The above statement is an example of an association rule. This means that there is a 2% transaction that bought bread and butter together and there are 60% of customers who bought bread as well as butter.

Support and Confidence for Itemset A and B are represented by formulas:

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

$$\text{Confidence } (A \longrightarrow B) = \frac{\text{Support(AUB)}}{\text{Support(A)}}$$

Association rule mining consists of 2 steps:

1. Find all the frequent itemsets.
2. Generate association rules from the above frequent itemsets.

Why Frequent Itemset Mining?
Frequent itemset or pattern mining is broadly used because of its wide applications in mining association rules, correlations and graph patterns constraint that is based on frequent patterns, sequential patterns, and many other data mining tasks.

Apriori Algorithm – Frequent Pattern Algorithms
Apriori algorithm was the first algorithm that was proposed for frequent itemset mining. It was later improved by R Agarwal and R Srikant and came to be known as Apriori. This algorithm

uses two steps “join” and “prune” to reduce the search space. It is an iterative approach to discover the most frequent itemsets.

Apriori says:
The probability that item I is not frequent is if:

- P(I) < minimum support threshold, then I is not frequent.
- P (I+A) < minimum support threshold, then I+A is not frequent, where A also belongs to the itemset.
- If an itemset set has value less than minimum support then all of its supersets will also fall below min support, and thus can be ignored. This property is called the Anti Monotone property.

The steps followed in the Apriori Algorithm of data mining are:

1. **Join Step**: This step generates (K+1) itemset from K-itemsets by joining each item with itself.
2. **Prune Step**: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.
3.

Steps In Apriori

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

#1) In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.

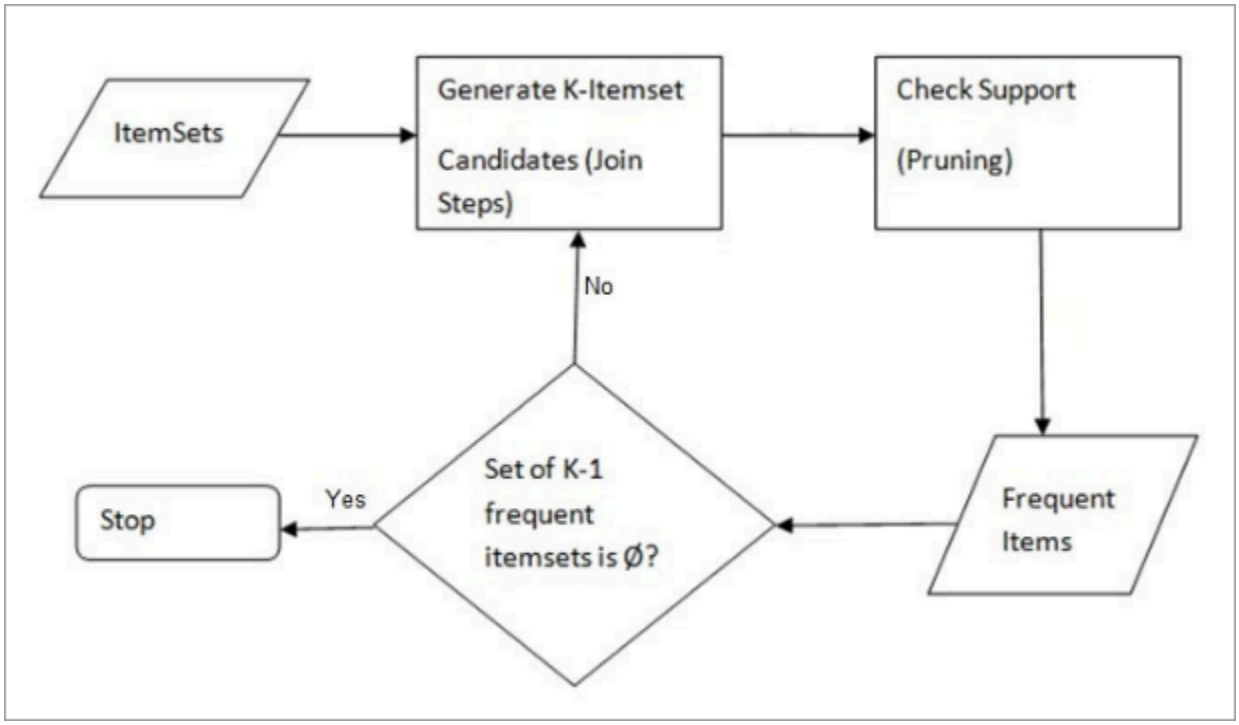
#2) Let there be some minimum support, min_sup (eg 2). The set of 1 – itemsets whose occurrence is satisfying the min sup are determined. Only those candidates which count more than or equal to min_sup, are taken ahead for the next iteration and the others are pruned.

#3) Next, 2-itemset frequent items with min_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.

#4) The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2 –itemsets with min-sup only.

#5) The next iteration will form 3 –itemsets using join and prune step. This iteration will follow anti monotone property where the subsets of 3-itemsets, that is the 2 –itemset subsets of each group fall in min_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.

#6) Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.



<u>Example of Apriori:</u> Support threshold=50%, Confidence= 60%

TABLE-1

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

Solution:
Support threshold=50% => 0.5*6= 3 => min_sup=3

Count Of Each Item

TABLE-2

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

Prune Step: TABLE -2 shows that I5 item does not meet min_sup=3, thus it is deleted, only I1, I2, I3, I4 meet min_sup count.

TABLE-3

Item	Count
I1	4
I2	5
I3	4
I4	4

Join Step: Form 2-itemset. From **TABLE-1** find out the occurrences of 2-itemset.

TABLE-4

Item	Count
I1,I2	4
I1,I3	3
I1,I4	2
I2,I3	4
I2,I4	3
I3,I4	2

Prune Step: TABLE -4 shows that item set {I1, I4} and {I3, I4} does not meet min_sup, thus it is deleted.

TABLE-5

Item	Count
I1,I2	4
I1,I3	3
I2,I3	4

Item	Count
I2,I4	3

Join and Prune Step: Form 3-itemset. From the **TABLE- 1** find out occurrences of 3-itemset. From **TABLE-5**, find out the 2-itemset subsets which support min_sup.
We can see for itemset {I1, I2, I3} subsets, {I1, I2}, {I1, I3}, {I2, I3} are occurring in **TABLE-5** thus {I1, I2, I3} is frequent.
We can see for itemset {I1, I2, I4} subsets, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} is not frequent, as it is not occurring in **TABLE-5** thus {I1, I2, I4} is not frequent, hence it is deleted.

TABLE-6

Item
I1,I2,I3
I1,I2,I4
I1,I3,I4
I2,I3,I4

Only {I1, I2, I3} is frequent.

Generate Association Rules: From the frequent itemset discovered above the association could be:
{I1, I2} => {I3}

Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4)* 100 = 75%

{I1, I3} => {I2}

Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3)* 100 = 100%

{I2, I3} => {I1}

Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4)* 100 = 75%

{I1} => {I2, I3}

Confidence = support {I1, I2, I3} / support {I1} = (3/ 4)* 100 = 75%

{I2} => {I1, I3}

Confidence = support {I1, I2, I3} / support {I2 = (3/ 5)* 100 = 60%

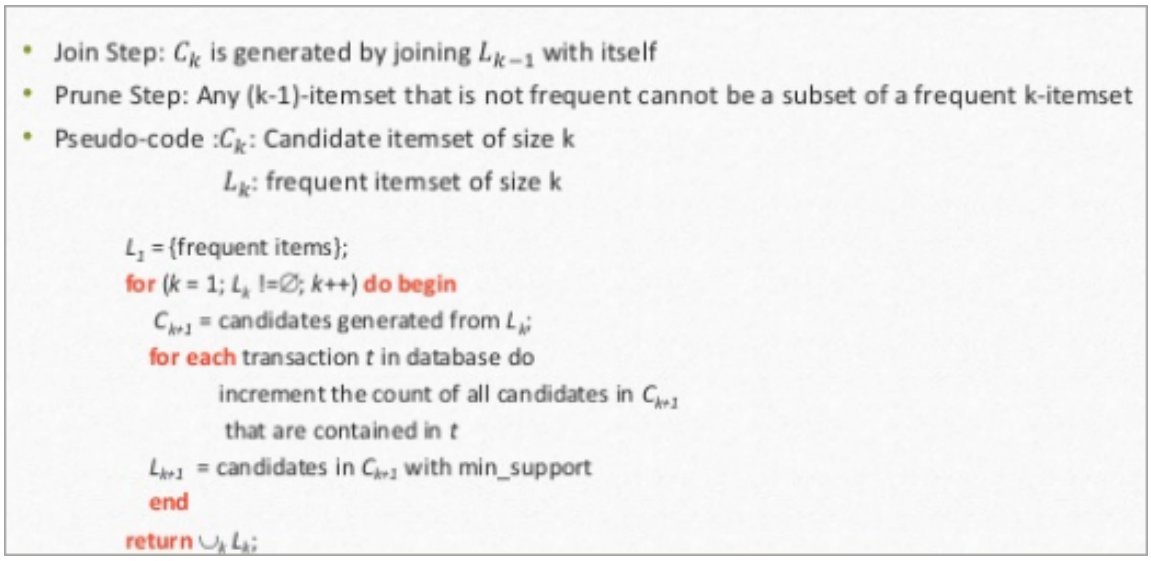
{I3} => {I1, I2}

Confidence = support {I1, I2, I3} / support {I3} = (3/ 4)* 100 = 75%

This shows that all the above association rules are strong if minimum confidence threshold is 60%.

The Apriori Algorithm: Pseudo Code
C: Candidate item set of size k

L: Frequent itemset of size k



Advantages

1. Easy to understand algorithm
2. Join and Prune steps are easy to implement on large itemsets in large databases

Disadvantages

1. It requires high computation if the itemsets are very large and the minimum support is kept very low.
2. The entire database needs to be scanned.

Methods To Improve Apriori Efficiency

Many methods are available for improving the efficiency of the algorithm.

1. **Hash-Based Technique:** This method uses a hash-based structure called a hash table for generating the k-itemsets and its corresponding count. It uses a hash function for generating the table.
2. **Transaction Reduction:** This method reduces the number of transactions scanning in iterations. The transactions which do not contain frequent items are marked or removed.
3. **Partitioning:** This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.
4. **Sampling:** This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.
5. **Dynamic Itemset Counting:** This technique can add new candidate itemsets at any marked start point of the database during the scanning of the database.

Applications Of Apriori Algorithm

1. **In Education Field:** Extracting association rules in data mining of admitted students through characteristics and specialties.

2. **In the Medical field:** For example Analysis of the patient's database.

3. **In Forestry:** Analysis of probability and intensity of forest fire with the forest fire data.

4. Apriori is used by many companies like Amazon in the **Recommender System** and by Google for the auto-complete feature.

Conclusion

Apriori algorithm is an efficient algorithm that scans the database only once.It reduces the size of the itemsets in the database considerably providing a good performance. Thus, data mining helps consumers and industries better in the decision-making process.

Dataset Creation

```
import pandas as pd
import numpy as np
dataset1={'Items':['Digital Camera','Lab Top','Desk Top','Printer','Flash Drive',
            'Microsoft Office','Speakers','Lab Top Case','Anti-Virus','External Hard-Drive']}
BestBuy_dataItems=pd.DataFrame(dataset1, index=[1,2,3,4,5,6,7,8,9,10])
#print(Bestbuy_dataItems)

dataset2={'Items':['Quilts','Bedspreads','Decorative Pillows','Bed Skirts','Sheets','Shams','Bedding Collections',
                   'Kids Bedding','Embroidered Bedspread','Towels']}
kmart_dataItems=pd.DataFrame(dataset2,index=[1,2,3,4,5,6,7,8,9,10])
#print(kmart_dataItems)

dataset3={'Items':['Running Shoe','Soccer Shoe','Socks','Swimming Shirt','Dry Fit V-Nick','Rash Guard','Sweatshirts',
                   'Hoodies','Tech Pants','Modern Pants']}
Nike_dataItems=pd.DataFrame(dataset3,index=[1,2,3,4,5,6,7,8,9,10])
#print(Nike_dataItems)

dataset4={'Items':['A Beginner Guide','Java:The Complete Reference','Java For Dummies',
                   'Android Programming:The Big Nerd Ranch','Head First Java 2nd Edition','Beginning Programming with Java',
                   'Java 8 Pocket Guide','C++ Programming in Easy Steps','Effective Java (2nd Edition)',
                   'HTML and CSS: Design and Build Websites']}
Amazon_dataItems=pd.DataFrame(dataset4,index=[1,2,3,4,5,6,7,8,9,10])
#print(Amazon_dataItems)

dataset5={'Items':['A','B','C','D']}
Generic_dataItems=pd.DataFrame(dataset5,index=[1,2,3,4])
#print(Generic_dataItems)

dataset6={'Items':['ink','pen','cheese','bag','juice','milk']}
Custom_dataItems=pd.DataFrame(dataset6,index=[1,2,3,4,5,6])
#print(Custom_dataItems)
```

```
data1 = {'Transaction': [['Desk Top', 'Printer', 'Flash Drive', 'Microsoft Office', 'Speakers', 'Anti-Virus'],
                         ['Lab Top', 'Flash Drive', 'Microsoft Office', 'Lab Top Case', 'Anti-Virus'],
                         ['Lab Top', 'Printer', 'Flash Drive', 'Microsoft Office', 'Anti-Virus', 'Lab Top Case', 'External Hard-Drive'],
                         ['Lab Top', 'Printer', 'Flash Drive', 'Anti-Virus', 'External Hard-Drive', 'Lab Top Case'],
                         ['Lab Top', 'Flash Drive', 'Lab Top Case', 'Anti-Virus'],
                         ['Lab Top', 'Printer', 'Flash Drive', 'Microsoft Office'],
                         ['Desk Top', 'Printer', 'Flash Drive', 'Microsoft Office'],
                         ['Lab Top', 'External Hard-Drive', 'Anti-Virus'],
                         ['Desk Top', 'Printer', 'Flash Drive', 'Microsoft Office', 'Lab Top Case', 'Anti-Virus', 'Speakers', 'External Hard-Drive'],
                         ['Digital Camera' , 'Lab Top', 'Desk Top', 'Printer', 'Flash Drive', 'Microsoft Office', 'Lab Top Case', 'Anti-Virus', 'External
                         ['Lab Top', 'Desk Top', 'Lab Top Case', 'External Hard-Drive', 'Speakers', 'Anti-Virus'],
                         ['Digital Camera' , 'Lab Top', 'Lab Top Case', 'External Hard-Drive', 'Anti-Virus', 'Speakers'],
                         ['Digital Camera' , 'Speakers'],
                         ['Digital Camera' , 'Desk Top', 'Printer', 'Flash Drive', 'Microsoft Office'],
                         ['Printer','Flash Drive', 'Microsoft Office', 'Anti-Virus', 'Lab Top Case', 'Speakers', 'External Hard-Drive'],
                         ['Digital Camera', 'Flash Drive', 'Microsoft Office', 'Anti-Virus', 'Lab Top Case', 'External Hard-Drive', 'Speakers'],
                         ['Digital Camera' , 'Lab Top', 'Lab Top Case'],
                         ['Digital Camera' , 'Lab Top Case', 'Speakers'],
                         ['Digital Camera' , 'Lab Top', 'Printer', 'Flash Drive', 'Microsoft Office', 'Speakers', 'Lab Top Case', 'Anti-Virus'],
                         ['Digital Camera' , 'Lab Top', 'Speakers', 'Anti-Virus', 'Lab Top Case']
    ]
       }


DB1 = pd.DataFrame(data1,index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB1.to_excel('BestBuy.xlsx')

data2 = {'Transaction': [['Decorative Pillows', 'Quilts', 'Embroidered Bedspread'],
                         ['Embroidered Bedspread', 'Shams', 'Kids Bedding', 'Bedding Collections', 'Bed Skirts', 'Bedspreads', 'Sheets'],
                         ['Decorative Pillows', 'Quilts', 'Embroidered Bedspread', 'Shams', 'Kids Bedding', 'Bedding Collections'],
                         ['Kids Bedding', 'Bedding Collections', 'Sheets', 'Bedspreads', 'Bed Skirts'],
                         ['Decorative Pillows', 'Kids Bedding', 'Bedding Collections', 'Sheets', 'Bed Skirts', 'Bedspreads'],
                         ['Bedding Collections', 'Bedspreads', 'Bed Skirts', 'Sheets', 'Shams', 'Kids Bedding'],
                         ['Decorative Pillows', 'Quilts'],
                         ['Decorative Pillows', 'Quilts','Embroidered Bedspread'],
                         ['Bedspreads', 'Bed Skirts', 'Shams', 'Kids Bedding', 'Sheets'],
                         ['Quilts', 'Embroidered Bedspread', 'Bedding Collections'],
                         ['Bedding Collections', 'Bedspreads', 'Bed Skirts', 'Kids Bedding', 'Shams', 'Sheets'],
                         ['Decorative Pillows', 'Quilts'],
                         ['Embroidered Bedspread', 'Shams'],
                         ['Sheets', 'Shams', 'Bed Skirts', 'Kids Bedding'],
                         ['Decorative Pillows', 'Quilts'],
                         ['Decorative Pillows', 'Kids Bedding', 'Bed Skirts', 'Shams'],
                         ['Decorative Pillows', 'Shams', 'Bed Skirts'],
                         ['Quilts', 'Sheets', 'Kids Bedding'],
                         ['Shams', 'Bed Skirts', 'Kids Bedding', 'Sheets'],
                         ['Decorative Pillows', 'Bedspreads', 'Shams', 'Sheets', 'Bed Skirts', 'Kids Bedding']]
       }
```

```
DB2 = pd.DataFrame(data2,index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB2.to_excel('kmart.xlsx')

data3 = {'Transaction': [['Running Shoe', 'Socks', 'Sweatshirts', 'Modern Pants'],
                    ['Running Shoe', 'Socks', 'Sweatshirts'],
                    ['Running Shoe', 'Socks', 'Sweatshirts', 'Modern Pants'],
                    ['Running Shoe', 'Sweatshirts', 'Modern Pants'],
                    ['Running Shoe', 'Socks', 'Sweatshirts', 'Modern Pants', 'Soccer Shoe'],
                    ['Running Shoe', 'Socks', 'Sweatshirts'],
                    ['Running Shoe', 'Socks', 'Sweatshirts', 'Modern Pants', 'Tech Pants', 'Rash Guard', 'Hoodies'],
                    ['Swimming Shirt', 'Socks', 'Sweatshirts'],
                    ['Swimming Shirt', 'Rash Guard', 'Dry Fit V-Nick', 'Hoodies', 'Tech Pants'],
                    ['Swimming Shirt', 'Rash Guard', 'Dry Fit V-Nick'],
                    ['Swimming Shirt', 'Rash Guard', 'Dry Fit V-Nick'],
                    ['Running Shoe', 'Swimming Shirt', 'Socks', 'Sweatshirts', 'Modern Pants', 'Soccer Shoe', 'Rash Guard', 'Hoodies', 'Tech Pants', 'Dry
                    ['Running Shoe', 'Swimming Shirt', 'Socks', 'Sweatshirts', 'Modern Pants', 'Soccer Shoe', 'Rash Guard', 'Tech Pants', 'Dry Fit V-Nick
                    ['Running Shoe', 'Swimming Shirt', 'Rash Guard', 'Tech Pants', 'Hoodies', 'Dry Fit V-Nick'],
                    ['Running Shoe', 'Swimming Shirt', 'Socks', 'Sweatshirts', 'Modern Pants', 'Dry Fit V-Nick', 'Rash Guard', 'Tech Pants'],
                    ['Swimming Shirt', 'Soccer Shoe', 'Hoodies', 'Dry Fit V-Nick', 'Tech Pants', 'Rash Guard'],
                    ['Running Shoe', 'Socks'],
                    ['Socks', 'Sweatshirts', 'Modern Pants', 'Soccer Shoe', 'Hoodies', 'Rash Guard', 'Tech Pants', 'Dry Fit V-Nick'],
                    ['Running Shoe', 'Swimming Shirt', 'Rash Guard'],
                    ['Running Shoe', 'Swimming Shirt', 'Socks', 'Sweatshirts', 'Modern Pants', 'Soccer Shoe', 'Hoodies', 'Tech Pants', 'Rash Guard', 'Dry
      }


DB3 = pd.DataFrame(data3,index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB3.to_excel('Nike.xlsx')

data4 = {'Transaction':[['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch', 'Head First J
                    ['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition' , 'Beginning Programming with Java'],
                    ['Android Programming: The Big Nerd Ranch', 'Beginning Programming with Java', 'Java 8 Pocket Guide'],
                    ['A Beginner Guide', 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
                    ['A Beginner Guide', 'Head First Java 2nd Edition' , 'Beginning Programming with Java'],
                    ['Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                    ['Java For Dummies', 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition' , 'Beginning Programming with Java'
                    ['Beginning Programming with Java', 'Java 8 Pocket Guide', 'C++ Programming in Easy Steps'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'HTML and CSS: Design and Build Websites'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Java 8 Pocket Guide', 'HTML and CSS: Design and Build w
                    ['Java For Dummies', 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
                    ['Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                    ['Head First Java 2nd Edition' , 'Beginning Programming with Java', 'Java 8 Pocket Guide'],
                    ['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
                    ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies']]
      }
```

```
DB3 = pd.DataFrame(data3,index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB3.to_excel('Nike.xlsx')

data4 = {'Transaction':[['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch', 'Head First
                        ['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition' , 'Beginning Programming with Java'],
                        ['Android Programming: The Big Nerd Ranch', 'Beginning Programming with Java', 'Java 8 Pocket Guide'],
                        ['A Beginner Guide', 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
                        ['A Beginner Guide', 'Head First Java 2nd Edition' , 'Beginning Programming with Java'],
                        ['Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                        ['Java For Dummies', 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition' , 'Beginning Programming with Java
                        ['Beginning Programming with Java', 'Java 8 Pocket Guide', 'C++ Programming in Easy Steps'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'HTML and CSS: Design and Build Websites'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Java 8 Pocket Guide', 'HTML and CSS: Design and Build
                        ['Java For Dummies', 'Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
                        ['Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies', 'Android Programming: The Big Nerd Ranch'],
                        ['Head First Java 2nd Edition' , 'Beginning Programming with Java', 'Java 8 Pocket Guide'],
                        ['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'],
                        ['A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies']]
      }


DB4 = pd.DataFrame(data4,index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB4.to_excel('Amazon.xlsx')


data5 = {'Transaction':[['A','B','C'],
                        ['A','B','C'],
                        ['A','B','C','D'],
                        ['A','B','C','D','E'],
                        ['A','B','D','E'],
                        ['A','D','E'],
                        ['A','E'],
                        ['A','E'],
                        ['A','C','E'],
                        ['A','C','E'],
                        ['A','C','E'],
                        ['A','B','C'],
                        ['A','B','C','D'],
                        ['A','B','C','D','E'],
                        ['A','B','D','E'],
                        ['A','D','E'],
                        ['A','E'],
                        ['A','E'],
                        ['A','C','E'],
                        ['A','C','E']]
       }
```

```python
DB5 = pd.DataFrame(data5, index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB5.to_excel('Generic.xlsx')


data6 = {'Transaction':[['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese'],
                        ['milk','juice'],
                        ['juice','milk','cheese'],
                        ['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese'],
                        ['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese'],
                        ['milk','juice'],
                        ['juice','milk','cheese'],
                        ['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese'],
                        ['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese'],
                        ['milk','juice'],
                        ['juice','milk','cheese'],
                        ['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese'],
                        ['ink','pen','cheese','bag'],
                        ['milk','pen','juice','cheese']]
        }

DB6 = pd.DataFrame(data6, index=['T1','T2','T3','T4','T5','T6','T7','T8','T9','T10','T11','T12','T13','T14','T15','T16','T17','T18','T19','T20'])
DB5.to_excel('Custom.xlsx')
```

Data is stored in excel files, It is read via pd.read_excel function.

Source code with snapshots:

1. Amazon database

By Apriori

a) Support = 50%, Confidence = 60%

```
#print(Combined)
```

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 1
Count of each items :

                                Element  Count
0             Head First Java 2nd Edition      8
1           C++ Programming in Easy Steps      1
2  Android Programming: The Big Nerd Ranch     13
3                         Java For Dummies     13
4            Java: The Complete Reference     10
5  HTML and CSS: Design and Build Websites      2
6                         A Beginner Guide     11
7          Beginning Programming with Java      6
8                     Java 8 Pocket Guide      4
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  50
Enter the confidence in percent:  60
Minimum support in quantity is  10
Minimum confidence is  0.6
Items with counts that satisfies the minimum support
                                Element  Count
2  Android Programming: The Big Nerd Ranch     13
3                         Java For Dummies     13
4            Java: The Complete Reference     10
6                         A Beginner Guide     11
```

```
Possible set of items meeting the minimum support
                                       Items  Support
0        [Android Programming: The Big Nerd Ranch]       13
1                           [Java For Dummies]       13
2                [Java: The Complete Reference]       10
3                           [A Beginner Guide]       11
7  [Java For Dummies, Java: The Complete Reference]       10


Frequent items are as below:

 [['Android Programming: The Big Nerd Ranch'], ['Java For Dummies'], ['Java: The Complete Reference'], ['A Beginner Guide'], [['Android Programming: Th
e Big Nerd Ranch'], ['Java For Dummies']], [['Android Programming: The Big Nerd Ranch'], ['Java: The Complete Reference']], [['Android Programming: The
Big Nerd Ranch'], ['A Beginner Guide']], [['Android Programming: The Big Nerd Ranch'], ['Java For Dummies', 'Java: The Complete Reference']], [['Java Fo
r Dummies'], ['Java: The Complete Reference']], [['Java For Dummies'], ['A Beginner Guide']], [['Java: The Complete Reference'], ['A Beginner Guide']],
[['A Beginner Guide'], ['Java For Dummies', 'Java: The Complete Reference']]]
```

```
                                          Items  Confidence  \
0  [[Android Programming: The Big Nerd Ranch], [J...    0.692308
1  [[Android Programming: The Big Nerd Ranch], [J...    0.461538
2  [[Android Programming: The Big Nerd Ranch], [A...    0.461538
3  [[Android Programming: The Big Nerd Ranch], [J...    0.461538
4  [[Java For Dummies], [Java: The Complete Refer...    0.769231
5          [[Java For Dummies], [A Beginner Guide]]    0.692308
6  [[Java: The Complete Reference], [A Beginner G...    0.900000
7  [[A Beginner Guide], [Java For Dummies, Java: ...    0.818182

   ConfidenceOfreverse
0             0.692308
1             0.600000
2             0.545455
3             0.600000
4             1.000000
5             0.818182
6             0.818182
7             0.900000
Association rules found are:

['Android Programming: The Big Nerd Ranch']  -->  ['Java For Dummies']     C = 0.69
['Java For Dummies']  -->  ['Java: The Complete Reference']     C = 0.77
['Java For Dummies']  -->  ['A Beginner Guide']     C = 0.69
['Java: The Complete Reference']  -->  ['A Beginner Guide']     C = 0.90
['A Beginner Guide']  -->  ['Java For Dummies', 'Java: The Complete Reference']     C = 0.82
['Java For Dummies']  -->  ['Android Programming: The Big Nerd Ranch']     C = 0.69
['Java: The Complete Reference']  -->  ['Android Programming: The Big Nerd Ranch']     C = 0.60
['Java For Dummies', 'Java: The Complete Reference']  -->  ['Android Programming: The Big Nerd Ranch']     C = 0.60
['Java: The Complete Reference']  -->  ['Java For Dummies']     C = 1.00
['A Beginner Guide']  -->  ['Java For Dummies']     C = 0.82
['A Beginner Guide']  -->  ['Java: The Complete Reference']     C = 0.82
['Java For Dummies', 'Java: The Complete Reference']  -->  ['A Beginner Guide']     C = 0.90
```

b) Support = 60%, Confidence = 70%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 1
Count of each items :

                                Element  Count
0             Head First Java 2nd Edition      8
1           C++ Programming in Easy Steps      1
2  Android Programming: The Big Nerd Ranch     13
3                         Java For Dummies     13
4            Java: The Complete Reference     10
5  HTML and CSS: Design and Build Websites      2
6                         A Beginner Guide     11
7          Beginning Programming with Java      6
8                      Java 8 Pocket Guide      4
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  60
Enter the confidence in percent:  70
Minimum support in quantity is  12
Minimum confidence is  0.7
Items with counts that satisfies the minimum support
                                Element  Count
2  Android Programming: The Big Nerd Ranch     13
3                         Java For Dummies     13
```

```
Possible set of items meeting the minimum support
                                Items  Support
0  [Android Programming: The Big Nerd Ranch]      13
1                      [Java For Dummies]      13


Frequent items are as below:

 [['Android Programming: The Big Nerd Ranch'], ['Java For Dummies'], [[['Android Programming: The Big Nerd Ranch'], ['Java For Dummies']]]]
```

```
                                           Items  Confidence  \
0  [[Android Programming: The Big Nerd Ranch], [J...    0.692308

   ConfidenceOfreverse
0             0.692308
No Association can be found at the given confidence
```

By Brute Force Method:

Support = 70%, Confidence = 70%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 1
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  70
Enter the confidence in percent:  70
Minimum support in quantity is 14
Minimum confidence is 0.7

Association Rules:

Execution Time: 0.0039160251617431 64 seconds
```

Support = 40%, Confidence = 40%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 1
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  40
Enter the confidence in percent:  40
Minimum support in quantity is 8
Minimum confidence is 0.4
('A Beginner Guide',) - Support: 11
('Java: The Complete Reference',) - Support: 10
('Android Programming: The Big Nerd Ranch',) - Support: 13
('Java For Dummies',) - Support: 13
('Head First Java 2nd Edition',) - Support: 8
('A Beginner Guide', 'Java: The Complete Reference') - Support: 9
('A Beginner Guide', 'Java For Dummies') - Support: 9
('Java: The Complete Reference', 'Java For Dummies') - Support: 10
('Android Programming: The Big Nerd Ranch', 'Java For Dummies') - Support: 9
('A Beginner Guide', 'Java: The Complete Reference', 'Java For Dummies') - Support: 9

Association Rules:
('A Beginner Guide',) => {'Java: The Complete Reference'} - Confidence: 0.8181818181818182
('Java: The Complete Reference',) => {'A Beginner Guide'} - Confidence: 0.9
('A Beginner Guide',) => {'Java For Dummies'} - Confidence: 0.8181818181818182
('Java For Dummies',) => {'A Beginner Guide'} - Confidence: 0.6923076923076923
('Java: The Complete Reference',) => {'Java For Dummies'} - Confidence: 1.0
('Java For Dummies',) => {'Java: The Complete Reference'} - Confidence: 0.7692307692307693
('Android Programming: The Big Nerd Ranch',) => {'Java For Dummies'} - Confidence: 0.6923076923076923
('Java For Dummies',) => {'Android Programming: The Big Nerd Ranch'} - Confidence: 0.6923076923076923
('A Beginner Guide',) => {'Java: The Complete Reference', 'Java For Dummies'} - Confidence: 0.8181818181818182
('Java: The Complete Reference',) => {'A Beginner Guide', 'Java For Dummies'} - Confidence: 0.9
('Java For Dummies',) => {'A Beginner Guide', 'Java: The Complete Reference'} - Confidence: 0.6923076923076923
('A Beginner Guide', 'Java: The Complete Reference') => {'Java For Dummies'} - Confidence: 1.0
('A Beginner Guide', 'Java For Dummies') => {'Java: The Complete Reference'} - Confidence: 1.0
('Java: The Complete Reference', 'Java For Dummies') => {'A Beginner Guide'} - Confidence: 0.9

Execution Time: 0.008970975875854492 seconds
```

2. Bestbuy Database

By Apriori

Support = 60%, Confidence = 70%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 2
Count of each items :

               Element  Count
0              Lab Top     12
1              Printer     10
2           Anti-Virus     14
3    Microsoft Office      11
4  External Hard-Drive      9
5             Speakers     11
6          Flash Drive     13
7         Lab Top Case     14
8             Desk Top      6
9       Digital Camera      9
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  60
Enter the confidence in percent:  70
Minimum support in quantity is  12
Minimum confidence is  0.7
Items with counts that satisfies the minimum support
        Element  Count
0       Lab Top     12
2    Anti-Virus     14
6   Flash Drive     13
7  Lab Top Case     14
```

```
Possible set of items meeting the minimum support
                       Items  Support
0                  [Lab Top]       12
1                [Anti-Virus]       14
2              [Flash Drive]       13
3             [Lab Top Case]       14
8  [Anti-Virus, Lab Top Case]       12


Frequent items are as below:

 [['Lab Top'], ['Anti-Virus'], ['Flash Drive'], ['Lab Top Case'], [['Lab Top'], ['Anti-Virus']], [['Lab Top'], ['Flash Drive']], [['Lab Top'], ['Lab To
p Case']], [['Lab Top'], ['Anti-Virus', 'Lab Top Case']], [['Anti-Virus'], ['Flash Drive']], [['Anti-Virus'], ['Lab Top Case']], [['Flash Drive'], ['Lab
Top Case']], [['Flash Drive'], ['Anti-Virus', 'Lab Top Case']]]
```

```
                                      Items  Confidence  \
0                    [[Lab Top], [Anti-Virus]]    0.833333
1                   [[Lab Top], [Flash Drive]]    0.583333
2                  [[Lab Top], [Lab Top Case]]    0.833333
3      [[Lab Top], [Anti-Virus, Lab Top Case]]    0.750000
4                 [[Anti-Virus], [Flash Drive]]    0.714286
5                [[Anti-Virus], [Lab Top Case]]    0.857143
6               [[Flash Drive], [Lab Top Case]]    0.692308
7  [[Flash Drive], [Anti-Virus, Lab Top Case]]    0.692308

   ConfidenceOfreverse
0             0.714286
1             0.538462
2             0.714286
3             0.750000
4             0.769231
5             0.857143
6             0.642857
7             0.750000
Association rules found are:

['Lab Top']  -->  ['Anti-Virus']     C = 0.83
['Lab Top']  -->  ['Lab Top Case']     C = 0.83
['Lab Top']  -->  ['Anti-Virus', 'Lab Top Case']     C = 0.75
['Anti-Virus']  -->  ['Flash Drive']     C = 0.71
['Anti-Virus']  -->  ['Lab Top Case']     C = 0.86
['Anti-Virus']  -->  ['Lab Top']     C = 0.71
['Lab Top Case']  -->  ['Lab Top']     C = 0.71
['Anti-Virus', 'Lab Top Case']  -->  ['Lab Top']     C = 0.75
['Flash Drive']  -->  ['Anti-Virus']     C = 0.77
['Lab Top Case']  -->  ['Anti-Virus']     C = 0.86
['Anti-Virus', 'Lab Top Case']  -->  ['Flash Drive']     C = 0.75
```

b) **Support = 50%, Confidence = 60%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 2
Count of each items :

               Element  Count
0              Lab Top     12
1              Printer     10
2           Anti-Virus     14
3     Microsoft Office     11
4  External Hard-Drive      9
5              Speakers     11
6          Flash Drive     13
7         Lab Top Case     14
8             Desk Top      6
9       Digital Camera      9
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  50
Enter the confidence in percent:  60
Minimum support in quantity is  10
Minimum confidence is  0.6
Items with counts that satisfies the minimum support
            Element  Count
0           Lab Top     12
1           Printer     10
2        Anti-Virus     14
3  Microsoft Office     11
5          Speakers     11
6       Flash Drive     13
7      Lab Top Case     14
```

Possible set of items meeting the minimum support

	Items	Support
0	[Lab Top]	12
1	[Printer]	10
2	[Anti-Virus]	14
3	[Microsoft Office]	11
4	[Speakers]	11
5	[Flash Drive]	13
6	[Lab Top Case]	14
8	[Lab Top, Anti-Virus]	10
12	[Lab Top, Lab Top Case]	10
16	[Printer, Flash Drive]	10
20	[Anti-Virus, Flash Drive]	10
21	[Anti-Virus, Lab Top Case]	12
23	[Microsoft Office, Flash Drive]	11

Frequent items are as below:

[['Lab Top'], ['Printer'], ['Anti-Virus'], ['Microsoft Office'], ['Speakers'], ['Flash Drive'], ['Lab Top Case'], [['Lab Top'], ['Printer']], [['Lab Top'], ['Anti-Virus']], [['Lab Top'], ['Microsoft Office']], [['Lab Top'], ['Speakers']], [['Lab Top'], ['Flash Drive']], [['Lab Top'], ['Lab Top Case']], [['Lab Top'], ['Printer', 'Flash Drive']], [['Lab Top'], ['Anti-Virus', 'Flash Drive']], [['Lab Top'], ['Anti-Virus', 'Lab Top Case']], [['Lab Top'], ['Microsoft Office', 'Flash Drive']], [['Printer'], ['Anti-Virus']], [['Printer'], ['Microsoft Office']], [['Printer'], ['Speakers']], [['Printer'], ['Flash Drive']], [['Printer'], ['Lab Top Case']], [['Printer'], ['Lab Top', 'Anti-Virus']], [['Printer'], ['Lab Top', 'Lab Top Case']], [['Printer'], ['Anti-Virus', 'Flash Drive']], [['Printer'], ['Anti-Virus', 'Lab Top Case']], [['Printer'], ['Microsoft Office', 'Flash Drive']], [['Anti-Virus'], ['Microsoft Office']], [['Anti-Virus'], ['Speakers']], [['Anti-Virus'], ['Flash Drive']], [['Anti-Virus'], ['Lab Top Case']], [['Anti-Virus'], ['Lab Top', 'Lab Top Case']], [['Anti-Virus'], ['Printer', 'Flash Drive']], [['Anti-Virus'], ['Microsoft Office', 'Flash Drive']], [['Microsoft Office'], ['Speakers']], [['Microsoft Office'], ['Flash Drive']], [['Microsoft Office'], ['Lab Top Case']], [['Microsoft Office'], ['Lab Top', 'Anti-Virus']], [['Microsoft Office'], ['Lab Top', 'Lab Top Case']], [['Microsoft Office'], ['Printer', 'Flash Drive']], [['Microsoft Office'], ['Anti-Virus', 'Flash Drive']], [['Microsoft Office'], ['Anti-Virus', 'Lab Top Case']], [['Speakers'], ['Flash Drive']], [['Speakers'], ['Lab Top Case']], [['Speakers'], ['Lab Top', 'Anti-Virus']], [['Speakers'], ['Lab Top', 'Lab Top Case']], [['Speakers'], ['Printer', 'Flash Drive']], [['Speakers'], ['Anti-Virus', 'Flash Drive']], [['Speakers'], ['Anti-Virus', 'Lab Top Case']], [['Speakers'], ['Microsoft Office', 'Flash Drive']], [['Flash Drive'], ['Lab Top Case']], [['Flash Drive'], ['Lab Top', 'Anti-Virus']], [['Flash Drive'], ['Lab Top', 'Lab Top Case']], [['Flash Drive'], ['Anti-Virus', 'Lab Top Case']], [['Lab Top Case'], ['Lab Top', 'Anti-Virus']], [['Lab Top Case'], ['Printer', 'Flash Drive']], [['Lab Top Case'], ['Anti-Virus', 'Flash Drive']], [['Lab Top Case'], ['Microsoft Office', 'Flash Drive']], [['Lab Top', 'Anti-Virus'], ['Printer', 'Flash Drive']], [['Lab Top', 'Anti-Virus'], ['Microsoft Office', 'Flash Drive']], [['Lab Top', 'Lab Top Case'], ['Printer', 'Flash Drive']], [['Lab Top', 'Lab Top Case'], ['Anti-Virus', 'Flash Drive']], [['Lab Top', 'Lab Top Case'], ['Microsoft Office', 'Flash Drive']], [['Printer', 'Flash Drive'], ['Anti-Virus', 'Lab Top Case']], [['Anti-Virus', 'Lab Top Case'], ['Microsoft Office'. 'Flash Drive']]]


```
['Printer', 'Flash Drive']  -->  ['Anti-Virus']     C = 0.70
['Microsoft Office', 'Flash Drive']  -->  ['Anti-Virus']     C = 0.73
['Flash Drive']  -->  ['Microsoft Office']     C = 0.85
['Printer', 'Flash Drive']  -->  ['Microsoft Office']     C = 0.90
['Anti-Virus', 'Flash Drive']  -->  ['Microsoft Office']     C = 0.80
['Lab Top Case']  -->  ['Speakers']     C = 0.64
['Anti-Virus', 'Flash Drive']  -->  ['Speakers']     C = 0.60
['Anti-Virus', 'Lab Top Case']  -->  ['Speakers']     C = 0.67
['Lab Top Case']  -->  ['Flash Drive']     C = 0.64
['Lab Top', 'Anti-Virus']  -->  ['Flash Drive']     C = 0.60
['Lab Top', 'Lab Top Case']  -->  ['Flash Drive']     C = 0.60
['Anti-Virus', 'Lab Top Case']  -->  ['Flash Drive']     C = 0.75
['Lab Top', 'Anti-Virus']  -->  ['Lab Top Case']     C = 0.90
['Printer', 'Flash Drive']  -->  ['Lab Top Case']     C = 0.60
['Anti-Virus', 'Flash Drive']  -->  ['Lab Top Case']     C = 0.90
['Microsoft Office', 'Flash Drive']  -->  ['Lab Top Case']     C = 0.64
['Anti-Virus', 'Flash Drive']  -->  ['Lab Top', 'Lab Top Case']     C = 0.60
['Microsoft Office', 'Flash Drive']  -->  ['Anti-Virus', 'Lab Top Case']     C = 0.64
```

By Brute Force Method:

a) Support =40%, Confidence = 50%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 2
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  40
Enter the confidence in percent:  50
Minimum support in quantity is 8
Minimum confidence is 0.5

Association Rules:
('Lab Top',) => {'Lab Top Case'} - Confidence: 0.8333333333333334
('Lab Top Case',) => {'Lab Top'} - Confidence: 0.7142857142857143
('Lab Top',) => {'Anti-Virus'} - Confidence: 0.8333333333333334
('Anti-Virus',) => {'Lab Top'} - Confidence: 0.7142857142857143
('Lab Top Case',) => {'Flash Drive'} - Confidence: 0.6428571428571429
('Flash Drive',) => {'Lab Top Case'} - Confidence: 0.6923076923076923
('Lab Top Case',) => {'Speakers'} - Confidence: 0.6428571428571429
('Speakers',) => {'Lab Top Case'} - Confidence: 0.8181818181818182
('Lab Top Case',) => {'Anti-Virus'} - Confidence: 0.8571428571428571
('Anti-Virus',) => {'Lab Top Case'} - Confidence: 0.8571428571428571
('Lab Top Case',) => {'External Hard-Drive'} - Confidence: 0.5714285714285714
('External Hard-Drive',) => {'Lab Top Case'} - Confidence: 0.8888888888888888
('Flash Drive',) => {'Anti-Virus'} - Confidence: 0.7692307692307693
('Anti-Virus',) => {'Flash Drive'} - Confidence: 0.7142857142857143
('Flash Drive',) => {'Microsoft Office'} - Confidence: 0.8461538461538461
('Microsoft Office',) => {'Flash Drive'} - Confidence: 1.0
('Flash Drive',) => {'Printer'} - Confidence: 0.7692307692307693
('Printer',) => {'Flash Drive'} - Confidence: 1.0
('Speakers',) => {'Anti-Virus'} - Confidence: 0.8181818181818182
('Anti-Virus',) => {'Speakers'} - Confidence: 0.6428571428571429
('Anti-Virus',) => {'Microsoft Office'} - Confidence: 0.5714285714285714
('Microsoft Office',) => {'Anti-Virus'} - Confidence: 0.7272727272727273
('Anti-Virus',) => {'External Hard-Drive'} - Confidence: 0.6428571428571429
('External Hard-Drive',) => {'Anti-Virus'} - Confidence: 1.0
('Microsoft Office',) => {'Printer'} - Confidence: 0.8181818181818182
('Printer',) => {'Microsoft Office'} - Confidence: 0.9
('Lab Top',) => {'Lab Top Case', 'Anti-Virus'} - Confidence: 0.75
('Lab Top Case',) => {'Lab Top', 'Anti-Virus'} - Confidence: 0.6428571428571429
('Anti-Virus',) => {'Lab Top', 'Lab Top Case'} - Confidence: 0.6428571428571429
('Lab Top', 'Lab Top Case') => {'Anti-Virus'} - Confidence: 0.9
('Lab Top', 'Anti-Virus') => {'Lab Top Case'} - Confidence: 0.9
('Lab Top Case', 'Anti-Virus') => {'Lab Top'} - Confidence: 0.75
('Lab Top Case',) => {'Flash Drive', 'Anti-Virus'} - Confidence: 0.6428571428571429
('Flash Drive',) => {'Lab Top Case', 'Anti-Virus'} - Confidence: 0.6923076923076923
('Anti-Virus',) => {'Lab Top Case', 'Flash Drive'} - Confidence: 0.6428571428571429
('Lab Top Case', 'Flash Drive') => {'Anti-Virus'} - Confidence: 1.0
('Lab Top Case', 'Anti-Virus') => {'Flash Drive'} - Confidence: 0.75
('Flash Drive', 'Anti-Virus') => {'Lab Top Case'} - Confidence: 0.9
('Lab Top Case',) => {'Speakers', 'Anti-Virus'} - Confidence: 0.5714285714285714
('Speakers',) => {'Lab Top Case', 'Anti-Virus'} - Confidence: 0.7272727272727273
('Anti-Virus',) => {'Speakers', 'Lab Top Case'} - Confidence: 0.5714285714285714
('Lab Top Case', 'Speakers') => {'Anti-Virus'} - Confidence: 0.8888888888888888
('Lab Top Case', 'Anti-Virus') => {'Speakers'} - Confidence: 0.6666666666666666
('Speakers', 'Anti-Virus') => {'Lab Top Case'} - Confidence: 0.8888888888888888
('Lab Top Case',) => {'External Hard-Drive', 'Anti-Virus'} - Confidence: 0.5714285714285714
('Anti-Virus',) => {'External Hard-Drive', 'Lab Top Case'} - Confidence: 0.5714285714285714
('External Hard-Drive',) => {'Lab Top Case', 'Anti-Virus'} - Confidence: 0.8888888888888888
('Lab Top Case', 'Anti-Virus') => {'External Hard-Drive'} - Confidence: 0.6666666666666666
('Lab Top Case', 'External Hard-Drive') => {'Anti-Virus'} - Confidence: 1.0
('Anti-Virus', 'External Hard-Drive') => {'Lab Top Case'} - Confidence: 0.8888888888888888
('Flash Drive',) => {'Microsoft Office', 'Anti-Virus'} - Confidence: 0.6153846153846154
('Anti-Virus',) => {'Microsoft Office', 'Flash Drive'} - Confidence: 0.5714285714285714
('Microsoft Office',) => {'Flash Drive', 'Anti-Virus'} - Confidence: 0.7272727272727273
('Flash Drive', 'Anti-Virus') => {'Microsoft Office'} - Confidence: 0.8
('Flash Drive', 'Microsoft Office') => {'Anti-Virus'} - Confidence: 0.7272727272727273
('Anti-Virus', 'Microsoft Office') => {'Flash Drive'} - Confidence: 1.0
('Flash Drive',) => {'Microsoft Office', 'Printer'} - Confidence: 0.6923076923076923
('Microsoft Office',) => {'Printer', 'Flash Drive'} - Confidence: 0.8181818181818182
('Printer',) => {'Microsoft Office', 'Flash Drive'} - Confidence: 0.9
```

```
('Microsoft Office',) => {'Printer', 'Flash Drive'} - Confidence: 0.8181818181818182
('Printer',) => {'Microsoft Office', 'Flash Drive'} - Confidence: 0.9
('Flash Drive', 'Microsoft Office') => {'Printer'} - Confidence: 0.8181818181818182
('Flash Drive', 'Printer') => {'Microsoft Office'} - Confidence: 0.9
('Microsoft Office', 'Printer') => {'Flash Drive'} - Confidence: 1.0

Execution Time: 0.01567220687866211 seconds
```

b) Support = 70%, Confidence = 80%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 2
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  70
Enter the confidence in percent:  80
Minimum support in quantity is 14
Minimum confidence is 0.8

Association Rules:

Execution Time: 0.0159454345703125 seconds
```

3. Kmart Database

By Apriori Method

a) Support = 70%, Confidence = 80%

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 3
Count of each items :

                Element  Count
0                 Shams     11
1            Bed Skirts     11
2            Bedspreads      7
3  Embroidered Bedspread      6
4                Sheets     10
5    Bedding Collections      7
6          Kids Bedding     12
7     Decorative Pillows     10
8                Quilts      8
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  70
Enter the confidence in percent:  80
Minimum support in quantity is  14
Minimum confidence is  0.8
Items with counts that satisfies the minimum support
Empty DataFrame
Columns: [Element, Count]
Index: []
Possible set of items meeting the minimum support
Empty DataFrame
Columns: [Items, Support]
Index: []



Frequent items are as below:

 Series([], Name: Items, dtype: float64)
Empty DataFrame
Columns: [Items, Confidence, ConfidenceOfreverse]
Index: []
No Association found between items with given support and confidence, Try with lesser support or confidence


No Association found between items with given support and confidence, Try with lesser support and confidence
```

b) **Support = 50%, Confidence = 60%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 3
Count of each items :

                 Element  Count
0                  Shams     11
1             Bed Skirts     11
2             Bedspreads      7
3  Embroidered Bedspread      6
4                 Sheets     10
5    Bedding Collections      7
6           Kids Bedding     12
7     Decorative Pillows     10
8                 Quilts      8
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  50
Enter the confidence in percent:  60
Minimum support in quantity is  10
Minimum confidence is  0.6
Items with counts that satisfies the minimum support
              Element  Count
0               Shams     11
1          Bed Skirts     11
4              Sheets     10
6        Kids Bedding     12
7  Decorative Pillows     10
Possible set of items meeting the minimum support
                         Items  Support
0                      [Shams]       11
1                 [Bed Skirts]       11
2                     [Sheets]       10
3               [Kids Bedding]       12
4         [Decorative Pillows]       10
10  [Bed Skirts, Kids Bedding]       10
12      [Sheets, Kids Bedding]       10
```

```
Frequent items are as below:

 [['Shams'], ['Bed Skirts'], ['Sheets'], ['Kids Bedding'], ['Decorative Pillows'], [['Shams'], ['Bed Skirts']], [['Shams'],
['Kids Bedding']], [['Shams'], ['Decorative Pillows']], [['Shams'], ['Bed Skirts', 'Kids Bedding']], [['Shams'], ['Sheets', '
rts'], ['Sheets']], [['Bed Skirts'], ['Kids Bedding']], [['Bed Skirts'], ['Decorative Pillows']], [['Bed Skirts'], ['Sheets',
s'], ['Kids Bedding']], [['Sheets'], ['Decorative Pillows']], [['Sheets'], ['Bed Skirts', 'Kids Bedding']], [['Kids Bedding']
[['Decorative Pillows'], ['Bed Skirts', 'Kids Bedding']], [['Decorative Pillows'], ['Sheets', 'Kids Bedding']]]
                                          Items  Confidence  \
0                            [[Shams], [Bed Skirts]]    0.818182
1                               [[Shams], [Sheets]]    0.636364
2                           [[Shams], [Kids Bedding]]    0.818182
3                     [[Shams], [Decorative Pillows]]    0.363636
4               [[Shams], [Bed Skirts, Kids Bedding]]    0.727273
5                   [[Shams], [Sheets, Kids Bedding]]    0.636364
6                           [[Bed Skirts], [Sheets]]    0.818182
7                     [[Bed Skirts], [Kids Bedding]]    0.909091
8                [[Bed Skirts], [Decorative Pillows]]    0.363636
9             [[Bed Skirts], [Sheets, Kids Bedding]]    0.818182
10                       [[Sheets], [Kids Bedding]]    1.000000
11                 [[Sheets], [Decorative Pillows]]    0.200000
12            [[Sheets], [Bed Skirts, Kids Bedding]]    0.900000
13            [[Kids Bedding], [Decorative Pillows]]    0.333333
14  [[Decorative Pillows], [Bed Skirts, Kids Beddi...    0.300000
15     [[Decorative Pillows], [Sheets, Kids Bedding]]    0.200000

    ConfidenceOfreverse
0              0.818182
1              0.700000
2              0.750000
3              0.400000
4              0.800000
5              0.700000
6              0.900000
7              0.833333
8              0.400000
9              0.900000
10             0.833333
11             0.200000
12             0.900000
13             0.400000
14             0.300000
15             0.200000
Association rules found are:

['Shams']  -->  ['Bed Skirts']     C = 0.82
['Shams']  -->  ['Sheets']     C = 0.64
['Shams']  -->  ['Kids Bedding']     C = 0.82
['Shams']  -->  ['Bed Skirts', 'Kids Bedding']     C = 0.73
['Shams']  -->  ['Sheets', 'Kids Bedding']     C = 0.64
['Bed Skirts']  -->  ['Sheets']     C = 0.82
['Bed Skirts']  -->  ['Kids Bedding']     C = 0.91
['Bed Skirts']  -->  ['Sheets', 'Kids Bedding']     C = 0.82
['Sheets']  -->  ['Kids Bedding']     C = 1.00
['Sheets']  -->  ['Bed Skirts', 'Kids Bedding']     C = 0.90
['Bed Skirts']  -->  ['Shams']     C = 0.82
['Sheets']  -->  ['Shams']     C = 0.70
['Kids Bedding']  -->  ['Shams']     C = 0.75
['Bed Skirts', 'Kids Bedding']  -->  ['Shams']     C = 0.80
['Sheets', 'Kids Bedding']  -->  ['Shams']     C = 0.70
['Sheets']  -->  ['Bed Skirts']     C = 0.90
['Kids Bedding']  -->  ['Bed Skirts']     C = 0.83
['Sheets', 'Kids Bedding']  -->  ['Bed Skirts']     C = 0.90
```

By Brute Force Method:

a)**Support = 40%, Confidence = 80%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 3
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  40
Enter the confidence in percent:  80
Minimum support in quantity is 8
Minimum confidence is 0.8

Association Rules:
('Sheets',) => {'Bed Skirts'} - Confidence: 0.9
('Bed Skirts',) => {'Sheets'} - Confidence: 0.8181818181818182
('Sheets',) => {'Kids Bedding'} - Confidence: 1.0
('Kids Bedding',) => {'Sheets'} - Confidence: 0.8333333333333334
('Shams',) => {'Bed Skirts'} - Confidence: 0.8181818181818182
('Bed Skirts',) => {'Shams'} - Confidence: 0.8181818181818182
('Shams',) => {'Kids Bedding'} - Confidence: 0.8181818181818182
('Bed Skirts',) => {'Kids Bedding'} - Confidence: 0.9090909090909091
('Kids Bedding',) => {'Bed Skirts'} - Confidence: 0.8333333333333334
('Sheets',) => {'Bed Skirts', 'Kids Bedding'} - Confidence: 0.9
('Bed Skirts',) => {'Sheets', 'Kids Bedding'} - Confidence: 0.8181818181818182
('Sheets', 'Bed Skirts') => {'Kids Bedding'} - Confidence: 1.0
('Sheets', 'Kids Bedding') => {'Bed Skirts'} - Confidence: 0.9
('Bed Skirts', 'Kids Bedding') => {'Sheets'} - Confidence: 0.9
('Shams', 'Bed Skirts') => {'Kids Bedding'} - Confidence: 0.8888888888888888
('Shams', 'Kids Bedding') => {'Bed Skirts'} - Confidence: 0.8888888888888888
('Bed Skirts', 'Kids Bedding') => {'Shams'} - Confidence: 0.8

Execution Time: 0.007974863052368164 seconds
```

4. Nike Database

By Apriori

- **Support = 70%, Confidence = 80%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 4
Count of each items :

          Element  Count
0  Swimming Shirt     11
1     Soccer Shoe      6
2     Sweatshirts     13
3         Hoodies      8
4      Rash Guard     12
5           Socks     13
6  Dry Fit V-Nick     10
7    Running Shoe     14
8      Tech Pants      9
9    Modern Pants     10
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  70
Enter the confidence in percent:  80
Minimum support in quantity is  14
Minimum confidence is  0.8
Items with counts that satisfies the minimum support
        Element  Count
7  Running Shoe     14
Possible set of items meeting the minimum support
            Items  Support
0  [Running Shoe]       14



Frequent items are as below:

 0    [Running Shoe]
Name: Items, dtype: object
Empty DataFrame
Columns: [Items, Confidence, ConfidenceOfreverse]
Index: []
No Association found between items with given support and confidence, Try with lesser support or confidence
```

- **Support = 60%, Confidence = 75%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 4
Count of each items :

          Element  Count
0  Swimming Shirt     11
1     Soccer Shoe      6
2     Sweatshirts     13
3         Hoodies      8
4      Rash Guard     12
5           Socks     13
6  Dry Fit V-Nick     10
7    Running Shoe     14
8      Tech Pants      9
9    Modern Pants     10
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  60
Enter the confidence in percent:  75
Minimum support in quantity is  12
Minimum confidence is  0.75
Items with counts that satisfies the minimum support
        Element  Count
2   Sweatshirts     13
4    Rash Guard     12
5         Socks     13
7  Running Shoe     14
Possible set of items meeting the minimum support
                 Items  Support
0         [Sweatshirts]       13
1          [Rash Guard]       12
2               [Socks]       13
3        [Running Shoe]       14
5  [Sweatshirts, Socks]       12



Frequent items are as below:

 [['Sweatshirts'], ['Rash Guard'], ['Socks'], ['Running Shoe'], [['Sweatshirts'], ['Rash Guard']], [['Sweatshirts'], ['Socks']]
nning Shoe']], [['Rash Guard'], ['Socks']], [['Rash Guard'], ['Running Shoe']], [['Rash Guard'], ['Sweatshirts', 'Socks']], [['S
e']], [['Running Shoe'], ['Sweatshirts', 'Socks']]]]
                                  Items  Confidence  ConfidenceOfreverse
0           [[Sweatshirts], [Rash Guard]]    0.461538             0.500000
1                [[Sweatshirts], [Socks]]    0.923077             0.923077
2         [[Sweatshirts], [Running Shoe]]    0.846154             0.785714
3                 [[Rash Guard], [Socks]]    0.500000             0.461538
4          [[Rash Guard], [Running Shoe]]    0.583333             0.500000
5    [[Rash Guard], [Sweatshirts, Socks]]    0.500000             0.500000
6               [[Socks], [Running Shoe]]    0.846154             0.785714
7  [[Running Shoe], [Sweatshirts, Socks]]    0.714286             0.833333
Association rules found are:

['Sweatshirts']  -->  ['Socks']     C = 0.92
['Sweatshirts']  -->  ['Running Shoe']     C = 0.85
['Socks']  -->  ['Running Shoe']     C = 0.85
['Socks']  -->  ['Sweatshirts']     C = 0.92
['Running Shoe']  -->  ['Sweatshirts']     C = 0.79
['Running Shoe']  -->  ['Socks']     C = 0.79
['Sweatshirts', 'Socks']  -->  ['Running Shoe']     C = 0.83
```

By Brute Force Method

- **Support = 55%, Confidence = 80%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 4
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  55
Enter the confidence in percent:  80
Minimum support in quantity is 11
Minimum confidence is 0.8

Association Rules:
('Sweatshirts',) => {'Running Shoe'} - Confidence: 0.8461538461538461
('Socks',) => {'Running Shoe'} - Confidence: 0.8461538461538461
('Sweatshirts',) => {'Socks'} - Confidence: 0.9230769230769231
('Socks',) => {'Sweatshirts'} - Confidence: 0.9230769230769231

Execution Time: 0.013894796371459961 seconds
```

- **Support = 40%, Confidence = 40%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 4
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  40
Enter the confidence in percent:  40
Minimum support in quantity is 8
Minimum confidence is 0.4

Association Rules:
('Running Shoe',) => {'Modern Pants'} - Confidence: 0.6428571428571429
('Modern Pants',) => {'Running Shoe'} - Confidence: 0.9
('Running Shoe',) => {'Sweatshirts'} - Confidence: 0.7857142857142857
('Sweatshirts',) => {'Running Shoe'} - Confidence: 0.8461538461538461
('Running Shoe',) => {'Socks'} - Confidence: 0.7857142857142857
('Socks',) => {'Running Shoe'} - Confidence: 0.8461538461538461
('Dry Fit V-Nick',) => {'Rash Guard'} - Confidence: 1.0
('Rash Guard',) => {'Dry Fit V-Nick'} - Confidence: 0.8333333333333334
('Dry Fit V-Nick',) => {'Tech Pants'} - Confidence: 0.8
('Tech Pants',) => {'Dry Fit V-Nick'} - Confidence: 0.8888888888888888
('Dry Fit V-Nick',) => {'Swimming Shirt'} - Confidence: 0.9
('Swimming Shirt',) => {'Dry Fit V-Nick'} - Confidence: 0.8181818181818182
('Hoodies',) => {'Rash Guard'} - Confidence: 1.0
('Rash Guard',) => {'Hoodies'} - Confidence: 0.6666666666666666
('Hoodies',) => {'Tech Pants'} - Confidence: 1.0
('Tech Pants',) => {'Hoodies'} - Confidence: 0.8888888888888888
('Modern Pants',) => {'Sweatshirts'} - Confidence: 1.0
('Sweatshirts',) => {'Modern Pants'} - Confidence: 0.7692307692307693
('Modern Pants',) => {'Socks'} - Confidence: 0.9
('Socks',) => {'Modern Pants'} - Confidence: 0.6923076923076923
('Rash Guard',) => {'Tech Pants'} - Confidence: 0.75
('Tech Pants',) => {'Rash Guard'} - Confidence: 1.0
('Rash Guard',) => {'Swimming Shirt'} - Confidence: 0.8333333333333334
('Swimming Shirt',) => {'Rash Guard'} - Confidence: 0.9090909090909091
('Sweatshirts',) => {'Socks'} - Confidence: 0.9230769230769231
('Socks',) => {'Sweatshirts'} - Confidence: 0.9230769230769231
('Running Shoe',) => {'Modern Pants', 'Sweatshirts'} - Confidence: 0.6428571428571429
('Modern Pants',) => {'Running Shoe', 'Sweatshirts'} - Confidence: 0.9
('Sweatshirts',) => {'Modern Pants', 'Running Shoe'} - Confidence: 0.6923076923076923
('Running Shoe', 'Modern Pants') => {'Sweatshirts'} - Confidence: 1.0
('Running Shoe', 'Sweatshirts') => {'Modern Pants'} - Confidence: 0.8181818181818182
('Modern Pants', 'Sweatshirts') => {'Running Shoe'} - Confidence: 0.9
('Running Shoe',) => {'Modern Pants', 'Socks'} - Confidence: 0.5714285714285714
('Modern Pants',) => {'Running Shoe', 'Socks'} - Confidence: 0.8
('Socks',) => {'Modern Pants', 'Running Shoe'} - Confidence: 0.6153846153846154
('Running Shoe', 'Modern Pants') => {'Socks'} - Confidence: 0.8888888888888888
('Running Shoe', 'Socks') => {'Modern Pants'} - Confidence: 0.7272727272727273
('Modern Pants', 'Socks') => {'Running Shoe'} - Confidence: 0.8888888888888888
('Running Shoe',) => {'Sweatshirts', 'Socks'} - Confidence: 0.7142857142857143
('Sweatshirts',) => {'Running Shoe', 'Socks'} - Confidence: 0.7692307692307693
('Socks',) => {'Running Shoe', 'Sweatshirts'} - Confidence: 0.7692307692307693
('Running Shoe', 'Sweatshirts') => {'Socks'} - Confidence: 0.9090909090909091
('Running Shoe', 'Socks') => {'Sweatshirts'} - Confidence: 0.9090909090909091
```

```
('Running Shoe', 'Modern Pants') => {'Sweatshirts', 'Socks'} - Confidence: 0.8888888888888888
('Running Shoe', 'Sweatshirts') => {'Modern Pants', 'Socks'} - Confidence: 0.7272727272727273
('Running Shoe', 'Socks') => {'Modern Pants', 'Sweatshirts'} - Confidence: 0.7272727272727273
('Modern Pants', 'Sweatshirts') => {'Running Shoe', 'Socks'} - Confidence: 0.8
('Modern Pants', 'Socks') => {'Running Shoe', 'Sweatshirts'} - Confidence: 0.8888888888888888
('Sweatshirts', 'Socks') => {'Modern Pants', 'Running Shoe'} - Confidence: 0.6666666666666666
('Running Shoe', 'Modern Pants', 'Sweatshirts') => {'Socks'} - Confidence: 0.8888888888888888
('Running Shoe', 'Modern Pants', 'Socks') => {'Sweatshirts'} - Confidence: 1.0
('Running Shoe', 'Sweatshirts', 'Socks') => {'Modern Pants'} - Confidence: 0.8
('Modern Pants', 'Sweatshirts', 'Socks') => {'Running Shoe'} - Confidence: 0.8888888888888888

Execution Time: 0.0191347599029541 seconds
```

5. Generic Database
By Apriori Method:

- **Support = 55%, Confidence = 80%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 5
Count of each items :

  Element  Count
0       A     20
1       E     15
2       D      8
3       C     12
4       B      9
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  55
Enter the confidence in percent:  80
Minimum support in quantity is  11
Minimum confidence is  0.8
Items with counts that satisfies the minimum support
  Element  Count
0       A     20
1       E     15
3       C     12
Possible set of items meeting the minimum support
    Items  Support
0     [A]       20
1     [E]       15
2     [C]       12
3  [A, E]       15
4  [A, C]       12



Frequent items are as below:

 [['A'], ['E'], ['C'], [['A'], ['E']], [['A'], ['C']], [['E'], ['C']], [['E'], ['A', 'C']], [['C'], ['A', 'E']]]
          Items  Confidence  ConfidenceOfreverse
0     [[A], [E]]    0.750000             1.000000
1     [[A], [C]]    0.600000             1.000000
2     [[E], [C]]    0.466667             0.583333
3  [[E], [A, C]]    0.466667             0.583333
4  [[C], [A, E]]    0.583333             0.466667
Association rules found are:

['E']  -->  ['A']     C = 1.00
['C']  -->  ['A']     C = 1.00
```

- **Support = 60%, Confidence = 85%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 5
Count of each items :

  Element  Count
0       A     20
1       E     15
2       D      8
3       C     12
4       B      9
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  60
Enter the confidence in percent:  85
Minimum support in quantity is  12
Minimum confidence is  0.85
Items with counts that satisfies the minimum support
  Element  Count
0       A     20
1       E     15
3       C     12
Possible set of items meeting the minimum support
    Items  Support
0     [A]       20
1     [E]       15
2     [C]       12
3  [A, E]       15
4  [A, C]       12



Frequent items are as below:

 [['A'], ['E'], ['C'], [['A'], ['E']], [['A'], ['C']], [['E'], ['C']], [['E'], ['A', 'C']], [['C'], ['A', 'E']]]
           Items  Confidence  ConfidenceOfreverse
0     [[A], [E]]    0.750000             1.000000
1     [[A], [C]]    0.600000             1.000000
2     [[E], [C]]    0.466667             0.583333
3  [[E], [A, C]]    0.466667             0.583333
4  [[C], [A, E]]    0.583333             0.466667
Association rules found are:

['E']  -->  ['A']     C = 1.00
['C']  -->  ['A']     C = 1.00
```

By Brute Force Method:

- **Support = 50%, Confidence = 50%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 5
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  50
Enter the confidence in percent:  50
Minimum support in quantity is 10
Minimum confidence is 0.5

Association Rules:
('A',) => {'C'} - Confidence: 0.6
('C',) => {'A'} - Confidence: 1.0
('A',) => {'E'} - Confidence: 0.75
('E',) => {'A'} - Confidence: 1.0

Execution Time: 0.0 seconds
```

- **Support = 40%, Confidence = 40%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 5
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  40
Enter the confidence in percent:  40
Minimum support in quantity is 8
Minimum confidence is 0.4

Association Rules:
('A',) => {'D'} - Confidence: 0.4
('D',) => {'A'} - Confidence: 1.0
('A',) => {'B'} - Confidence: 0.45
('B',) => {'A'} - Confidence: 1.0
('A',) => {'C'} - Confidence: 0.6
('C',) => {'A'} - Confidence: 1.0
('A',) => {'E'} - Confidence: 0.75
('E',) => {'A'} - Confidence: 1.0

Execution Time: 0.0009398460388183594 seconds
```

6. Custom Database:
By Apriori Method:

- **Support = 70%, Confidence = 80%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 6
Count of each items :

  Element  Count
0       A     20
1       E     15
2       D      8
3       C     12
4       B      9
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  70
Enter the confidence in percent:  80
Minimum support in quantity is  14
Minimum confidence is  0.8
Items with counts that satisfies the minimum support
  Element  Count
0       A     20
1       E     15
Possible set of items meeting the minimum support
    Items  Support
0     [A]       20
1     [E]       15
2  [A, E]       15



Frequent items are as below:

 [['A'], ['E'], [['A'], ['E']]]
        Items  Confidence  ConfidenceOfreverse
0  [[A], [E]]        0.75                  1.0
Association rules found are:

['E']  -->  ['A']     C = 1.00
```

- **Support = 60%, Confidence = 30%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 6
Count of each items :

  Element  Count
0       A     20
1       E     15
2       D      8
3       C     12
4       B      9
This is a database of 20 transactions.Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  60
Enter the confidence in percent:  30
Minimum support in quantity is  12
Minimum confidence is  0.3
Items with counts that satisfies the minimum support
  Element  Count
0       A     20
1       E     15
3       C     12
Possible set of items meeting the minimum support
    Items  Support
0     [A]       20
1     [E]       15
2     [C]       12
3  [A, E]       15
4  [A, C]       12



Frequent items are as below:

 [['A'], ['E'], ['C'], [['A'], ['E']], [['A'], ['C']], [['E'], ['C']], [['E'], ['A', 'C']], [['C'], ['A', 'E']]]
          Items  Confidence  ConfidenceOfreverse
0     [[A], [E]]    0.750000             1.000000
1     [[A], [C]]    0.600000             1.000000
2     [[E], [C]]    0.466667             0.583333
3  [[E], [A, C]]    0.466667             0.583333
4  [[C], [A, E]]    0.583333             0.466667
Association rules found are:

['A']  -->  ['E']     C = 0.75
['A']  -->  ['C']     C = 0.60
['E']  -->  ['C']     C = 0.47
['E']  -->  ['A', 'C']     C = 0.47
['C']  -->  ['A', 'E']     C = 0.58
['E']  -->  ['A']     C = 1.00
['C']  -->  ['A']     C = 1.00
['C']  -->  ['E']     C = 0.58
['A', 'C']  -->  ['E']     C = 0.58
['A', 'E']  -->  ['C']     C = 0.47
```

By Brute Force Method:

- **Support = 40%, Confidence = 50%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 6
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  40
Enter the confidence in percent:  50
Minimum support in quantity is 8
Minimum confidence is 0.5

Association Rules:
('D',) => {'A'} - Confidence: 1.0
('B',) => {'A'} - Confidence: 1.0
('A',) => {'C'} - Confidence: 0.6
('C',) => {'A'} - Confidence: 1.0
('A',) => {'E'} - Confidence: 0.75
('E',) => {'A'} - Confidence: 1.0

Execution Time: 0.0009813308715820312 seconds
```

- **Support =50%, Confidence = 60%**

```
Please select one out of 6 databases
Enter 1 for Amazon
Enter 2 for BestBuy
Enter 3 for kmart
Enter 4 for Nike
Enter 5 for Generic
Enter 6 for Custom.
 6
This is a database of 20 transactions. Enter any value of support and confidence between 10 to 100%
Enter the support in percent:  50
Enter the confidence in percent:  60
Minimum support in quantity is 10
Minimum confidence is 0.6

Association Rules:
('A',) => {'C'} - Confidence: 0.6
('C',) => {'A'} - Confidence: 1.0
('A',) => {'E'} - Confidence: 0.75
('E',) => {'A'} - Confidence: 1.0

Execution Time: 0.0 seconds
```

How to run the python program:

There are two python programs that need to be run, Database.py to create the database locally and AprioriCode.py to implement the apriori algorithm on those 5 databases.

Step1:
Run the Database.py to store the 5 databases in your local pc/laptop.

Step2:
Run the AprioriCode.py in the same current working directory so that it can access the database easily and it will print all the output in the console.

Step3:
Run the Brute.py in the same current working directory so that it can access the database easily and it will print all the output in the console.

Github link:

https://github.com/Aakashnjit/Aakash_siricilla_midtermCS634

References:

Lecture Slides

https://en.wikipedia.org/wiki/Apriori_algorithm

Thank you!!