Apriori Algorithm Implementation

Using Python

ALAA SALIH

As422@njit.edu

Data Mining

CS634

APRIORI

-An algorithm behind
"You may also like"

Table of Content:

1. Introduction

In the period of unconvinced life and digital shopping and everything could be available online still will try to find a time to go supermarkets for shopping or quick pick up.

But did we think why certain items are placed together and are there any reason behind their placement such as milk near cookies, conditioners placed near shampoo and so on …
The shopkeeper knows the customers' sentiment so they look to optimizing the sales and makes a profit out of it.
Market Basket Analysis is a common technique used *specially* by large retailers to find hidden patterns on customer behaviors.

1.1 Apriori Algorithm Overview

is an algorithm for discovering frequent itemsets in transaction databases. This algorithm, introduced by R Agrawal and R Srikant in 1994 has great significance in data mining.
Apriori is one of the algorithms that use in E-commerce website as recommendation features websites to get recommended contents. Applied when we have transactions databases.In final we are looking to get frequent item sets i.e. items which are bought most frequently.

1.2 Association Rule Mining

Association Rule Mining is used when you want to find an association between different objects in a set, find frequent patterns in a transaction database, relational databases or any other information repository. The applications of Association Rule Mining are found in Marketing, Basket Data Analysis (or Market Basket Analysis) in retailing, clustering and classification.
The most common approach to find these patterns is Market Basket Analysis, which is a key technique used by large retailers like Amazon, Flipkart, etc to analyze customer buying habits by finding associations between the different items that customers place in their "shopping baskets". The discovery of these associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers.

Support

Support tells about the items that are frequently bought together. Support count is the frequency of occurrence of an itemset.

Confidence

If items A & B are bought together, Confidence tells us the number of times that A & B are bought together, given the number of times A is bought. For every purchase of A, Confidence tells us the number of times that B was also bought along with A.

Confidence c = frequency(A & B)/frequency(A)

1.3 Difference between Association and Recommendation

Association rules do not extract an individual's preference, rather find relationships between sets of elements of every distinct transaction. This is what makes them different than Collaborative filtering which is used in recommendation systems.

To understand it better take a look at below snapshot from amazon.com and you notice 2 headings "Frequently Bought Together" and the "Customers who bought this item also bought" on each product's info page.

"Frequently Bought Together" → Association

"Customers who bought this item also bought" → Recommendation

2. Applications used:

1- E-commerce website: online websites that sell products use this algorithm to suggest similar items to buyer to buy in future.

2- Social media:like Facebook, Instagram,printest etc.. by suggest friends to new account.

3- Location servies: Propose a place based on past visits that a person can visit in future.(Resturants,Mall...)

4- Autocomplete feature by search engine.

5- Find association in the student's database, patients' database, et

6- Others

2.1 Steps to implement:

Step 1: Scan the whole transaction database to fetch the support value S for each item.

Step 2: If the Support S is more than or equal to the minimum threshold, add the item to frequent itemset (L1), else go to step 1.

Step 3: Join Lk-1 and Lk-1, and generate the set of candidate k-itemsets

Step 4: For each k-itemset, get the support S and check the minimum support threshold

Step 5: Repeat the iteration in step 4, if support is not more than or equal to the minimum value

Step 6: If S is more than the required value, add to the frequent k-itemsets

Step 7: If there are no itemsets, stop the algorithm

Step 8: Till there are frequent itemsets, for each frequent itemset L, get all the non-empty subsets

Step 9: For each frequent subset of L, find the confidence C

Step 10: If the Confidence C is more than or equal to the minimum required Confidence, add it to the strong rules, else move to the next frequent subset.

3. Algorithm Architecture:

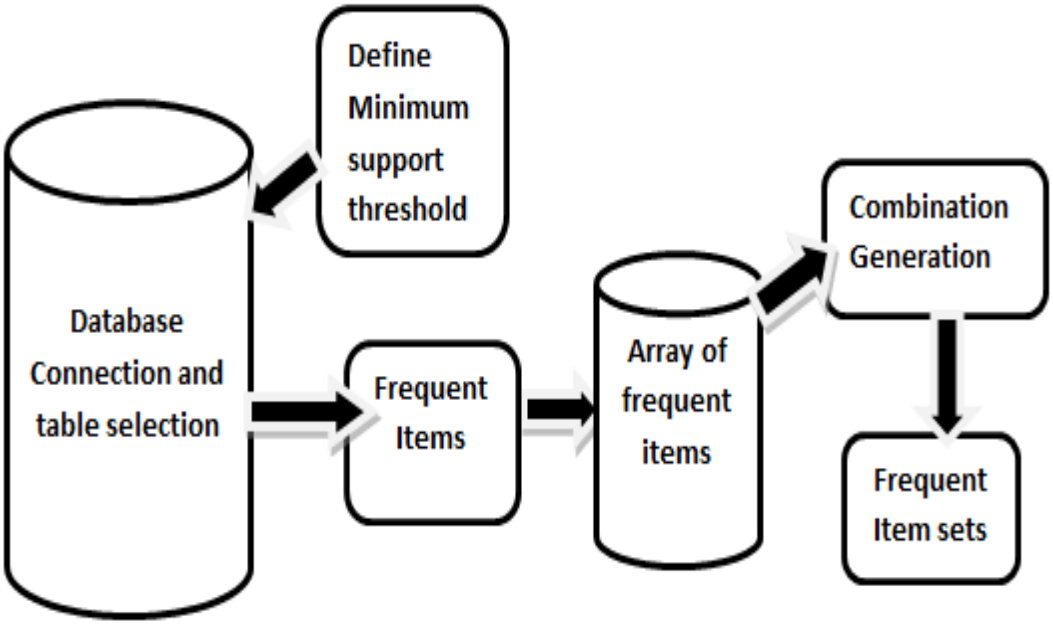

Figure 1 : Block Diagram implementation

4. Flow chart of Apriori:

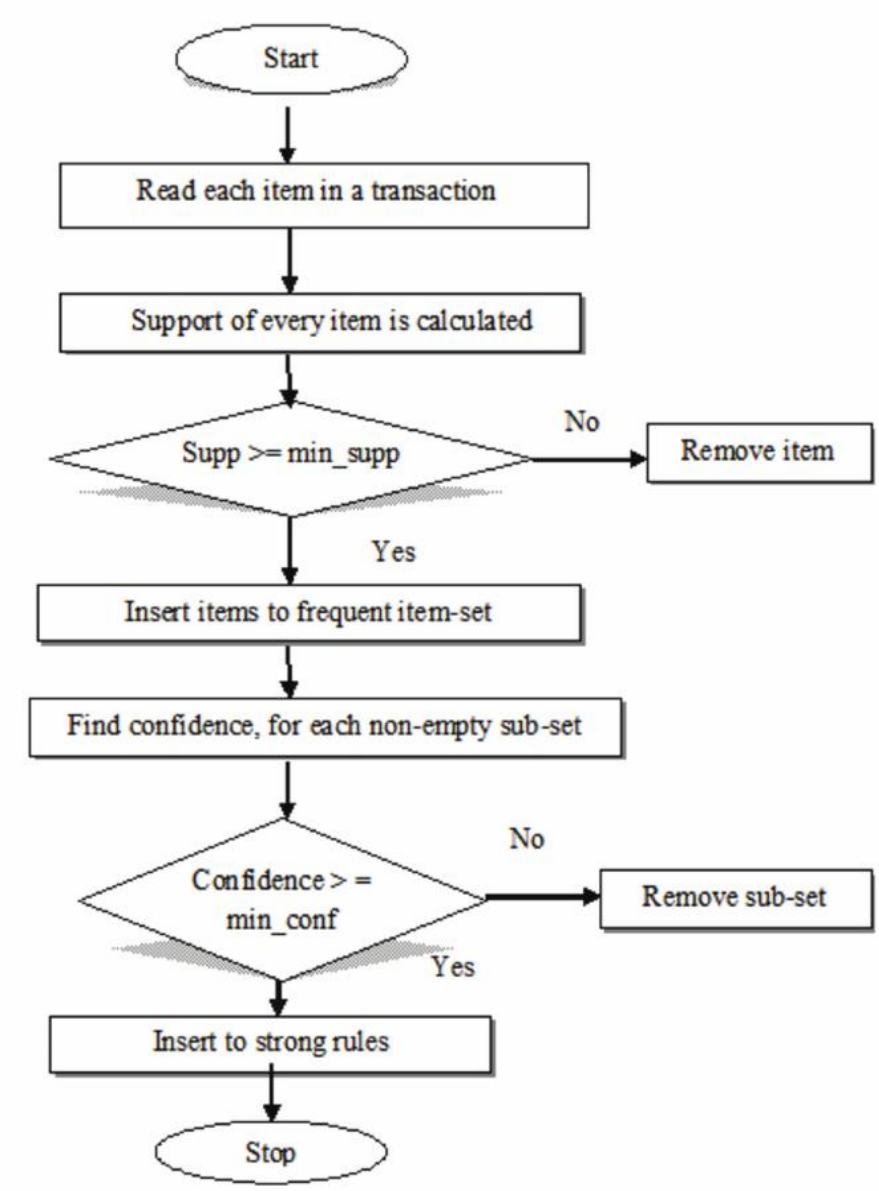

Figure 2: Flowchart

5. Project Description:

Write a Apriori algorithm, generate and print out all the association rules and the input transactions for each of the 4 transactional databases given (support and confidence should be user-specified parameters, so the output should show different support and confidence values with respect to different databases).

5.1 Minimum Requirements:

❖ **Hardware Specification:**

- Laptop processor Intel core i5
- 64 bit Windows Operating System

❖ **Software Specification:**

- Python 3.8, Anaconda Jupiter Notebook
- Libraries : Pandas, Numpy, OS and Itertools

Data Sets Source:

The data sets consist of data items names and 20 tuples Transactions. User can custom any data set as long as the file follows the same format of file.

1- Amazon

Item #	Item Name
1	A Beginner s Guide
2	Java: The Complete Reference

3	Java For Dummies
4	Android Programming: The Big Nerd Ranch
5	Head First Java 2nd Edition
6	Beginning Programming with Java
7	Java 8 Pocket Guide
8	C++ Programming in Easy Steps
9	Effective Java (2nd Edition)
10	HTML and CSS: Design and Build Websites

Table 1 Amazon Item Names

Transaction ID	Transaction
Trans1	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, Android Programming: The Big Nerd Ranch
Trans2	A Beginner s Guide, Java: The Complete Reference,Java For Dummies
Trans3	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans4	Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition ,Beginning Programming with Java,
Trans5	Android Programming: The Big Nerd Ranch, Beginning Programming with Java, Java 8 Pocket Guide
Trans6	A Beginner s Guide, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans7	A Beginner s Guide, Head First Java 2nd Edition ,Beginning Programming with Java
Trans8	Java: The Complete Reference,Java For Dummies, Android Programming: The Big Nerd Ranch,
Trans9	Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition ,Beginning Programming with Java,
Trans10	Beginning Programming with Java, Java 8 Pocket Guide, C++ Programming in Easy Steps
Trans11	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, Android Programming: The Big Nerd Ranch
Trans12	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, HTML and CSS: Design and Build Websites
Trans13	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, Java 8 Pocket Guide, HTML and CSS: Design and Build Websites
Trans14	Java For Dummies, Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans15	Java For Dummies, Android Programming: The Big Nerd Ranch
Trans16	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, Android Programming: The Big Nerd Ranch
Trans17	A Beginner s Guide, Java: The Complete Reference,Java For Dummies, Android Programming: The Big Nerd Ranch
Trans18	Head First Java 2nd Edition ,Beginning Programming with Java, Java 8 Pocket Guide
Trans19	Android Programming: The Big Nerd Ranch, Head First Java 2nd Edition
Trans20	A Beginner s Guide, Java: The Complete Reference,Java For Dummies

Table 2 Amazon Data Sets and Transactions

2- Best Buy

Item #	Item Name
1	Digital Camera

2	Lab Top
3	Desk Top
4	Printer
5	Flash Drive
6	Microsoft Office
7	Speakers
8	Lab Top Case
9	Anti-Virus
10	External Hard-Drive

Table 2 Best Buy Items Names

Transaction ID	Transaction
Trans1	Desk Top,Printer,Flash Drive,Microsoft Office,Speakers,Anti-Virus
Trans2	Lab Top,Flash Drive,Microsoft Office,Lab Top Case,Anti-Virus
Trans3	Lab Top,Printer,Flash Drive,Microsoft Office,Anti-Virus,Lab Top Case,External Hard-Drive
Trans4	Lab Top,Printer,Flash Drive,Anti-Virus,External Hard-Drive,Lab Top Case
Trans5	Lab Top,Flash Drive,Lab Top Case,Anti-Virus
Trans6	Lab Top,Printer,Flash Drive,Microsoft Office
Trans7	Desk Top,Printer,Flash Drive,Microsoft Office
Trans8	Lab Top,External Hard-Drive,Anti-Virus
Trans9	Desk Top,Printer,Flash Drive,Microsoft Office,Lab Top Case,Anti-Virus,Speakers,External Hard-Drive
Trans10	Digital Camera ,Lab Top,Desk Top,Printer,Flash Drive,Microsoft Office,Lab Top Case,Anti-Virus,External Hard-Drive,Speakers
Trans11	Lab Top,Desk Top,Lab Top Case,External Hard-Drive,Speakers,Anti-Virus
Trans12	Digital Camera ,Lab Top,Lab Top Case,External Hard-Drive,Anti-Virus,Speakers
Trans13	Digital Camera ,Speakers
Trans14	Digital Camera ,Desk Top,Printer,Flash Drive,Microsoft Office
Trans15	Printer,Flash Drive,Microsoft Office,Anti-Virus,Lab Top Case,Speakers,External Hard-Drive
Trans16	Digital Camera,Flash Drive,Microsoft Office,Anti-Virus,Lab Top Case,External Hard-Drive,Speakers
Trans17	Digital Camera ,Lab Top,Lab Top Case
Trans18	Digital Camera ,Lab Top Case,Speakers
Trans19	Digital Camera ,Lab Top,Printer,Flash Drive,Microsoft Office,Speakers,Lab Top Case,Anti-Virus
Trans20	Digital Camera ,Lab Top,Speakers,Anti-Virus,Lab Top Case

Table 2 Best Buy Data Sets Transactions

3- K-mart

Item #	Item Name
1	Quilts

2	Bedspreads
3	Decorative Pillows
4	Bed Skirts
5	Sheets
6	Shams
7	Bedding Collections
8	Kids Bedding
9	Embroidered Bedspread
10	Towels

Table 3 K-Mart data items

Transaction ID	Transaction
Trans1	Decorative Pillows,Quilts,Embroidered Bedspread
Trans2	Embroidered Bedspread,Shams,Kids Bedding,Bedding Collections,Bed Skirts,Bedspreads,Sheets
Trans3	Decorative Pillows,Quilts,Embroidered Bedspread,Shams,Kids Bedding,Bedding Collections
Trans4	Kids Bedding,Bedding Collections,Sheets,Bedspreads,Bed Skirts
Trans5	Decorative Pillows,Kids Bedding,Bedding Collections,Sheets,Bed Skirts,Bedspreads
Trans6	Bedding Collections,Bedspreads,Bed Skirts,Sheets,Shams,Kids Bedding
Trans7	Decorative Pillows,Quilts
Trans8	Decorative Pillows,Quilts,Embroidered Bedspread
Trans9	Bedspreads,Bed Skirts,Shams,Kids Bedding,Sheets
Trans10	Quilts,Embroidered Bedspread,Bedding Collections
Trans11	Bedding Collections,Bedspreads,Bed Skirts,Kids Bedding,Shams,Sheets
Trans12	Decorative Pillows,Quilts
Trans13	Embroidered Bedspread,Shams
Trans14	Sheets,Shams,Bed Skirts,Kids Bedding
Trans15	Decorative Pillows,Quilts
Trans16	Decorative Pillows,Kids Bedding,Bed Skirts,Shams
Trans17	Decorative Pillows,Shams,Bed Skirts
Trans18	Quilts,Sheets,Kids Bedding
Trans19	Shams,Bed Skirts,Kids Bedding,Sheets
Trans20	Decorative Pillows,Bedspreads,Shams,Sheets,Bed Skirts,Kids Bedding

Table 3 K-Mart Data Sets Transactions

4- Nike

Item #	Item Name

1	Running Shoe
2	Soccer Shoe
3	Socks
4	Swimming Shirt
5	Dry Fit V-Nick
6	Rash Guard
7	Sweatshirts
8	Hoodies
9	Tech Pants
10	Modern Pants

Table 4 Nike Data Items

Transaction ID	Transaction
Trans1	Running Shoe,Socks,Sweatshirts,Modern Pants
Trans2	Running Shoe,Socks,Sweatshirts
Trans3	Running Shoe,Socks,Sweatshirts,Modern Pants
Trans4	Running Shoe,Sweatshirts,Modern Pants
Trans5	Running Shoe,Socks,Sweatshirts,Modern Pants,Soccer Shoe
Trans6	Running Shoe,Socks,Sweatshirts
Trans7	Running Shoe,Socks,Sweatshirts,Modern Pants,Tech Pants,Rash Guard,Hoodies
Trans8	Swimming Shirt,Socks,Sweatshirts
Trans9	Swimming Shirt,Rash Guard,Dry Fit V-Nick,Hoodies,Tech Pants
Trans10	Swimming Shirt,Rash Guard,Dry
Trans11	Swimming Shirt,Rash Guard,Dry Fit V-Nick
Trans12	Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Soccer Shoe,Rash Guard,Hoodies,Tech Pants,Dry Fit V-Nick
Trans13	Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Soccer Shoe,Rash Guard,Tech Pants,Dry Fit V-Nick,Hoodies
Trans14	Running Shoe,Swimming Shirt,Rash Guard,Tech Pants,Hoodies,Dry Fit V-Nick
Trans15	Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Dry Fit V-Nick,Rash Guard,Tech Pants
Trans16	Swimming Shirt,Soccer Shoe,Hoodies,Dry Fit V-Nick,Tech Pants,Rash Guard
Trans17	Running Shoe,Socks
Trans18	Socks,Sweatshirts,Modern Pants,Soccer Shoe,Hoodies,Rash Guard,Tech Pants,Dry Fit V-Nick
Trans19	Running Shoe,Swimming Shirt,Rash Guard
Trans20	Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Soccer Shoe,Hoodies,Tech Pants,Rash Guard,Dry Fit V-Nick

Table 4 Nike Data Sets Transactions

5- Generic

Item #	Item Name

1	A
2	B
3	C
4	D
5	E
6	F

Table 5 Generic Items Names

Transaction ID	Transaction
Trans1	A,B,C
Trans2	A,B,C
Trans3	A,B,C,D
Trans4	A,B,C,D,E
Trans5	A,B,D,E
Trans6	A,D,E
Trans7	A,E
Trans8	A,E
Trans9	A,C,E
Trans10	A,C,E
Trans11	A,C,E

Table 5 Generic Data Sets and Transactions

6. Implementation:

- The program takes **data source**, **Minimum Support in percentage** and **Minimum Confidence in percentage** as input.
- **Data Source:** This is to select where the input is coming from. For this test, the data is coming from one of the five files that were created using generateDatabse.py.
- **Minimum Support:** A minimum support is applied to find all frequent itemsets in a database.
- **Minimum Confidence:** A minimum confidence is applied to these frequent itemsets in order to form rules.
- **Result:** The result will show the association rules in the given dataset with the given minimum support and minimum confidence if there are any. If there are no association rules in the the set with the given support and confidence conditions, try to plug in some different (*if you didn't get any results, try feeding some lower values*) values of them.

7. Testing & Execution Outputs

This sections describes several test for each data sets type with different support and confidence values. It tests each data set respectively.

Next will show an example for different min_support values and min_confidence values with the result of each.

7.1 User Manual :

Run the Application

The program will use start and prompt for options menu:

```
import pandas as pd
import numpy as np
from itertools import combinations

choice = input("Please, Select your Dataset for \n 1 Amazon.\n 2 bestbuy.\n 3 K_mart.\n 4 Nike.\n 5 Generic. \n ")
choice = int(choice)

if choice == 1:
    df =pd.read_csv('data/amazon.csv')
    print(df)
elif choice == 2:
    df = pd.read_csv('data/bestbuy.csv')
    print(df.head())
elif choice == 3:
    df = pd.read_csv('data/kmart.csv')
    print(df)
elif choice == 4:
    df = pd.read_csv('data/nike.csv')
    print(df)
elif choice == 5:
    df = pd.read_csv('data/generic.csv')
    print(df)
else:
    print("Wrong Choice")
```


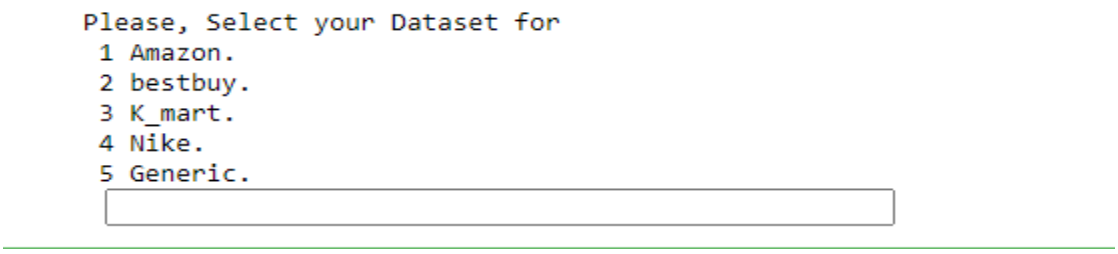

Figure 1 Running Program

1. Amazon

Running Amazon data set with support 20 and confidence 30

```
Please, Select your Dataset for
 1 Amazon.
 2 bestbuy.
 3 K_mart.
 4 Nike.
1
                                                    0
0    A Beginners Guide,Java: The Complete Reference...
1    A Beginners Guide,Java: The Complete Reference...
2    A Beginners Guide,Java: The Complete Reference...
3    Android Programming:The Big Nerd Ranch,Head Fi...
4    Android Programming:The Big Nerd Ranch,Beginni...
5    A Beginners Guide,Android Programming:The Big ...
6    A Beginners Guide,Head First Java 2nd Edition,...
7    Java: The Complete Reference,Java For Dummies,...
8    Java For Dummies,Android Programming:The Big N...
9    Beginning Programming with Java,Java 8 Pocket ...
10   A Beginners Guide,Java:The Complete Reference,...
11   A Beginners Guide,Java:The Complete Reference,...
12   A Beginners Guide,Java:The Complete Reference,...
13   Java For Dummies,Android Programming:The Big N...
14   Java For Dummies,Android Programming: The Big ...
15   A Beginners Guide,Java:The Complete Reference,...
16   A Beginners Guide,Java:The Complete Reference,...
17   Head First Java 2nd Edition,Beginning Programm...
18   Android Programming:The Big Nerd Ranch,Head Fi...
19   A Beginners Guide,Java: The Complete Reference...
```

Figure 2 Amazon Data Set Selection

```
min_sup = input("Please, input your Min. Support \n")
min_sup = float(min_sup)
min_con = input("Please, input your Min. confidence \n")
min_con = float(min_con)
```

```
Please, input your Min. Support
0.2
Please, input your Min. confidence
0.3
```

Figure 3 Amazon data set with support 20 and confidence 30

```
frequent patterns
 [(' Head First Java 2nd Edition ',), (' Beginning Programming with Java',), (' Android Programming: The Big Nerd Ranch',)]

Association rules
```

Figure 4 Amazon data set with support 20 and confidence 30 frequent pattern

```
frequent patterns
 [(' Java: The Complete Reference',), (' Head First Java 2nd Edition ',), (' Beginning Programming with Java',), (' Android Programming: The Big Nerd Ranch',), (' Java Fo
r Dummies',), (' Java 8 Pocket Guide',), (' Head First Java 2nd Edition',), (' Java: The Complete Reference', ' Head First Java 2nd Edition '), (' Java: The Complete Refe
rence', ' Android Programming: The Big Nerd Ranch'), (' Head First Java 2nd Edition ', ' Beginning Programming with Java'), (' Head First Java 2nd Edition ', ' Android Pr
ogramming: The Big Nerd Ranch'), (' Head First Java 2nd Edition ', ' Head First Java 2nd Edition'), (' Beginning Programming with Java', ' Head First Java 2nd Edition'),
(' Java: The Complete Reference', ' Head First Java 2nd Edition ', ' Android Programming: The Big Nerd Ranch'), (' Head First Java 2nd Edition ', ' Beginning Programming
with Java', ' Head First Java 2nd Edition')]

Association rules
(' Java: The Complete Reference',)  --->  [' Head First Java 2nd Edition ']    confidence =  1.0
(' Head First Java 2nd Edition ',)  --->  [' Java: The Complete Reference']    confidence =  0.5
(' Java: The Complete Reference',)  --->  [' Android Programming: The Big Nerd Ranch']    confidence =  1.0
(' Android Programming: The Big Nerd Ranch',)  --->  [' Java: The Complete Reference']    confidence =  0.5
(' Head First Java 2nd Edition ',)  --->  [' Beginning Programming with Java']    confidence =  0.5
(' Beginning Programming with Java',)  --->  [' Head First Java 2nd Edition ']    confidence =  0.5
(' Head First Java 2nd Edition ',)  --->  [' Android Programming: The Big Nerd Ranch']    confidence =  0.5
(' Android Programming: The Big Nerd Ranch',)  --->  [' Head First Java 2nd Edition ']    confidence =  0.5
(' Head First Java 2nd Edition ',)  --->  [' Head First Java 2nd Edition']    confidence =  0.5
(' Head First Java 2nd Edition',)  --->  [' Head First Java 2nd Edition ']    confidence =  1.0
(' Beginning Programming with Java',)  --->  [' Head First Java 2nd Edition']    confidence =  0.5
(' Head First Java 2nd Edition',)  --->  [' Beginning Programming with Java']    confidence =  1.0
(' Java: The Complete Reference', ' Head First Java 2nd Edition ')  --->  [' Android Programming: The Big Nerd Ranch']    confidence =  1.0
(' Java: The Complete Reference', ' Android Programming: The Big Nerd Ranch')  --->  [' Head First Java 2nd Edition ']    confidence =  1.0
(' Head First Java 2nd Edition ', ' Android Programming: The Big Nerd Ranch')  --->  [' Java: The Complete Reference']    confidence =  1.0
(' Head First Java 2nd Edition ', ' Beginning Programming with Java')  --->  [' Head First Java 2nd Edition']    confidence =  1.0
(' Head First Java 2nd Edition ', ' Head First Java 2nd Edition')  --->  [' Beginning Programming with Java']    confidence =  1.0
(' Beginning Programming with Java', ' Head First Java 2nd Edition')  --->  [' Head First Java 2nd Edition ']    confidence =  1.0
```

Figure 5 Amazon dataset with support 20 and confidence 30 associate rules

2. Best Buy

Running Best Buy data set with support 20 and confidence 40

```
Please, Select your Dataset for
 1 Amazon.
 2 bestbuy.
 3 K_mart.
 4 Nike.
2
                                                    0
0   Desk Top,Printer,Flash Drive,Microsoft Office,...
1   Lab Top,Flash Drive,Microsoft Office,Lab Top C...
2   Lab Top,Printer,Flash Drive,Microsoft Office,A...
3   Lab Top,Printer,Flash Drive,Anti-Virus,Externa...
4         Lab Top,Flash Drive,Lab Top Case,Anti-Virus
5       Lab Top,Printer,Flash Drive,Microsoft Office
6      Desk Top,Printer,Flash Drive,Microsoft Office
7             Lab Top,External Hard-Drive,Anti-Virus
8   Desk Top,Printer,Flash Drive,Microsoft Office,...
9   Digital Camera,Lab Top,Desk Top,Printer,Flash ...
10  Lab Top,Desk Top,Lab Top Case,External Hard-Dr...
11  Digital Camera,Lab Top,Lab Top Case,External H...
12                          Digital Camera,Speakers
13  Digital Camera,Desk Top,Printer,Flash Drive,Mi...
14  Printer,Flash Drive,Microsoft Office,Anti-Viru...
15  Digital Camera,Flash Drive,Microsoft Office,An...
16               Digital Camera,Lab Top,Lab Top Case
17              Digital Camera,Lab Top Case,Speakers
18  Digital Camera,Lab Top,Printer,Flash Drive,Mic...
19  Digital Camera,Lab Top,Speakers,Anti-Virus,Lab...
```

Figure 6 Bestbuy DataSet Selection

```
frequent patterns
 [(' Printer',), (' Flash Drive',), (' External Hard-Drive',), (' Lab Top',), (' Desk Top',), (' Speakers',), (' Lab Top Cas
e',), (' Printer', ' Flash Drive'), (' Printer', ' External Hard-Drive'), (' Printer', ' Desk Top'), (' Flash Drive', ' Externa
l Hard-Drive'), (' Flash Drive', ' Desk Top'), (' External Hard-Drive', ' Desk Top'), (' Lab Top', ' Desk Top'), (' Lab Top', '
Speakers'), (' Lab Top', ' Lab Top Case'), (' Desk Top', ' Speakers'), (' Desk Top', ' Lab Top Case'), (' Speakers', ' Lab Top
Case'), (' Printer', ' Flash Drive', ' External Hard-Drive'), (' Printer', ' Flash Drive', ' Desk Top'), (' Printer', ' Externa
l Hard-Drive', ' Desk Top'), (' Flash Drive', ' External Hard-Drive', ' Desk Top'), (' Lab Top', ' Desk Top', ' Speakers'), ('
Lab Top', ' Desk Top', ' Lab Top Case'), (' Lab Top', ' Speakers', ' Lab Top Case'), (' Desk Top', ' Speakers', ' Lab Top Cas
e'), (' Printer', ' Flash Drive', ' External Hard-Drive', ' Desk Top'), (' Lab Top', ' Desk Top', ' Speakers', ' Lab Top Cas
e')]
```


Figure 7 Best Buy frequent pattern with Sup=20 , Conf=40

```
Association rules
(' Printer',)  --->  [' Flash Drive']    confidence =  0.5
(' Printer',)  --->  [' External Hard-Drive']    confidence =  0.5
(' External Hard-Drive',)  --->  [' Printer']    confidence =  1.0
(' Printer',)  --->  [' Desk Top']    confidence =  0.5
(' Desk Top',)  --->  [' Printer']    confidence =  0.5
(' External Hard-Drive',)  --->  [' Flash Drive']    confidence =  1.0
(' Desk Top',)  --->  [' Flash Drive']    confidence =  0.5
(' External Hard-Drive',)  --->  [' Desk Top']    confidence =  1.0
(' Desk Top',)  --->  [' External Hard-Drive']    confidence =  0.5
(' Lab Top',)  --->  [' Desk Top']    confidence =  1.0
(' Desk Top',)  --->  [' Lab Top']    confidence =  0.5
(' Lab Top',)  --->  [' Speakers']    confidence =  1.0
(' Speakers',)  --->  [' Lab Top']    confidence =  1.0
(' Lab Top',)  --->  [' Lab Top Case']    confidence =  1.0
(' Lab Top Case',)  --->  [' Lab Top']    confidence =  1.0
(' Desk Top',)  --->  [' Speakers']    confidence =  0.5
(' Speakers',)  --->  [' Desk Top']    confidence =  1.0
(' Desk Top',)  --->  [' Lab Top Case']    confidence =  0.5
(' Lab Top Case',)  --->  [' Desk Top']    confidence =  1.0
(' Speakers',)  --->  [' Lab Top Case']    confidence =  1.0
(' Lab Top Case',)  --->  [' Speakers']    confidence =  1.0
(' Printer', ' Flash Drive')  --->  [' External Hard-Drive']    confidence =  1.0
(' Printer', ' External Hard-Drive')  --->  [' Flash Drive']    confidence =  1.0
(' Flash Drive', ' External Hard-Drive')  --->  [' Printer']    confidence =  1.0
(' Printer', ' Flash Drive')  --->  [' Desk Top']    confidence =  1.0
(' Printer', ' Desk Top')  --->  [' Flash Drive']    confidence =  1.0
(' Flash Drive', ' Desk Top')  --->  [' Printer']    confidence =  1.0
(' Printer', ' External Hard-Drive')  --->  [' Desk Top']    confidence =  1.0
(' Printer', ' Desk Top')  --->  [' External Hard-Drive']    confidence =  1.0
(' External Hard-Drive', ' Desk Top')  --->  [' Printer']    confidence =  1.0
(' Flash Drive', ' External Hard-Drive')  --->  [' Desk Top']    confidence =  1.0
(' Flash Drive', ' Desk Top')  --->  [' External Hard-Drive']    confidence =  1.0
(' External Hard-Drive', ' Desk Top')  --->  [' Flash Drive']    confidence =  1.0
(' Lab Top', ' Desk Top')  --->  [' Speakers']    confidence =  1.0
(' Lab Top', ' Speakers')  --->  [' Desk Top']    confidence =  1.0
(' Desk Top', ' Speakers')  --->  [' Lab Top']    confidence =  1.0
(' Lab Top', ' Desk Top')  --->  [' Lab Top Case']    confidence =  1.0
(' Lab Top', ' Lab Top Case')  --->  [' Desk Top']    confidence =  1.0
(' Desk Top', ' Lab Top Case')  --->  [' Lab Top']    confidence =  1.0
(' Lab Top', ' Speakers')  --->  [' Lab Top Case']    confidence =  1.0
(' Lab Top', ' Lab Top Case')  --->  [' Speakers']    confidence =  1.0
(' Speakers', ' Lab Top Case')  --->  [' Lab Top']    confidence =  1.0
(' Desk Top', ' Speakers')  --->  [' Lab Top Case']    confidence =  1.0
(' Desk Top', ' Lab Top Case')  --->  [' Speakers']    confidence =  1.0
(' Desk Top', ' Speakers', ' Lab Top Case')  --->  [' Lab Top']    confidence =  1.0
(' Printer', ' Flash Drive', ' External Hard-Drive')  --->  [' Desk Top']    confidence =  1.0
(' Printer', ' Flash Drive', ' Desk Top')  --->  [' External Hard-Drive']    confidence =  1.0
(' Printer', ' External Hard-Drive', ' Desk Top')  --->  [' Flash Drive']    confidence =  1.0
(' Flash Drive', ' External Hard-Drive', ' Desk Top')  --->  [' Printer']    confidence =  1.0
(' Lab Top', ' Desk Top', ' Speakers')  --->  [' Lab Top Case']    confidence =  1.0
(' Lab Top', ' Desk Top', ' Lab Top Case')  --->  [' Speakers']    confidence =  1.0
(' Lab Top', ' Speakers', ' Lab Top Case')  --->  [' Desk Top']    confidence =  1.0
(' Desk Top', ' Speakers', ' Lab Top Case')  --->  [' Lab Top']    confidence =  1.0
```

Figure 8 Best Buy Data with Sup=20, Conf=40 association Rules

```
frequent patterns
 [(' Flash Drive',)]

Association rules
```

Figure 9 Best buy frequent pattern with Sup= 50, Conf=50

3. K-Mart

Running K-Mart Data Set example with Support 20 and Confidence 30

```
Please, Select your Dataset for
 1 Amazon.
 2 bestbuy.
 3 K_mart.
 4 Nike.
3
                                                    0
0     Decorative Pillows,Quilts,Embroidered Bedspread
1   Embroidered Bedspread,Shams,Kids Bedding,Beddi...
2   Decorative Pillows,Quilts,Embroidered Bedsprea...
3   Kids Bedding,Bedding Collections,Sheets,Bedspr...
4   Decorative Pillows,Kids Bedding,Bedding Collec...
5   Bedding Collections,Bedspreads,Bed Skirts,Shee...
6                          Decorative Pillows,Quilts
7     Decorative Pillows,Quilts,Embroidered Bedspread
8     Bedspreads,Bed Skirts,Shams,Kids Bedding,Sheets
9    Quilts,Embroidered Bedspread,Bedding Collections
10  Bedding Collections,Bedspreads,Bed Skirts,Kids...
11                         Decorative Pillows,Quilts
12                       Embroidered Bedspread,Shams
13              Sheets,Shams,Bed Skirts,Kids Bedding
14                         Decorative Pillows,Quilts
15   Decorative Pillows,Kids Bedding,Bed Skirts,Shams
16                Decorative Pillows,Shams,Bed Skirts
17                         Quilts,Sheets,Kids Bedding
18              Shams,Bed Skirts,Kids Bedding,Sheets
19  Decorative Pillows,Bedspreads,Shams,Sheets,Bed...
```

Figure 10 K_mart Data Set Selection

```
frequent patterns
 [('Decorative Pillows, Quilts, Embroidered Bedspread',), ('Decorative Pillows, Quilts',)]

Association rules
```

Figure 11 K-Mart Data Set frequent pattern with Sup=20, Confi=30

```
frequent patterns
 []

Association rules
```

Figure 12 K-Mart with Sup=70 and Conf=80,No rules found

```
frequent patterns
 [(' Quilts',), (' Shams',), (' Bedding Collections',), (' Kids Bedding',), (' Bedspreads',), (' Bed Skirts',), (' Embroidered
Bedspread',), (' Sheets',), (' Quilts', ' Shams'), (' Quilts', ' Kids Bedding'), (' Quilts', ' Bedspreads'), (' Shams', ' Kids
Bedding'), (' Shams', ' Bedspreads'), (' Kids Bedding', ' Bedspreads'), (' Embroidered Bedspread', ' Sheets'), (' Quilts', ' Sh
ams', ' Kids Bedding'), (' Quilts', ' Shams', ' Bedspreads'), (' Quilts', ' Kids Bedding', ' Bedspreads'), (' Shams', ' Kids Be
dding', ' Bedspreads'), (' Quilts', ' Shams', ' Kids Bedding', ' Bedspreads')]
```

Figure 13 K-Mart Data Set frequent pattern with Sup=0.1, Confi=0.1

```
Association rules
(' Quilts',)  --->  [' Shams']    confidence =  1.0
(' Shams',)  --->  [' Quilts']    confidence =  0.3333333333333333
(' Quilts',)  --->  [' Kids Bedding']    confidence =  1.0
(' Kids Bedding',)  --->  [' Quilts']    confidence =  1.0
(' Quilts',)  --->  [' Bedspreads']    confidence =  1.0
(' Bedspreads',)  --->  [' Quilts']    confidence =  0.5
(' Shams',)  --->  [' Kids Bedding']    confidence =  0.3333333333333333
(' Kids Bedding',)  --->  [' Shams']    confidence =  1.0
(' Shams',)  --->  [' Bedspreads']    confidence =  0.3333333333333333
(' Bedspreads',)  --->  [' Shams']    confidence =  0.5
(' Kids Bedding',)  --->  [' Bedspreads']    confidence =  1.0
(' Bedspreads',)  --->  [' Kids Bedding']    confidence =  0.5
(' Embroidered Bedspread',)  --->  [' Sheets']    confidence =  1.0
(' Sheets',)  --->  [' Embroidered Bedspread']    confidence =  1.0
(' Quilts', ' Shams')  --->  [' Kids Bedding']    confidence =  1.0
(' Quilts', ' Kids Bedding')  --->  [' Shams']    confidence =  1.0
(' Shams', ' Kids Bedding')  --->  [' Quilts']    confidence =  1.0
(' Quilts', ' Shams')  --->  [' Bedspreads']    confidence =  1.0
(' Quilts', ' Bedspreads')  --->  [' Shams']    confidence =  1.0
(' Shams', ' Bedspreads')  --->  [' Quilts']    confidence =  1.0
(' Quilts', ' Kids Bedding')  --->  [' Bedspreads']    confidence =  1.0
(' Quilts', ' Bedspreads')  --->  [' Kids Bedding']    confidence =  1.0
(' Kids Bedding', ' Bedspreads')  --->  [' Quilts']    confidence =  1.0
(' Shams', ' Kids Bedding')  --->  [' Bedspreads']    confidence =  1.0
(' Shams', ' Bedspreads')  --->  [' Kids Bedding']    confidence =  1.0
(' Kids Bedding', ' Bedspreads')  --->  [' Shams']    confidence =  1.0
(' Quilts', ' Shams', ' Kids Bedding')  --->  [' Bedspreads']    confidence =  1.0
(' Quilts', ' Shams', ' Bedspreads')  --->  [' Kids Bedding']    confidence =  1.0
(' Quilts', ' Kids Bedding', ' Bedspreads')  --->  [' Shams']    confidence =  1.0
(' Shams', ' Kids Bedding', ' Bedspreads')  --->  [' Quilts']    confidence =  1.0
```

Figure 14 K-Mart Data Set with Sup=0.1, Confi=0.1, association Rules

4. Nike

Running Nike Data Set example with Support 70 and Confidence 80

```
Please, Select your Dataset for
 1 Amazon.
 2 bestbuy.
 3 K_mart.
 4 Nike.
4
                                                    0
0         Running Shoe,Socks,Sweatshirts,Modern Pants
1                      Running Shoe,Socks,Sweatshirts
2         Running Shoe,Socks,Sweatshirts,Modern Pants
3              Running Shoe,Sweatshirts, Modern Pants
4   Running Shoe,Socks,Sweatshirts,Modern Pants,So...
5                      Running Shoe,Socks,Sweatshirts
6   Running Shoe,Socks,Sweatshirts,Modern Pants,Te...
7                    Swimming Shirt,Socks,Sweatshirts
8   Swimming Shirt,Rash Guard,Dry Fit V-Nick,Hoodi...
9                       Swimming Shirt,Rash Guard,Dry
10           Swimming Shirt,Rash Guard,Dry Fit V-Nick
11  Running Shoe,Swimming Shirt,Socks,Sweatshirts,...
12  Running Shoe,Swimming Shirt,Socks,Sweatshirts,...
13  Running Shoe,Swimming Shirt,Rash Guard, Tech P...
14  Running Shoe,Swimming Shirt,Socks,Sweatshirts,...
15  Swimming Shirt,Soccer Shoe,Hoodies,Dry Fit V-N...
16                                 Running Shoe,Socks
17  Socks,Sweatshirts,Modern Pants,Soccer Shoe,Hoo...
18             Running Shoe,Swimming Shirt,Rash Guard
19  Running Shoe,Swimming Shirt,Socks,Sweatshirts,...
```

Figure 15 Nike Data Set user selection

```
frequent patterns
 []

Association rules
```

Figure 16 Nike Data Set Sup=70, Conf= 80, No rules

On more Running Nike Data Set example with Support 10 and Confidence 10

```
frequent patterns
 [(' Socks',), (' Sweatshirts',), (' Rash Guard',), (' Swimming Shirt',), (' Soccer Shoe',), (' Socks', ' Sweatshirts'), (' Soc
ks', ' Rash Guard'), (' Socks', ' Swimming Shirt'), (' Socks', ' Soccer Shoe'), (' Sweatshirts', ' Swimming Shirt'), (' Rash Gu
ard', ' Soccer Shoe'), (' Socks', ' Sweatshirts', ' Swimming Shirt'), (' Socks', ' Rash Guard', ' Soccer Shoe')]

Association rules
(' Socks',)  --->  [' Sweatshirts']    confidence =  0.5
(' Sweatshirts',)  --->  [' Socks']    confidence =  0.5
(' Socks',)  --->  [' Rash Guard']    confidence =  0.5
(' Rash Guard',)  --->  [' Socks']    confidence =  1.0
(' Socks',)  --->  [' Swimming Shirt']    confidence =  0.5
(' Swimming Shirt',)  --->  [' Socks']    confidence =  1.0
(' Socks',)  --->  [' Soccer Shoe']    confidence =  0.5
(' Soccer Shoe',)  --->  [' Socks']    confidence =  1.0
(' Sweatshirts',)  --->  [' Swimming Shirt']    confidence =  0.5
(' Swimming Shirt',)  --->  [' Sweatshirts']    confidence =  1.0
(' Rash Guard',)  --->  [' Soccer Shoe']    confidence =  1.0
(' Soccer Shoe',)  --->  [' Rash Guard']    confidence =  1.0
(' Socks', ' Sweatshirts')  --->  [' Swimming Shirt']    confidence =  1.0
(' Socks', ' Swimming Shirt')  --->  [' Sweatshirts']    confidence =  1.0
(' Sweatshirts', ' Swimming Shirt')  --->  [' Socks']    confidence =  1.0
(' Socks', ' Rash Guard')  --->  [' Soccer Shoe']    confidence =  1.0
(' Socks', ' Soccer Shoe')  --->  [' Rash Guard']    confidence =  1.0
(' Rash Guard', ' Soccer Shoe')  --->  [' Socks']    confidence =  1.0
```


Figure 17 Nike dataset Sup=0.1, Conf=0.1 association rules

5. Generic

Running generic Data Set example with Support 10 and Confidence 10

```
Please, Select your Dataset for
 1 Amazon.
 2 bestbuy.
 3 K_mart.
 4 Nike.
 5 Generic.
 5
              0
0       A,B,C,D
1           B,C
2         E,F,I
3         A,G,J
4         D,H,I
5           I,J
6     B,D,F,H,J
7         C,D,E
8         A,E,J
9       F,G,H,I
10    B,C,D,E,F
11    D,G,H,I,J
12          B,F
13          C,D
14      F,G,H,I
15        H,I,J
16        A,B,C
17    A,B,C,D,E
18      G,H,I,J
19    E,F,G,H,I
```

Figure 18 Generic Data Set selection

```
frequent patterns
 [(nan,), ('B',), ('C',), ('F',), ('G',), ('H',), ('J',), ('D',), ('E',), ('I',), ('B', 'G'), ('B', 'E'), ('C', 'F'), ('C',
'D'), ('F', 'D'), ('G', 'H'), ('G', 'E'), ('B', 'G', 'E'), ('C', 'F', 'D')]

Association rules
('B',)  --->  ['G']    confidence =  1.0
('G',)  --->  ['B']    confidence =  0.3333333333333333
('B',)  --->  ['E']    confidence =  1.0
('E',)  --->  ['B']    confidence =  1.0
('C',)  --->  ['F']    confidence =  1.0
('F',)  --->  ['C']    confidence =  0.5
('C',)  --->  ['D']    confidence =  1.0
('D',)  --->  ['C']    confidence =  0.5
('F',)  --->  ['D']    confidence =  0.5
('D',)  --->  ['F']    confidence =  0.5
('G',)  --->  ['H']    confidence =  0.3333333333333333
('H',)  --->  ['G']    confidence =  0.5
('G',)  --->  ['E']    confidence =  0.3333333333333333
('E',)  --->  ['G']    confidence =  1.0
('B', 'G')  --->  ['E']    confidence =  1.0
('B', 'E')  --->  ['G']    confidence =  1.0
('G', 'E')  --->  ['B']    confidence =  1.0
('C', 'F')  --->  ['D']    confidence =  1.0
('C', 'D')  --->  ['F']    confidence =  1.0
('F', 'D')  --->  ['C']    confidence =  1.0
```


Figure 19 Generic Data frequent pattern with Sup= 10 and Conf=10

One more Running generic Data Set example with Support 50 and Confidence 50

```
frequent patterns
 []

Association rules
```

Figure 20 Generic Dataset with Sup= 50 and Conf=50, No rules found

Conclusion

Association rule mining algorithms such as Apriori algorithim is good way to finding simple associations between data items (Transactions). We could conclude that no rules found happens to have a very low support, so that we do not have enough information on the relationship between its items.

For a beginner, it provides an easy way to understand the association rules and quickly apply for market basket analysis. Although there are limitations such as

- Time consuming: when we have huge transactional database; program take much time; faced this problem.
- Inefficiency when memory capacity is limited with large number of transactions.

Reference:

1. URL: https://www.kdnuggets.com/2019/12/market-basket-analysis.html
2. https://www.learnpython.org/
3. https://www.w3schools.com/python/
4. Solved error refer to https://stackoverflow.com/questions/
5. Lectures Notes

Appendix

Source code in python:

```
import pandas as pd
import numpy as np
from itertools import combinations
choice = input("Please, Select your Dataset for \n 1 Amazon.\n 2 bestbuy.\n 3 K_mart.\n 4 Nike.\n 5
Generic. \n ")
choice = int(choice)

if choice == 1:
    df =pd.read_csv('data/amazon.csv')
    print(df)
elif choice == 2:
    df = pd.read_csv('data/bestbuy.csv')
    print(df.head())
elif choice == 3:
    df = pd.read_csv('data/kmart.csv')
    print(df)
elif choice == 4:
    df = pd.read_csv('data/nike.csv')
    print(df)
elif choice == 5:
    df = pd.read_csv('data/generic.csv')
    print(df)
else:
    print("Wrong Choice")
min_sup = input("Please, input your Min. Support \n")
min_sup = float(min_sup)
min_con = input("Please, input your Min. confidence \n")
min_con = float(min_con)
```

```
# Convert transctions to List
names = list(df.columns)
tid = df[names[0]]
items = df[names[1]]
uni_items = df[names[1]].unique()
uni_tid = df[names[0]].unique()

def build_transactions(uni_tid, tid, items):
    transactions = []
    for i in uni_tid:
        temp_list = []
        for j in range(0, len(tid)):
            if tid[j] == i:
                temp_list.append(items[j])
        transactions.append(temp_list)
    return(transactions)

transactions = build_transactions(uni_tid, tid, items)
num_trans = len(transactions)
def check_pattern(list1, list2):
    x = 0
    if(all(x in list2 for x in list1)):
        x = 1
    return x

def update_fre_items (a, b):
    f = []

    for i in a:
        for j in i:
            f.append(j)
```

```
    temp = []
    for i in b:
        if i  in f:
            temp.append(i)

    return temp

pat_size = 1
fre_pat = []
#Number of pattrens
fre_pat_count = []
temp_fre_pat = [1]
fre_items = list(uni_items)
while (temp_fre_pat):

    # generate acceptable patterns
    pats = combinations(fre_items, pat_size)
    temp_fre_pat = [] # frequent patterns
    for f in list(pats):
        count = 0
        for t in transactions:
            count = count + check_pattern(f, t)
        if count >= min_sup * num_trans:
            temp_fre_pat.append(f)
            fre_pat_count.append(count)

    fre_pat = fre_pat + temp_fre_pat
    pat_size += 1
    # update frequent items list for creating new patterns
    fre_items = update_fre_items(temp_fre_pat, fre_items)
```

```
print('frequent patterns   \n',fre_pat)

print('\nAssociation rules')

for i in fre_pat:

   if len(i) > 1:

      sub_groups = list(combinations(i, len(i) - 1))
      #print(sub_groups)
      for j in sub_groups:
         temp = []
         for k in j:
            temp.append(k)
         z = list(set(i).difference(set(temp)))
         confidence = fre_pat_count[fre_pat.index(i)] / fre_pat_count[fre_pat.index(j)]
         if confidence > min_con:
            print(j,' ---> ', z ,'   confidence = ',confidence)
```

//
//