COURSE: CS634

SUBJECT: MIDTERM PROJECT

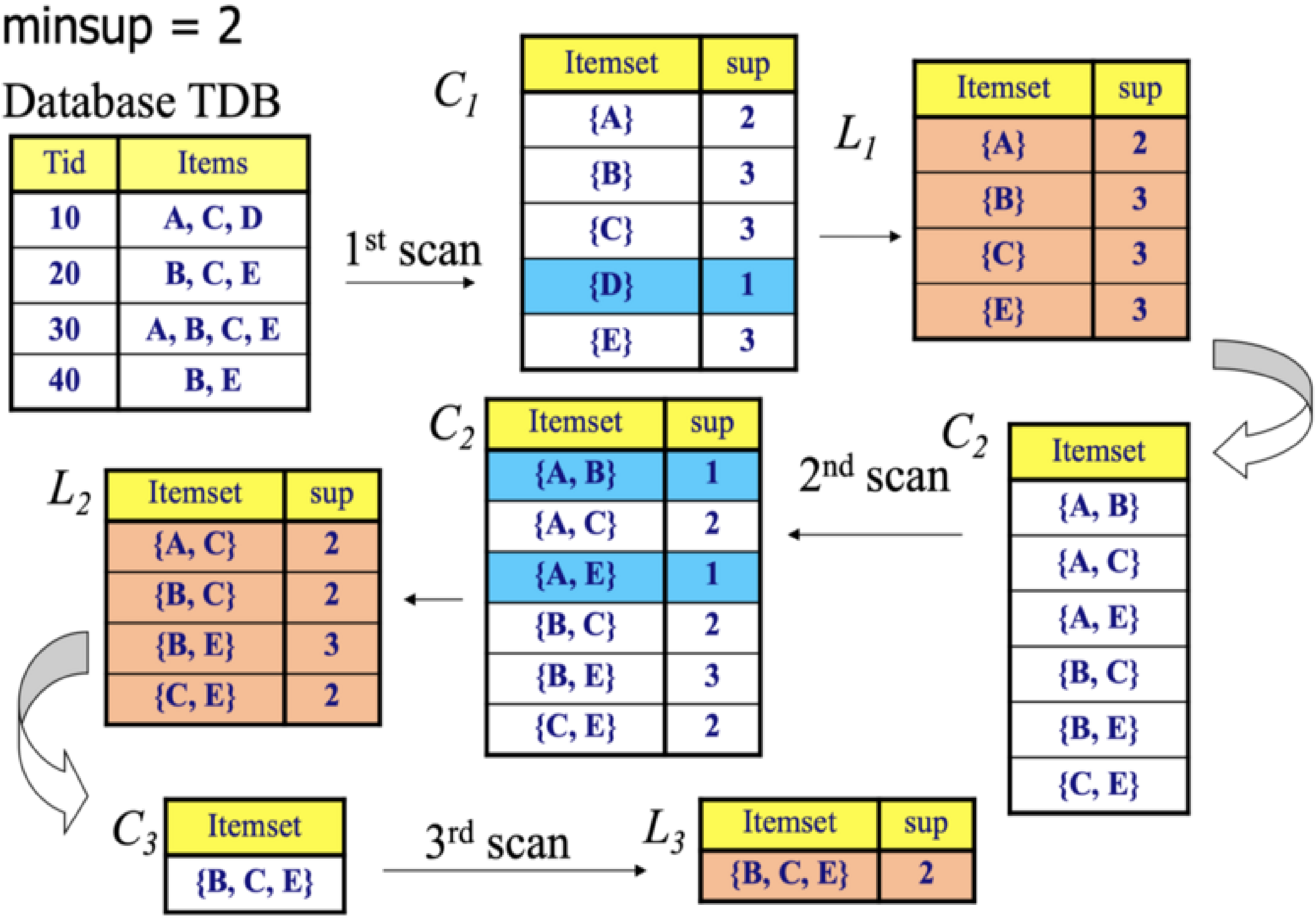
IMPLEMENTATION TOPIC: APRIORI ALGORITHM IMPLEMENTATION

NAME: NIKITA NEMANE

EMAIL:nn43@njit.edu

Nikita Nemane: nn43@njit.edu

Apriori Algorithm Implementation



minsup = 2

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

Contents

Introduction

Apriori Algorithm Overview

Apriori is an algorithm for frequent item set mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the dataset and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the dataset. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database.

It is designed to operate on datasets containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.

Implementation Overview

The implementation is built from scratch. It uses Python as the main programming language. The implementation uses CSV flat files to store the items sets and transactions.

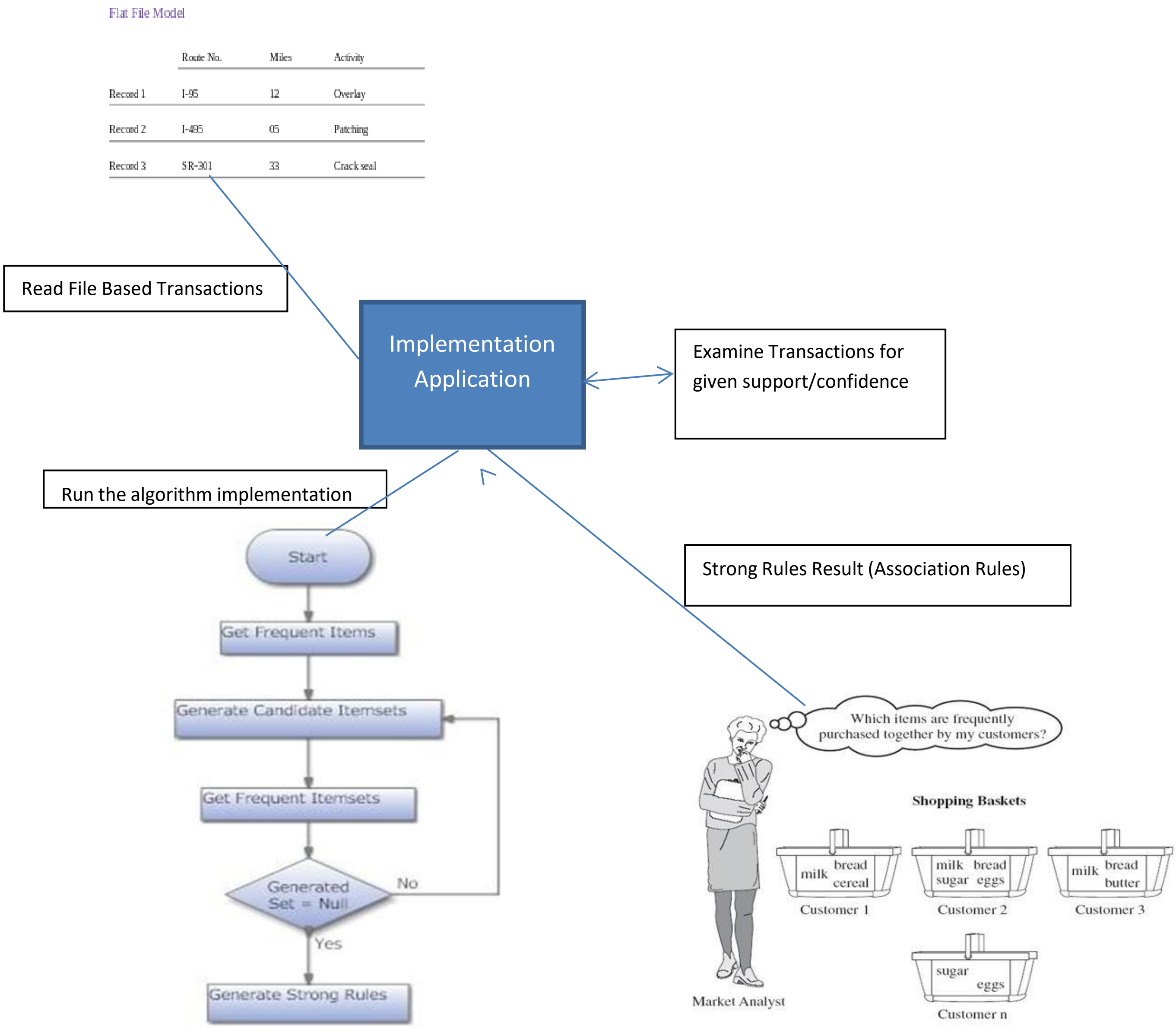
When you run the program, it will prompt you with number of choices to choose from based on the dataset type, support and confidence.

Based on the choice, the program will read the flat file dataset, parse the items names and transactions, and then load them to parse the transaction for support and confidence to find the association rules using Apriori algorithm.

GitHub Link to Source Code

https://github.com/nemanenikita/nemane_nikita_midtermproj.git

Implementation Architecture



Flat File Model

	Route No.	Miles	Activity
Record 1	I-95	12	Overlay
Record 2	I-495	05	Patching
Record 3	SR-301	33	Crack seal

Read File Based Transactions
Implementation Application
Examine Transactions for given support/confidence
Run the algorithm implementation
Strong Rules Result (Association Rules)
Start
Get Frequent Items
Generate Candidate Itemsets
Get Frequent Itemsets
Generated Set = Null
No
Yes
Generate Strong Rules
Which items are frequently purchased together by my customers?
Shopping Baskets
milk bread cereal
Customer 1
milk bread sugar eggs
Customer 2
milk bread butter
Customer 3
sugar eggs
Customer n
Market Analyst

Assumptions

The assumptions are as following:

- Apriori algorithm assumes that any subset of a frequent itemset must be frequent.
  Example: Say, a transaction containing {a, b, c} also contains {a, b}. So, according to the principle of Apriori, if {a, b, c} is frequent, then {a, b} must also be frequent.
- Datasets are assumed to be CSV flat files with comma as a delimiter.

Requirement

Software

- Programing Languages: Python version: 3.8.8
- IDE: Jupyter NoteBook version: 6.4.4
- Additional libraries
  - pandas
    - An open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.
    - Using this to read a comma-separated values (csv) file into DataFrame.
  - Numpy
    - A library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of various mathematical functions to operate on these arrays.

Hardware

- Operating System: Windows, Unix, Linux

Implementation List of Functions

The implementation is built from scratch. It uses Python as the main programming language. The implementation uses CSV files to store the items sets and transactions.

The program consists of the following python functions.

1) def load_data_set ()

This function is written to take user input to select the database. And returns the path of the selected database by the user.

2) def single_element()

This function uses data path given by user then traverse the Transactions in that CSV file. Split the data from CSV file and uniquely finds the elements and returns the order of the elements.

3) def load_data()

This function takes data path and order of elements as parameters. It then uniquely identifies the elements and append those elements in a Transaction list and returns the Transactions in the list format.

4) def get_frequent_items()

This function takes item sets, Transactions, given support and previously discarded items as parameters. This function will 1st check the length of discarded itemsets if it is more than 0 then it will check if the subset of previously discarded items is present in given itemset or not this is getting checked using pythons inbuilt set and subset functions. If it is present then it will discard that itemset also. Otherwise, it will call the function occurrence_count to take count of that itemset.

If the support of that itemset is greater than the given support then it is added in the main list otherwise that element is added in discarded list. This function will return List of frequent items, list of its support count and list of discarded items.

5) def occurrence_count()

This function goes over the transactions and count the number of occurrences and using python sets operations checks if the itemset is subset of the set or not and then increments the count and returns the count.

6) def combination_sets()

This function will call combination_two_sets() functions to check if it returns empty list or not and if not then it appends that list to join the two itemsets.

7) def combination_two_sets()

This function joins two sets, it first sorts itemsets according to their order. Checks if they are joinable, means all the items within the sorted itemsets are same except the last one and in the last one the one belonging to second itemset must be greater than the last one in the 1st itemset.

If it satisfies this then it returns joined itemsets otherwise it will return empty list.

8) def powerset()

This function uses python’s inbuilt functions of combination to generate the association rules.

9) def write_rules()

This function is written just to write rules in a user readable format.

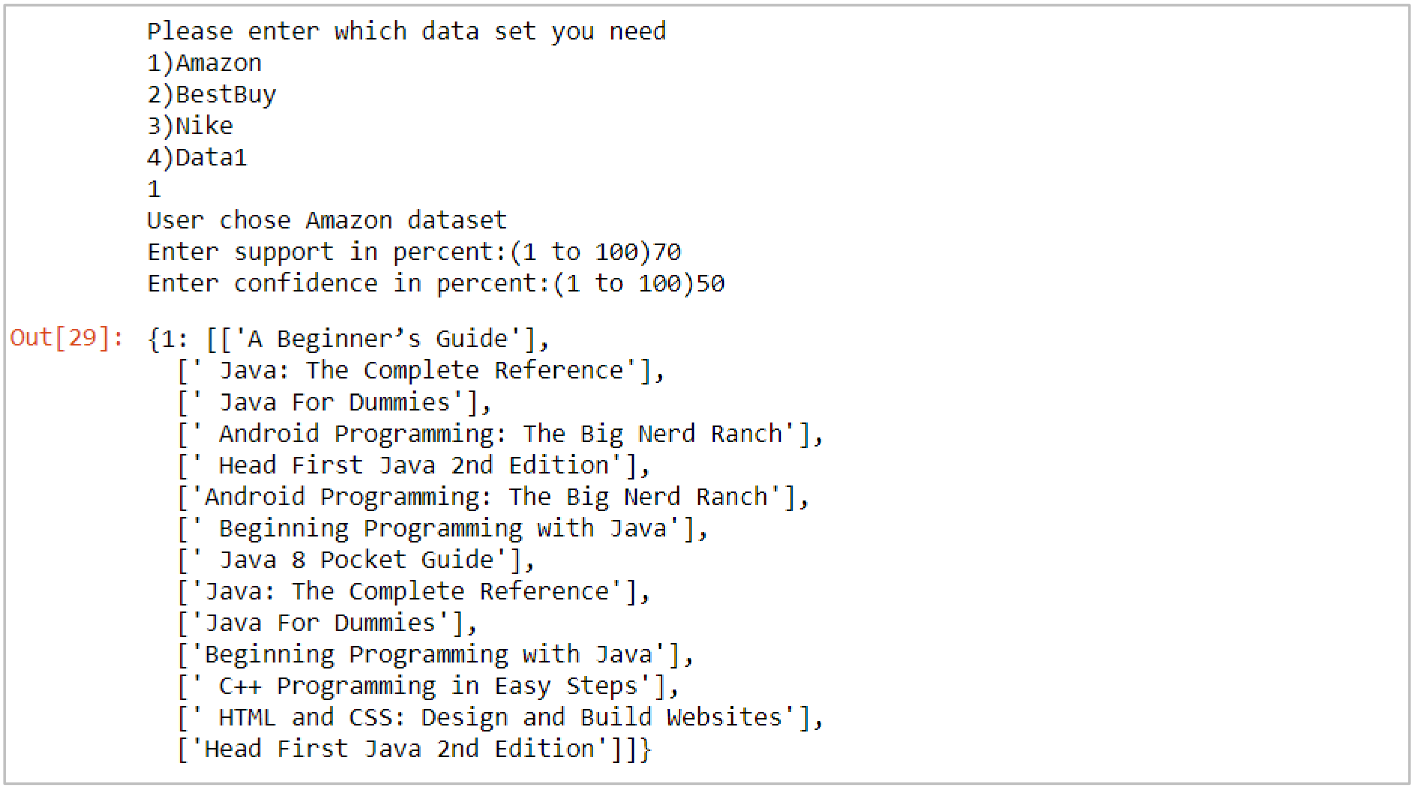
10) def print_table()

This function is written just to write table in a user readable format.

How to Run the Application

- Open Jupyter Notebook(anaconda3).exe which will open up a command prompt window with Python NoteBook
- It will pop up local code directory, navigate to project directory to open the project namend: nemane_nikita_midtermproj. ipynb file
- Click on 'Run' button to execute each cell of the Python program
- Enter dataset choice between 1 to 4, Support and Confidence
- Click on Run button until you find frequent item sets and finally association rules

The program will use start and prompt for options menu:



```
          Please enter which data set you need
          1)Amazon
          2)BestBuy
          3)Nike
          4)Data1
          1
          User chose Amazon dataset
          Enter support in percent:(1 to 100)70
          Enter confidence in percent:(1 to 100)50

Out[29]:  {1: [['A Beginner’s Guide'],
            [' Java: The Complete Reference'],
            [' Java For Dummies'],
            [' Android Programming: The Big Nerd Ranch'],
            [' Head First Java 2nd Edition'],
            ['Android Programming: The Big Nerd Ranch'],
            [' Beginning Programming with Java'],
            [' Java 8 Pocket Guide'],
            ['Java: The Complete Reference'],
            ['Java For Dummies'],
            ['Beginning Programming with Java'],
            [' C++ Programming in Easy Steps'],
            [' HTML and CSS: Design and Build Websites'],
            ['Head First Java 2nd Edition']]}
```

Figure 1 Running the Application: - User Input for Database selection, Support and Confidence

List of Data Sets

The data sets consist of data items names and 20 tuples for each dataset. The items and translations are selected randomly based on data from each site. Generic data base is using the labels A,B,C, D, E as items names and transactions.

The Custom data is filled by the user to any other data to use. The custom data must follow the example provided to the user in other data in order to work.
NOTE: Any data set can be changed as long as the file follows the same format

Data sets are included with the project package in the folder data. There are 4 data sets in CSV flat files:

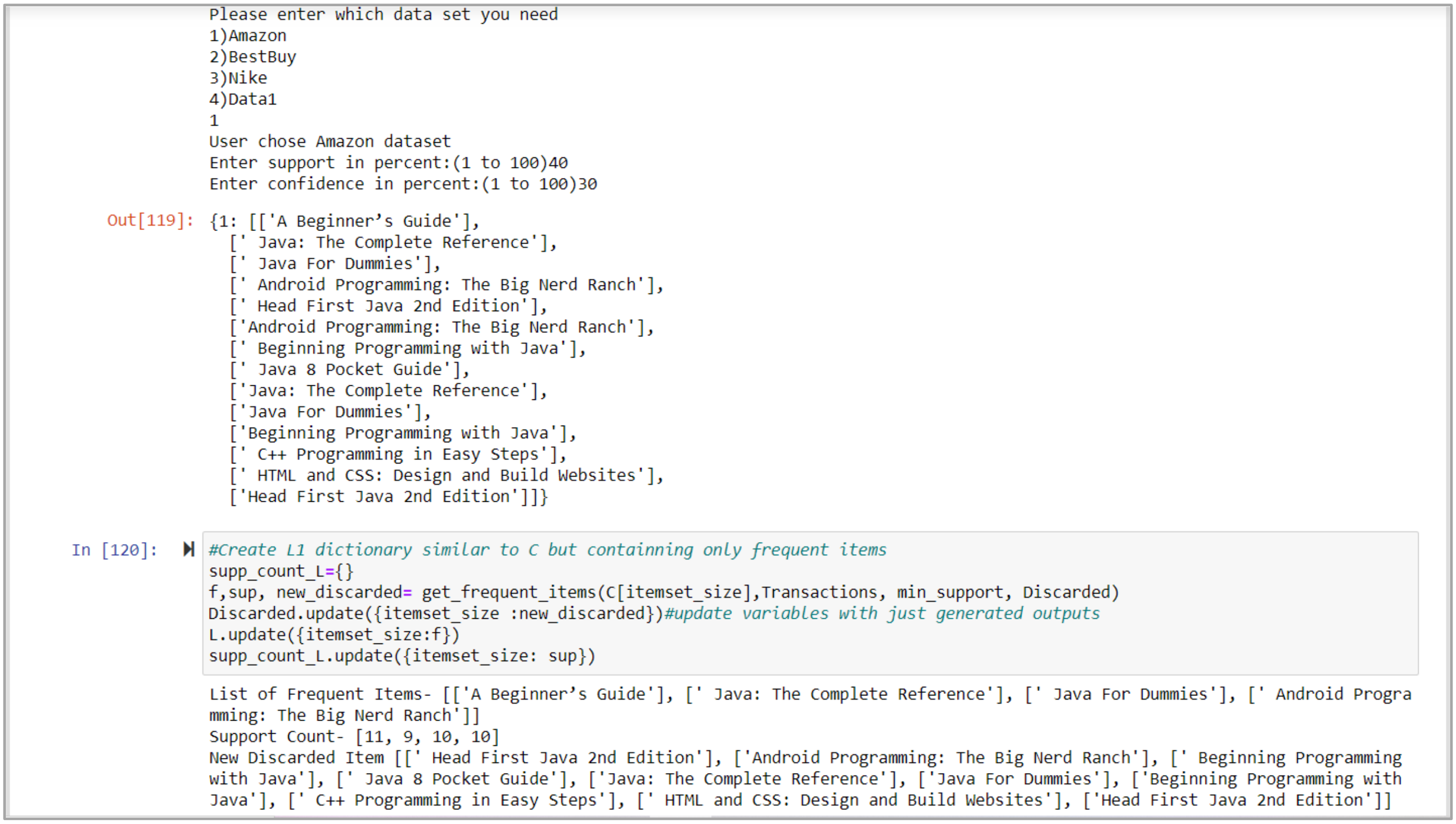
- Amazon
- BestBuy
- Nike
- Data1 which is generic dataset.

Testing Implementation and Data Set

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

1) Amazon

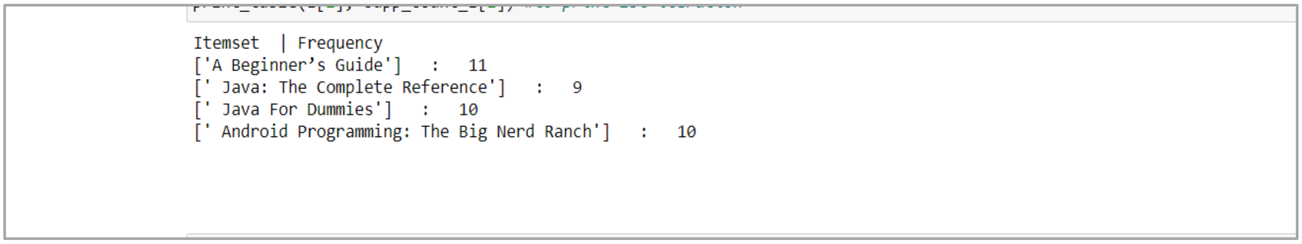
Running Amazon data set with support 40 and confidence 30



```
             Please enter which data set you need
             1)Amazon
             2)BestBuy
             3)Nike
             4)Data1
             1
             User chose Amazon dataset
             Enter support in percent:(1 to 100)40
             Enter confidence in percent:(1 to 100)30

   Out[119]: {1: [['A Beginner’s Guide'],
               [' Java: The Complete Reference'],
               [' Java For Dummies'],
               [' Android Programming: The Big Nerd Ranch'],
               [' Head First Java 2nd Edition'],
               ['Android Programming: The Big Nerd Ranch'],
               [' Beginning Programming with Java'],
               [' Java 8 Pocket Guide'],
               ['Java: The Complete Reference'],
               ['Java For Dummies'],
               ['Beginning Programming with Java'],
               [' C++ Programming in Easy Steps'],
               [' HTML and CSS: Design and Build Websites'],
               ['Head First Java 2nd Edition']]}
```

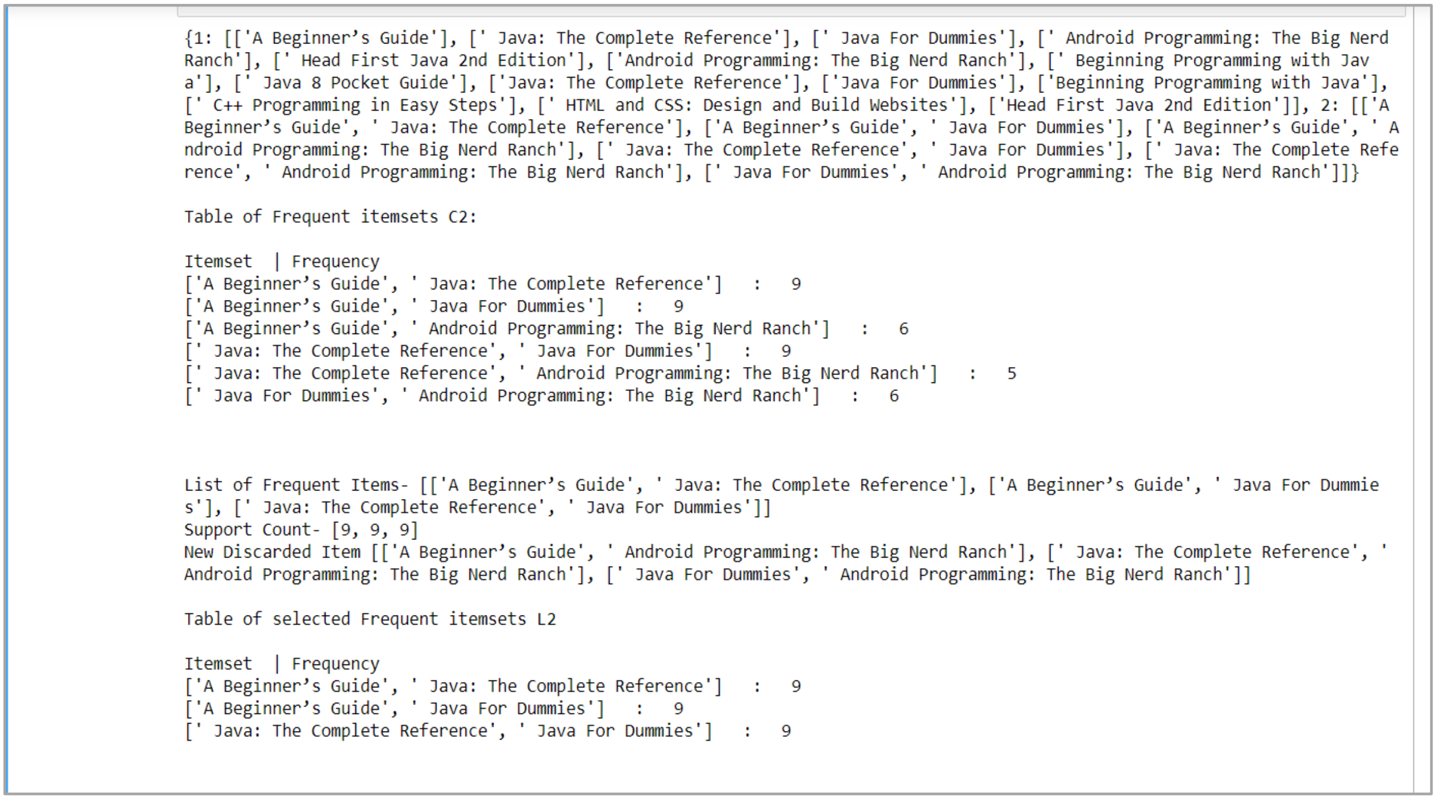
```
In [120]:   #Create L1 dictionary similar to C but containning only frequent items
            supp_count_L={}
            f,sup, new_discarded= get_frequent_items(C[itemset_size],Transactions, min_support, Discarded)
            Discarded.update({itemset_size :new_discarded})#update variables with just generated outputs
            L.update({itemset_size:f})
            supp_count_L.update({itemset_size: sup})

            List of Frequent Items- [['A Beginner’s Guide'], [' Java: The Complete Reference'], [' Java For Dummies'], [' Android Progra
            mming: The Big Nerd Ranch']]
            Support Count- [11, 9, 10, 10]
            New Discarded Item [[' Head First Java 2nd Edition'], ['Android Programming: The Big Nerd Ranch'], [' Beginning Programming
            with Java'], [' Java 8 Pocket Guide'], ['Java: The Complete Reference'], ['Java For Dummies'], ['Beginning Programming with
            Java'], [' C++ Programming in Easy Steps'], [' HTML and CSS: Design and Build Websites'], ['Head First Java 2nd Edition']]
```

Figure 1 Amazon Data Set Example Support 40, confidence 30



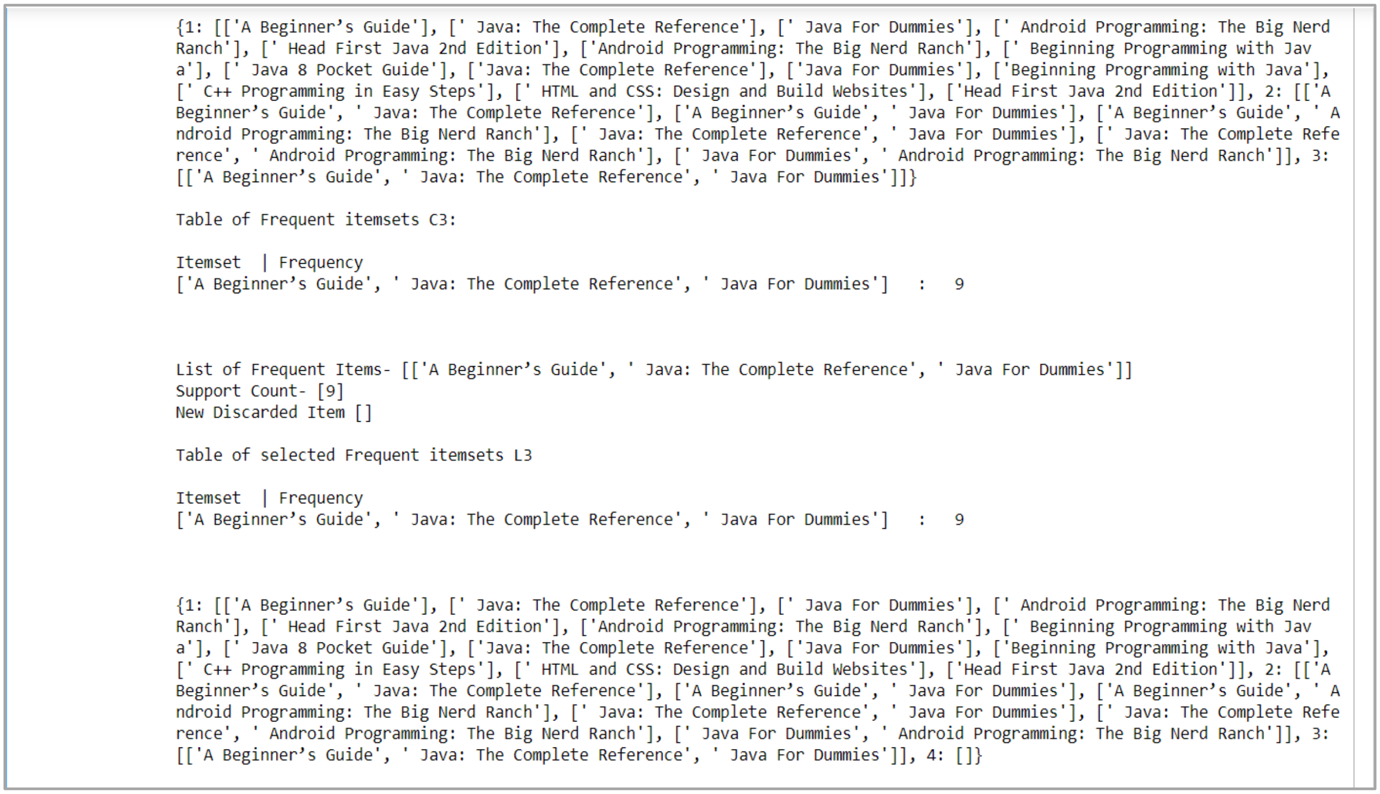
```
Itemset  | Frequency
['A Beginner’s Guide']   :   11
[' Java: The Complete Reference']   :   9
[' Java For Dummies']   :   10
[' Android Programming: The Big Nerd Ranch']   :   10
```

Figure 2 Amazon data set Example Support 40, confidence 30



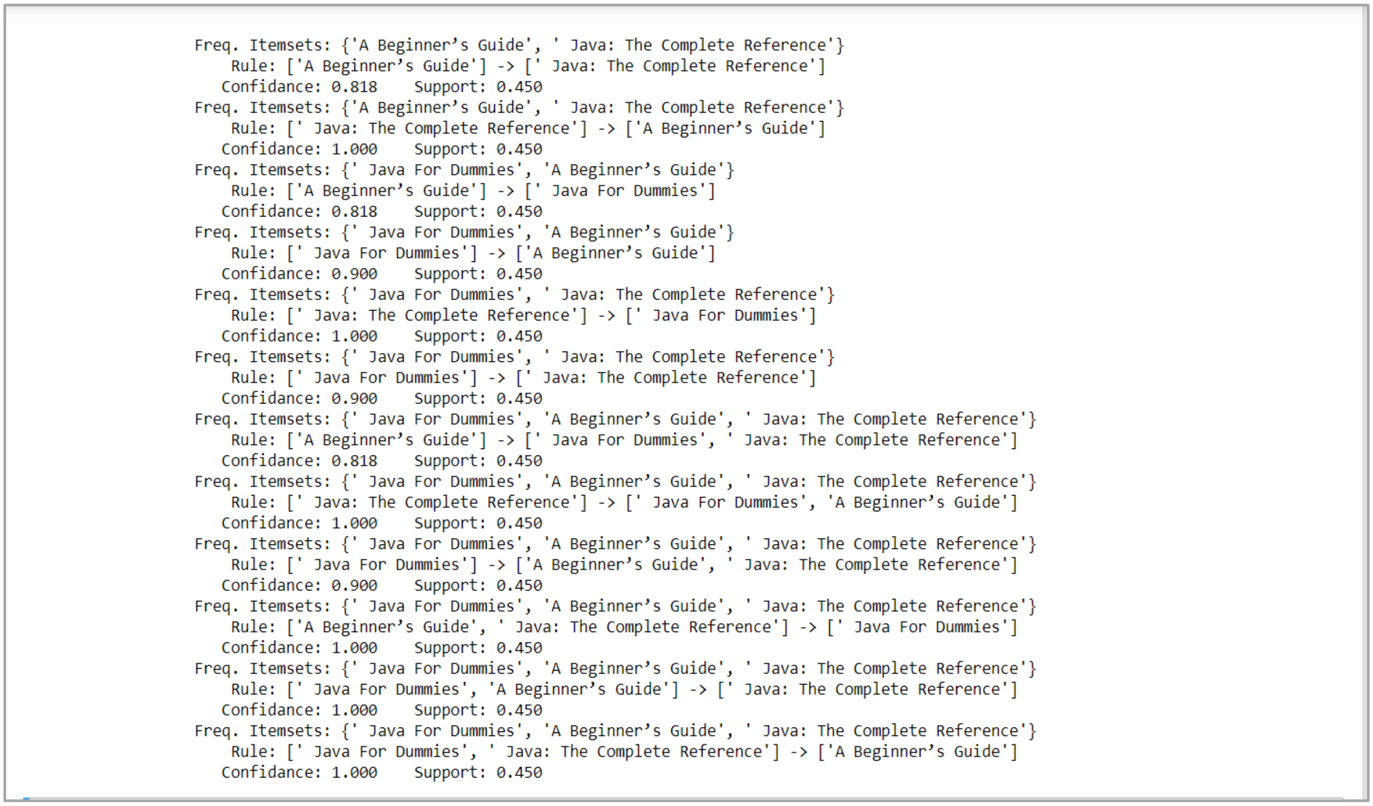
```
{1: [['A Beginner’s Guide'], [' Java: The Complete Reference'], [' Java For Dummies'], [' Android Programming: The Big Nerd
Ranch'], [' Head First Java 2nd Edition'], ['Android Programming: The Big Nerd Ranch'], [' Beginning Programming with Jav
a'], [' Java 8 Pocket Guide'], ['Java: The Complete Reference'], ['Java For Dummies'], ['Beginning Programming with Java'],
[' C++ Programming in Easy Steps'], [' HTML and CSS: Design and Build Websites'], ['Head First Java 2nd Edition']], 2: [['A
Beginner’s Guide', ' Java: The Complete Reference'], ['A Beginner’s Guide', ' Java For Dummies'], ['A Beginner’s Guide', ' A
ndroid Programming: The Big Nerd Ranch'], [' Java: The Complete Reference', ' Java For Dummies'], [' Java: The Complete Refe
rence', ' Android Programming: The Big Nerd Ranch'], [' Java For Dummies', ' Android Programming: The Big Nerd Ranch']]}

Table of Frequent itemsets C2:

Itemset  | Frequency
['A Beginner’s Guide', ' Java: The Complete Reference']   :   9
['A Beginner’s Guide', ' Java For Dummies']   :   9
['A Beginner’s Guide', ' Android Programming: The Big Nerd Ranch']   :   6
[' Java: The Complete Reference', ' Java For Dummies']   :   9
[' Java: The Complete Reference', ' Android Programming: The Big Nerd Ranch']   :   5
[' Java For Dummies', ' Android Programming: The Big Nerd Ranch']   :   6



List of Frequent Items- [['A Beginner’s Guide', ' Java: The Complete Reference'], ['A Beginner’s Guide', ' Java For Dummie
s'], [' Java: The Complete Reference', ' Java For Dummies']]
Support Count- [9, 9, 9]
New Discarded Item [['A Beginner’s Guide', ' Android Programming: The Big Nerd Ranch'], [' Java: The Complete Reference', '
Android Programming: The Big Nerd Ranch'], [' Java For Dummies', ' Android Programming: The Big Nerd Ranch']]

Table of selected Frequent itemsets L2

Itemset  | Frequency
['A Beginner’s Guide', ' Java: The Complete Reference']   :   9
['A Beginner’s Guide', ' Java For Dummies']   :   9
[' Java: The Complete Reference', ' Java For Dummies']   :   9
```

Figure 3 Amazon data set Example Support 40, confidence 30



```
{1: [['A Beginner’s Guide'], [' Java: The Complete Reference'], [' Java For Dummies'], [' Android Programming: The Big Nerd
Ranch'], [' Head First Java 2nd Edition'], ['Android Programming: The Big Nerd Ranch'], [' Beginning Programming with Jav
a'], [' Java 8 Pocket Guide'], ['Java: The Complete Reference'], ['Java For Dummies'], ['Beginning Programming with Java'],
[' C++ Programming in Easy Steps'], [' HTML and CSS: Design and Build Websites'], ['Head First Java 2nd Edition']], 2: [['A
Beginner’s Guide', ' Java: The Complete Reference'], ['A Beginner’s Guide', ' Java For Dummies'], ['A Beginner’s Guide', ' A
ndroid Programming: The Big Nerd Ranch'], [' Java: The Complete Reference', ' Java For Dummies'], [' Java: The Complete Refe
rence', ' Android Programming: The Big Nerd Ranch'], [' Java For Dummies', ' Android Programming: The Big Nerd Ranch']], 3:
[['A Beginner’s Guide', ' Java: The Complete Reference', ' Java For Dummies']]}

Table of Frequent itemsets C3:

Itemset  | Frequency
['A Beginner’s Guide', ' Java: The Complete Reference', ' Java For Dummies']   :   9



List of Frequent Items- [['A Beginner’s Guide', ' Java: The Complete Reference', ' Java For Dummies']]
Support Count- [9]
New Discarded Item []

Table of selected Frequent itemsets L3

Itemset  | Frequency
['A Beginner’s Guide', ' Java: The Complete Reference', ' Java For Dummies']   :   9



{1: [['A Beginner’s Guide'], [' Java: The Complete Reference'], [' Java For Dummies'], [' Android Programming: The Big Nerd
Ranch'], [' Head First Java 2nd Edition'], ['Android Programming: The Big Nerd Ranch'], [' Beginning Programming with Jav
a'], [' Java 8 Pocket Guide'], ['Java: The Complete Reference'], ['Java For Dummies'], ['Beginning Programming with Java'],
[' C++ Programming in Easy Steps'], [' HTML and CSS: Design and Build Websites'], ['Head First Java 2nd Edition']], 2: [['A
Beginner’s Guide', ' Java: The Complete Reference'], ['A Beginner’s Guide', ' Java For Dummies'], ['A Beginner’s Guide', ' A
ndroid Programming: The Big Nerd Ranch'], [' Java: The Complete Reference', ' Java For Dummies'], [' Java: The Complete Refe
rence', ' Android Programming: The Big Nerd Ranch'], [' Java For Dummies', ' Android Programming: The Big Nerd Ranch']], 3:
[['A Beginner’s Guide', ' Java: The Complete Reference', ' Java For Dummies']], 4: []}
```

Figure 4 Amazon data set Example Support 40, confidence 30

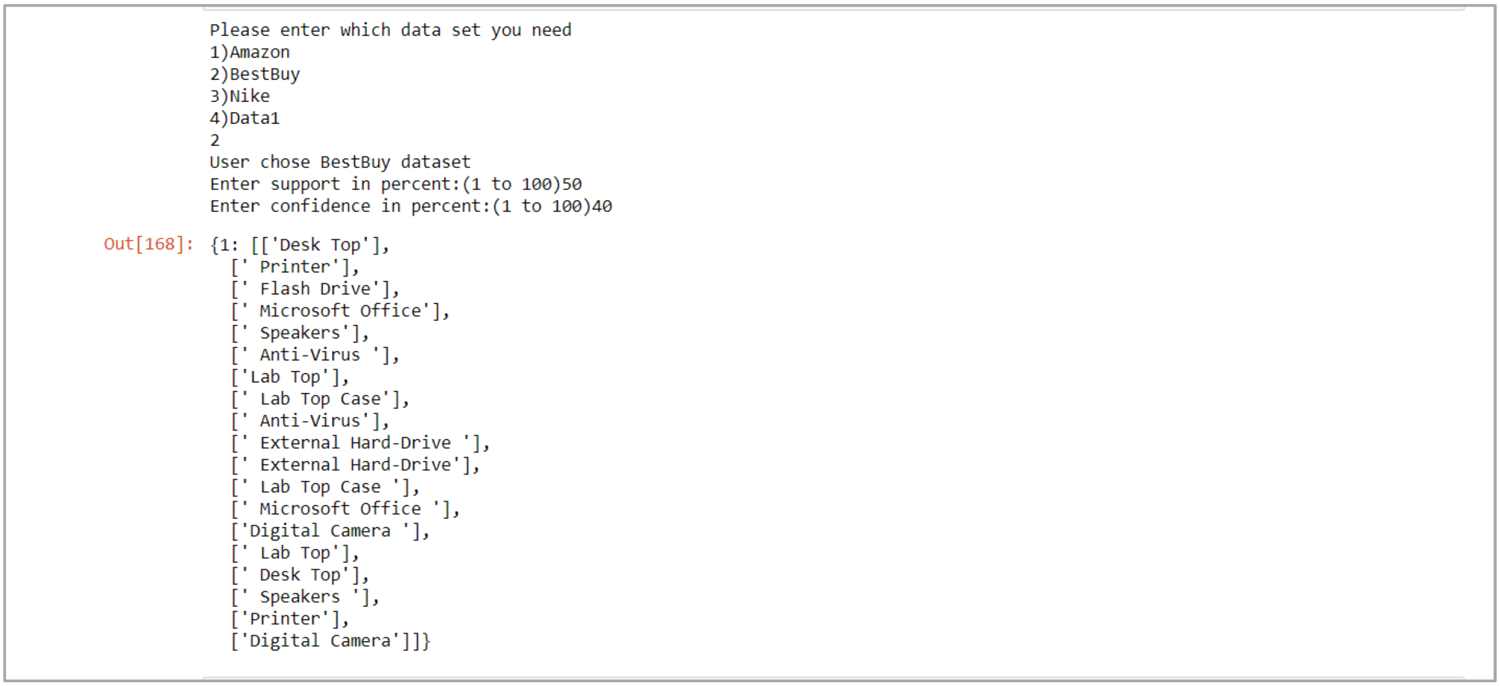


```
Freq. Itemsets: {'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: ['A Beginner’s Guide'] -> [' Java: The Complete Reference']
   Confidance: 0.818     Support: 0.450
Freq. Itemsets: {'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: [' Java: The Complete Reference'] -> ['A Beginner’s Guide']
   Confidance: 1.000     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide'}
    Rule: ['A Beginner’s Guide'] -> [' Java For Dummies']
   Confidance: 0.818     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide'}
    Rule: [' Java For Dummies'] -> ['A Beginner’s Guide']
   Confidance: 0.900     Support: 0.450
Freq. Itemsets: {' Java For Dummies', ' Java: The Complete Reference'}
    Rule: [' Java: The Complete Reference'] -> [' Java For Dummies']
   Confidance: 1.000     Support: 0.450
Freq. Itemsets: {' Java For Dummies', ' Java: The Complete Reference'}
    Rule: [' Java For Dummies'] -> [' Java: The Complete Reference']
   Confidance: 0.900     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: ['A Beginner’s Guide'] -> [' Java For Dummies', ' Java: The Complete Reference']
   Confidance: 0.818     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: [' Java: The Complete Reference'] -> [' Java For Dummies', 'A Beginner’s Guide']
   Confidance: 1.000     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: [' Java For Dummies'] -> ['A Beginner’s Guide', ' Java: The Complete Reference']
   Confidance: 0.900     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: ['A Beginner’s Guide', ' Java: The Complete Reference'] -> [' Java For Dummies']
   Confidance: 1.000     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: [' Java For Dummies', 'A Beginner’s Guide'] -> [' Java: The Complete Reference']
   Confidance: 1.000     Support: 0.450
Freq. Itemsets: {' Java For Dummies', 'A Beginner’s Guide', ' Java: The Complete Reference'}
    Rule: [' Java For Dummies', ' Java: The Complete Reference'] -> ['A Beginner’s Guide']
   Confidance: 1.000     Support: 0.450
```

Figure 5 Amazon data set Example Support 40, confidence 30 Final Association Rules

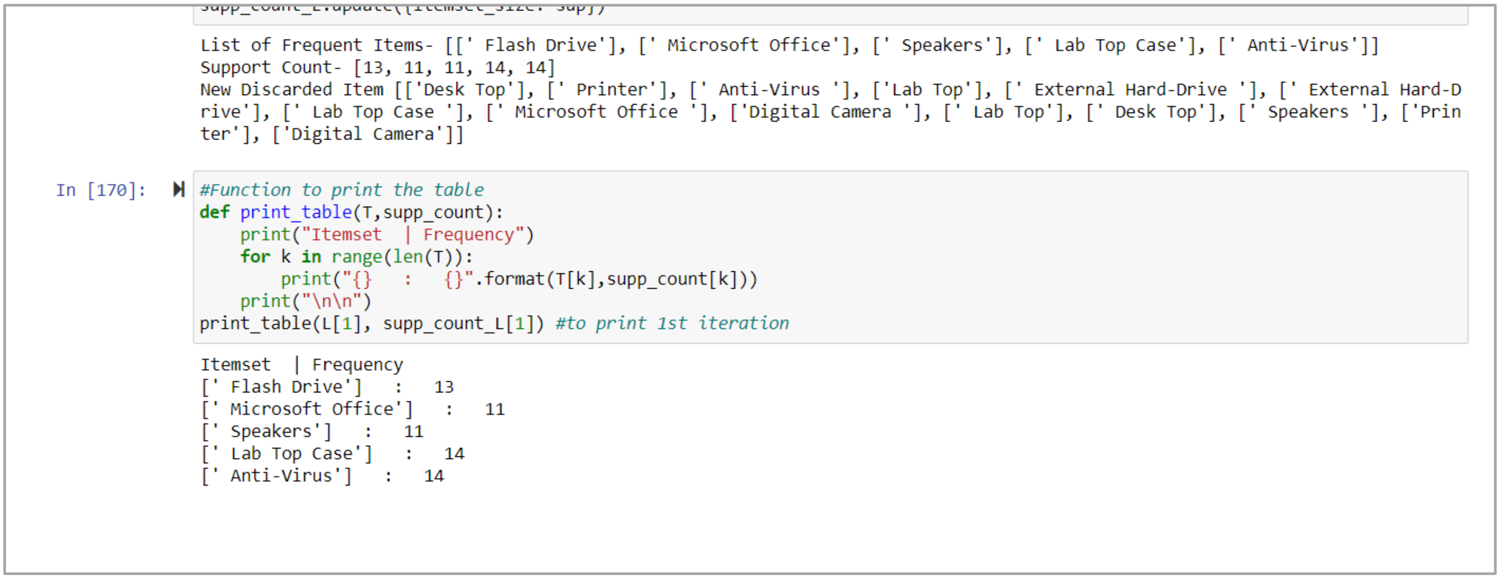
2) Best Buy

Running Best Buy data set with support 50 and confidence 40



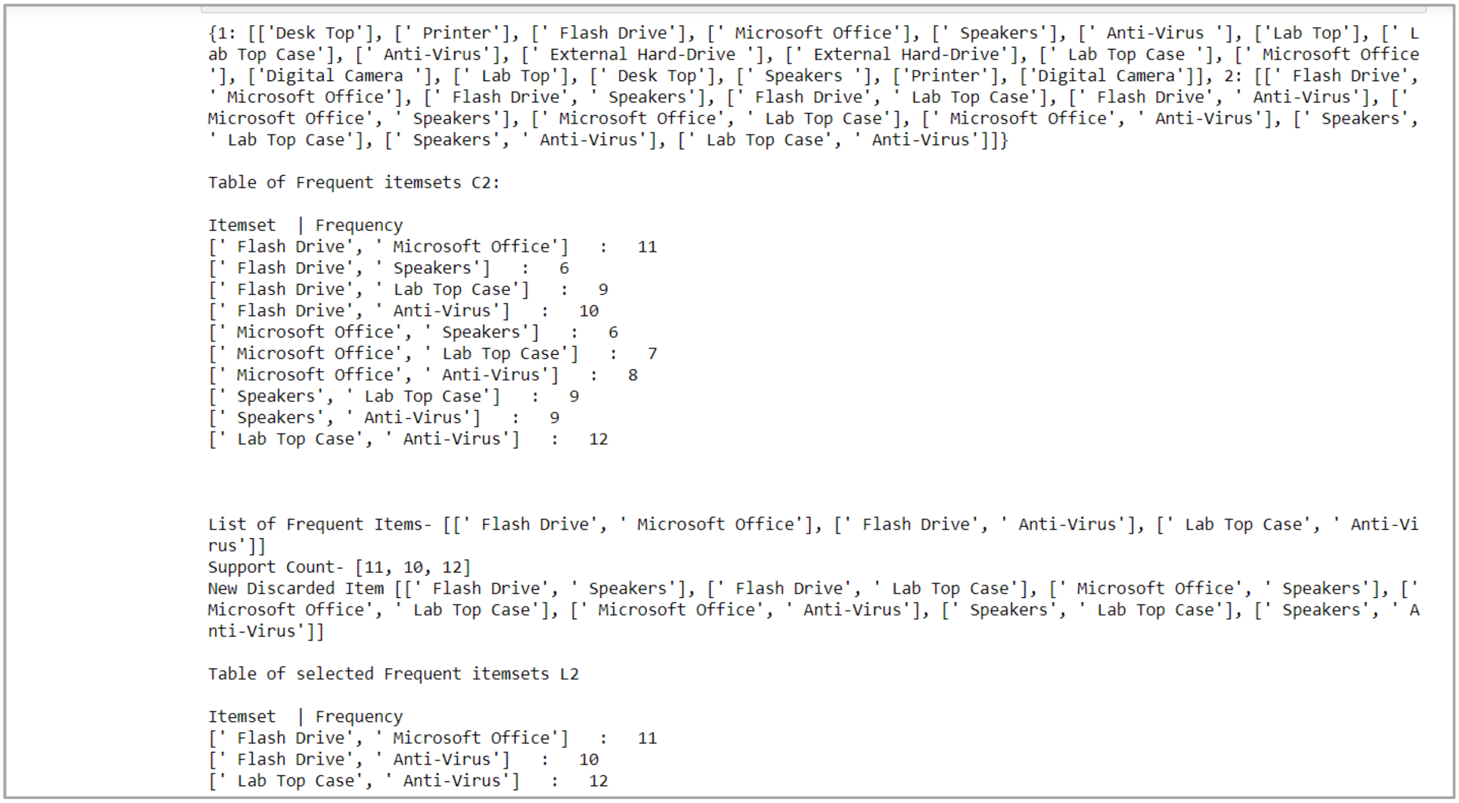
```
         Please enter which data set you need
         1)Amazon
         2)BestBuy
         3)Nike
         4)Data1
         2
         User chose BestBuy dataset
         Enter support in percent:(1 to 100)50
         Enter confidence in percent:(1 to 100)40

Out[168]: {1: [['Desk Top'],
            [' Printer'],
            [' Flash Drive'],
            [' Microsoft Office'],
            [' Speakers'],
            [' Anti-Virus '],
            ['Lab Top'],
            [' Lab Top Case'],
            [' Anti-Virus'],
            [' External Hard-Drive '],
            [' External Hard-Drive'],
            [' Lab Top Case '],
            [' Microsoft Office '],
            ['Digital Camera '],
            [' Lab Top'],
            [' Desk Top'],
            [' Speakers '],
            ['Printer'],
            ['Digital Camera']]}
```

Figure 1 Best Buy Data Set Example Support 50 , Confidence 40



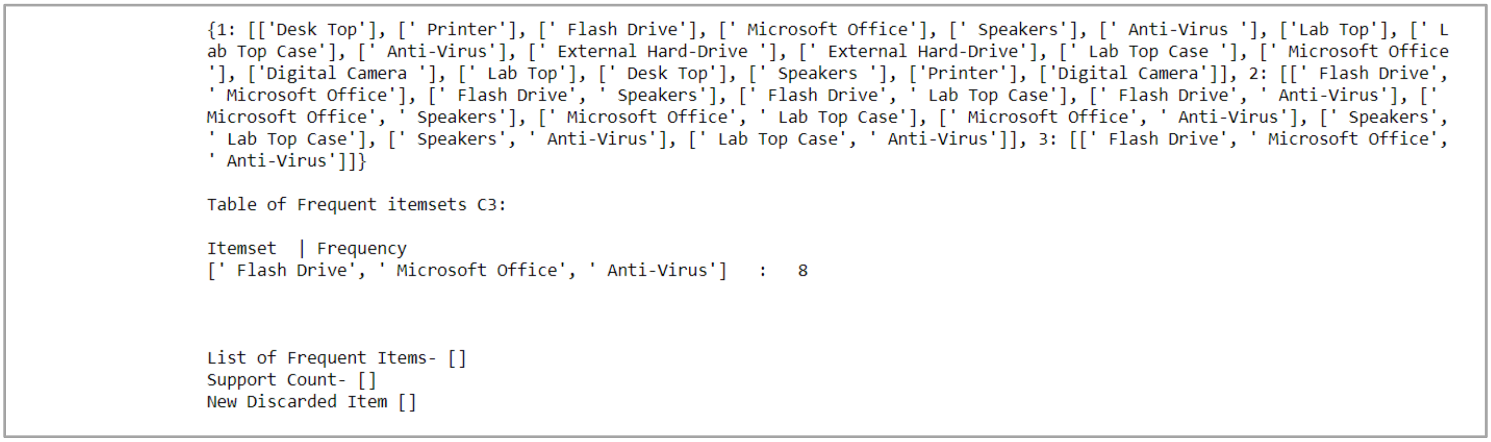
```
         supp_count_L.update({itemset_size: supp})
         List of Frequent Items- [[' Flash Drive'], [' Microsoft Office'], [' Speakers'], [' Lab Top Case'], [' Anti-Virus']]
         Support Count- [13, 11, 11, 14, 14]
         New Discarded Item [['Desk Top'], [' Printer'], [' Anti-Virus '], ['Lab Top'], [' External Hard-Drive '], [' External Hard-D
         rive'], [' Lab Top Case '], [' Microsoft Office '], ['Digital Camera '], [' Lab Top'], [' Desk Top'], [' Speakers '], ['Prin
         ter'], ['Digital Camera']]

In [170]: #Function to print the table
         def print_table(T,supp_count):
             print("Itemset   | Frequency")
             for k in range(len(T)):
                 print("{}   :   {}".format(T[k],supp_count[k]))
             print("\n\n")
         print_table(L[1], supp_count_L[1]) #to print 1st iteration

         Itemset  | Frequency
         [' Flash Drive']   :   13
         [' Microsoft Office']   :   11
         [' Speakers']   :   11
         [' Lab Top Case']   :   14
         [' Anti-Virus']   :   14
```

Figure 2 Best Buy Data Set Example Support 50 , Confidence 40



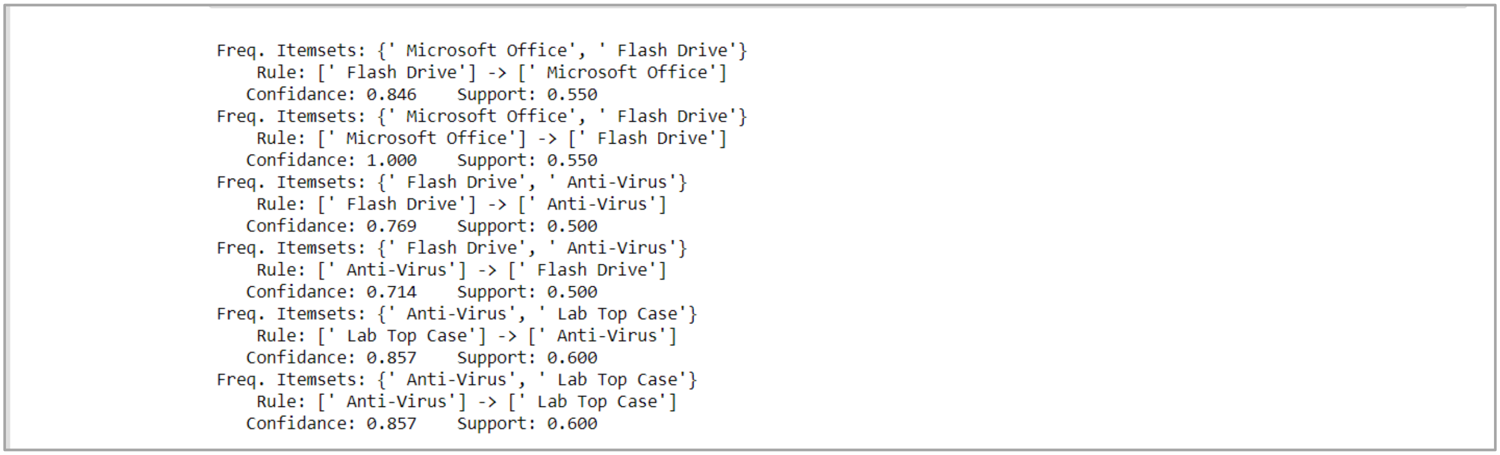
```
{1: [['Desk Top'], [' Printer'], [' Flash Drive'], [' Microsoft Office'], [' Speakers'], [' Anti-Virus '], ['Lab Top'], [' L
ab Top Case'], [' Anti-Virus'], [' External Hard-Drive '], [' External Hard-Drive'], [' Lab Top Case '], [' Microsoft Office
'], ['Digital Camera '], [' Lab Top'], [' Desk Top'], [' Speakers '], ['Printer'], ['Digital Camera']], 2: [[' Flash Drive',
' Microsoft Office'], [' Flash Drive', ' Speakers'], [' Flash Drive', ' Lab Top Case'], [' Flash Drive', ' Anti-Virus'], ['
Microsoft Office', ' Speakers'], [' Microsoft Office', ' Lab Top Case'], [' Microsoft Office', ' Anti-Virus'], [' Speakers',
' Lab Top Case'], [' Speakers', ' Anti-Virus'], [' Lab Top Case', ' Anti-Virus']]}

Table of Frequent itemsets C2:

Itemset   | Frequency
[' Flash Drive', ' Microsoft Office']   :   11
[' Flash Drive', ' Speakers']   :   6
[' Flash Drive', ' Lab Top Case']   :   9
[' Flash Drive', ' Anti-Virus']   :   10
[' Microsoft Office', ' Speakers']   :   6
[' Microsoft Office', ' Lab Top Case']   :   7
[' Microsoft Office', ' Anti-Virus']   :   8
[' Speakers', ' Lab Top Case']   :   9
[' Speakers', ' Anti-Virus']   :   9
[' Lab Top Case', ' Anti-Virus']   :   12



List of Frequent Items- [[' Flash Drive', ' Microsoft Office'], [' Flash Drive', ' Anti-Virus'], [' Lab Top Case', ' Anti-Vi
rus']]
Support Count- [11, 10, 12]
New Discarded Item [[' Flash Drive', ' Speakers'], [' Flash Drive', ' Lab Top Case'], [' Microsoft Office', ' Speakers'], ['
Microsoft Office', ' Lab Top Case'], [' Microsoft Office', ' Anti-Virus'], [' Speakers', ' Lab Top Case'], [' Speakers', ' A
nti-Virus']]

Table of selected Frequent itemsets L2

Itemset   | Frequency
[' Flash Drive', ' Microsoft Office']   :   11
[' Flash Drive', ' Anti-Virus']   :   10
[' Lab Top Case', ' Anti-Virus']   :   12
```

Figure 3 Best Buy Data Set Example Support 50 , Confidence 40



```
{1: [['Desk Top'], [' Printer'], [' Flash Drive'], [' Microsoft Office'], [' Speakers'], [' Anti-Virus '], ['Lab Top'], [' L
ab Top Case'], [' Anti-Virus'], [' External Hard-Drive '], [' External Hard-Drive'], [' Lab Top Case '], [' Microsoft Office
'], ['Digital Camera '], [' Lab Top'], [' Desk Top'], [' Speakers '], ['Printer'], ['Digital Camera']], 2: [[' Flash Drive',
' Microsoft Office'], [' Flash Drive', ' Speakers'], [' Flash Drive', ' Lab Top Case'], [' Flash Drive', ' Anti-Virus'], ['
Microsoft Office', ' Speakers'], [' Microsoft Office', ' Lab Top Case'], [' Microsoft Office', ' Anti-Virus'], [' Speakers',
' Lab Top Case'], [' Speakers', ' Anti-Virus'], [' Lab Top Case', ' Anti-Virus']], 3: [[' Flash Drive', ' Microsoft Office',
' Anti-Virus']]}

Table of Frequent itemsets C3:

Itemset   | Frequency
[' Flash Drive', ' Microsoft Office', ' Anti-Virus']   :   8



List of Frequent Items- []
Support Count- []
New Discarded Item []
```

Figure 4 Best Buy Data Set Example Support 50 , Confidence 40

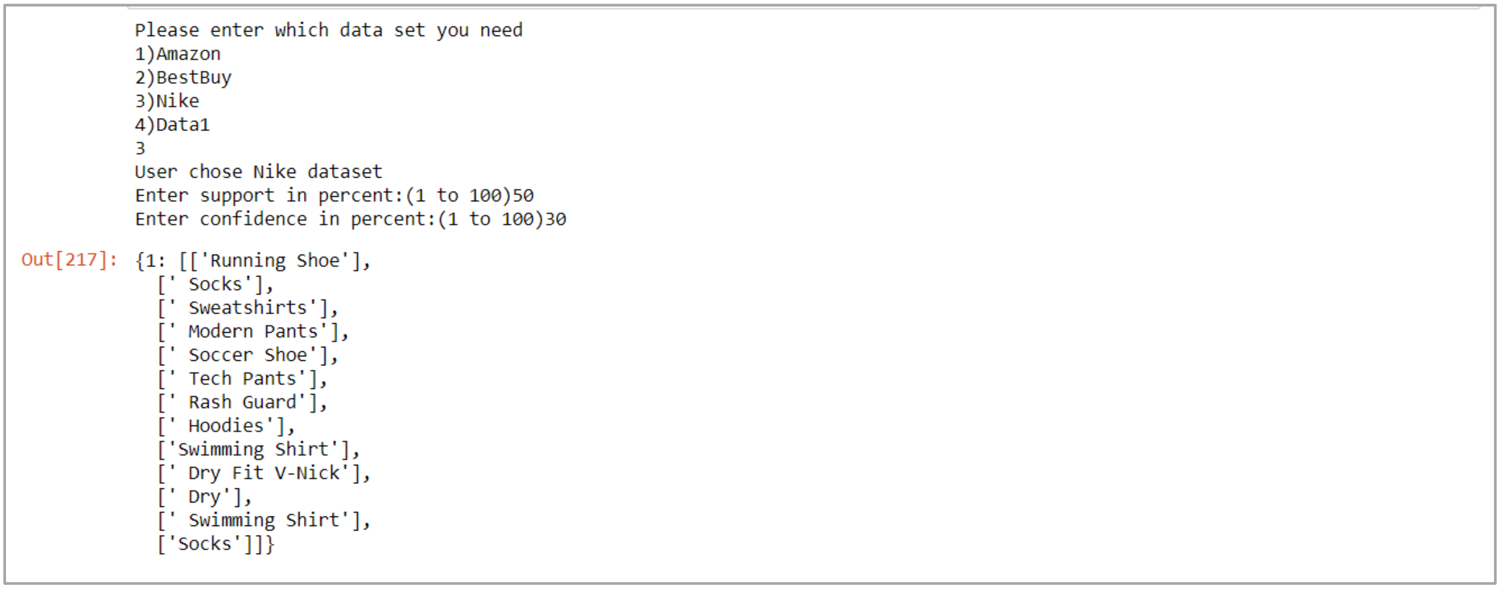


```
Freq. Itemsets: {' Microsoft Office', ' Flash Drive'}
    Rule: [' Flash Drive'] -> [' Microsoft Office']
   Confidance: 0.846    Support: 0.550
Freq. Itemsets: {' Microsoft Office', ' Flash Drive'}
    Rule: [' Microsoft Office'] -> [' Flash Drive']
   Confidance: 1.000    Support: 0.550
Freq. Itemsets: {' Flash Drive', ' Anti-Virus'}
    Rule: [' Flash Drive'] -> [' Anti-Virus']
   Confidance: 0.769    Support: 0.500
Freq. Itemsets: {' Flash Drive', ' Anti-Virus'}
    Rule: [' Anti-Virus'] -> [' Flash Drive']
   Confidance: 0.714    Support: 0.500
Freq. Itemsets: {' Anti-Virus', ' Lab Top Case'}
    Rule: [' Lab Top Case'] -> [' Anti-Virus']
   Confidance: 0.857    Support: 0.600
Freq. Itemsets: {' Anti-Virus', ' Lab Top Case'}
    Rule: [' Anti-Virus'] -> [' Lab Top Case']
   Confidance: 0.857    Support: 0.600
```

Figure 5 Best Buy Data Set Example Support 50 , Confidence 40 Final Association Rules

3) Nike

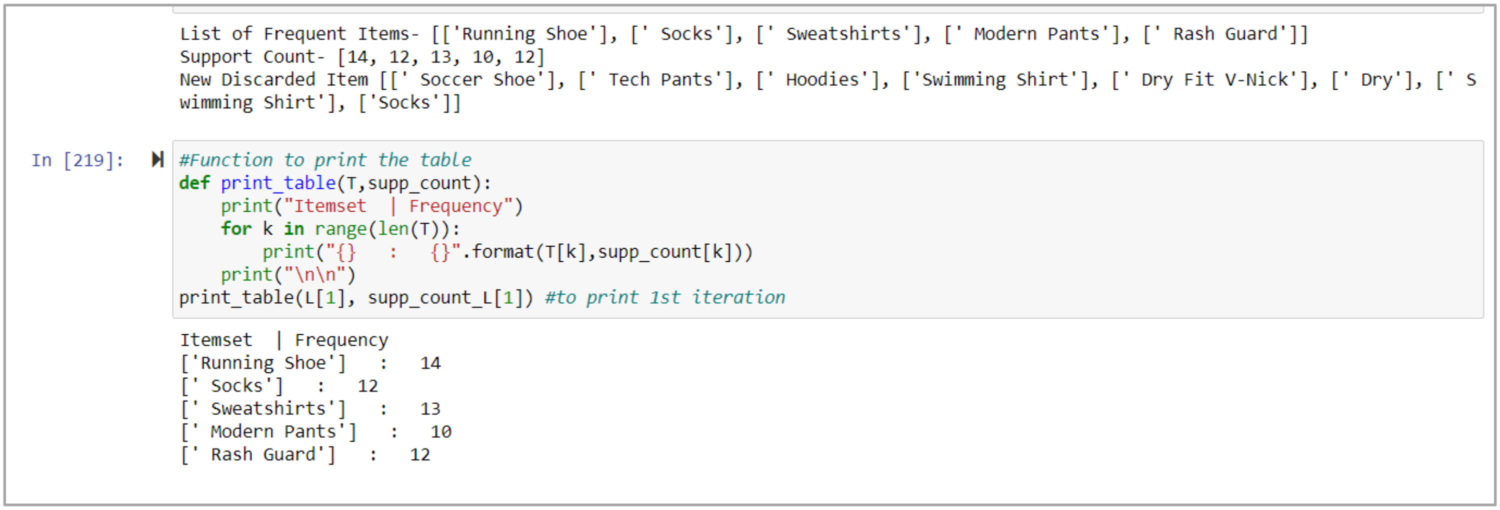
Running Nike Data Set example with Support 50 and Confidence 30



```
Please enter which data set you need
1)Amazon
2)BestBuy
3)Nike
4)Data1
3
User chose Nike dataset
Enter support in percent:(1 to 100)50
Enter confidence in percent:(1 to 100)30
```

```
Out[217]: {1: [['Running Shoe'],
            [' Socks'],
            [' Sweatshirts'],
            [' Modern Pants'],
            [' Soccer Shoe'],
            [' Tech Pants'],
            [' Rash Guard'],
            [' Hoodies'],
            ['Swimming Shirt'],
            [' Dry Fit V-Nick'],
            [' Dry'],
            [' Swimming Shirt'],
            ['Socks']]}
```

Figure 1 Nike Data Set Example Support 50, Confidence 30

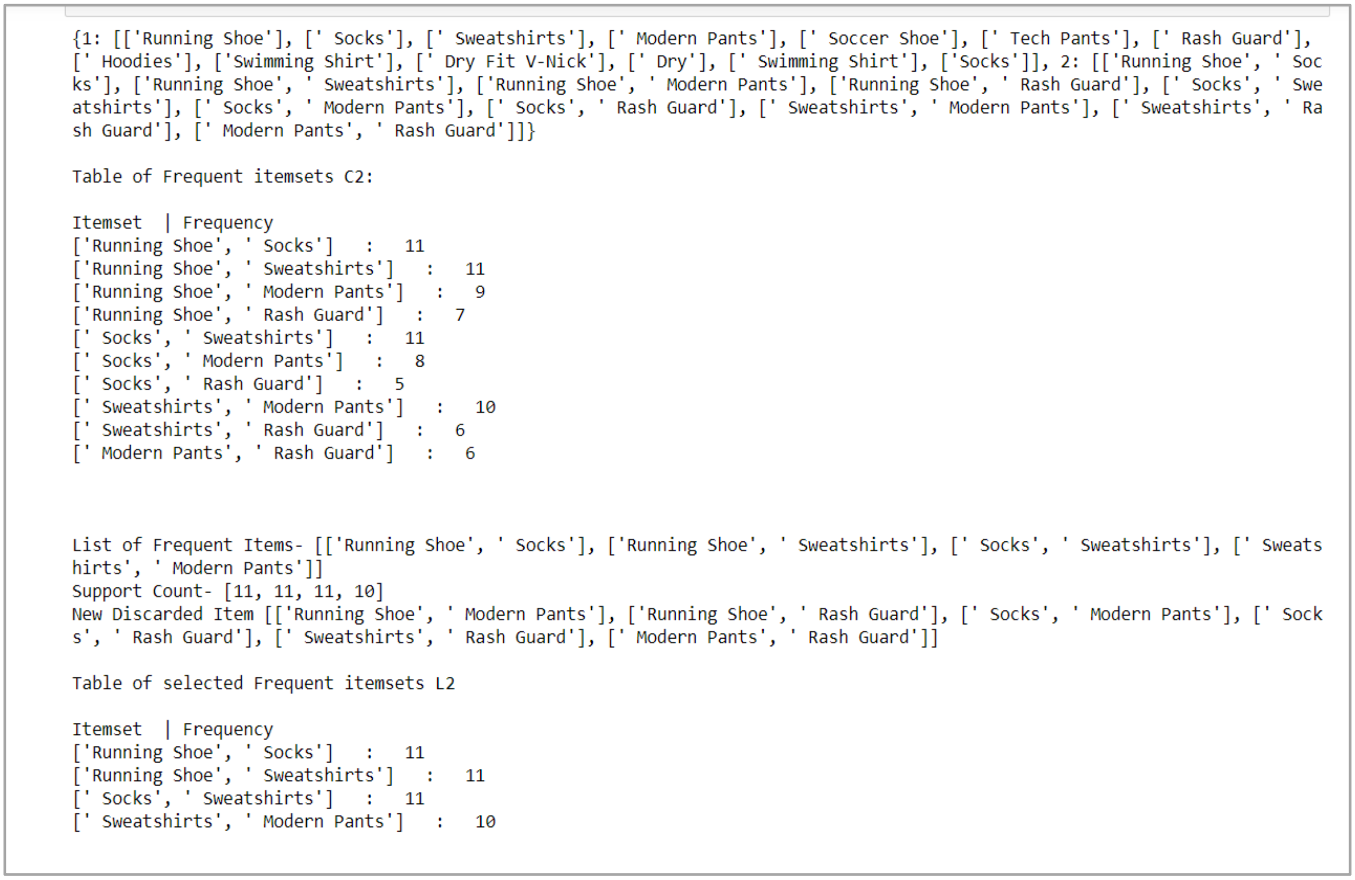


```
List of Frequent Items- [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Rash Guard']]
Support Count- [14, 12, 13, 10, 12]
New Discarded Item [[' Soccer Shoe'], [' Tech Pants'], [' Hoodies'], ['Swimming Shirt'], [' Dry Fit V-Nick'], [' Dry'], [' S
wimming Shirt'], ['Socks']]
```

```
In [219]: #Function to print the table
          def print_table(T,supp_count):
              print("Itemset   | Frequency")
              for k in range(len(T)):
                  print("{}   :   {}".format(T[k],supp_count[k]))
              print("\n\n")
          print_table(L[1], supp_count_L[1]) #to print 1st iteration
```

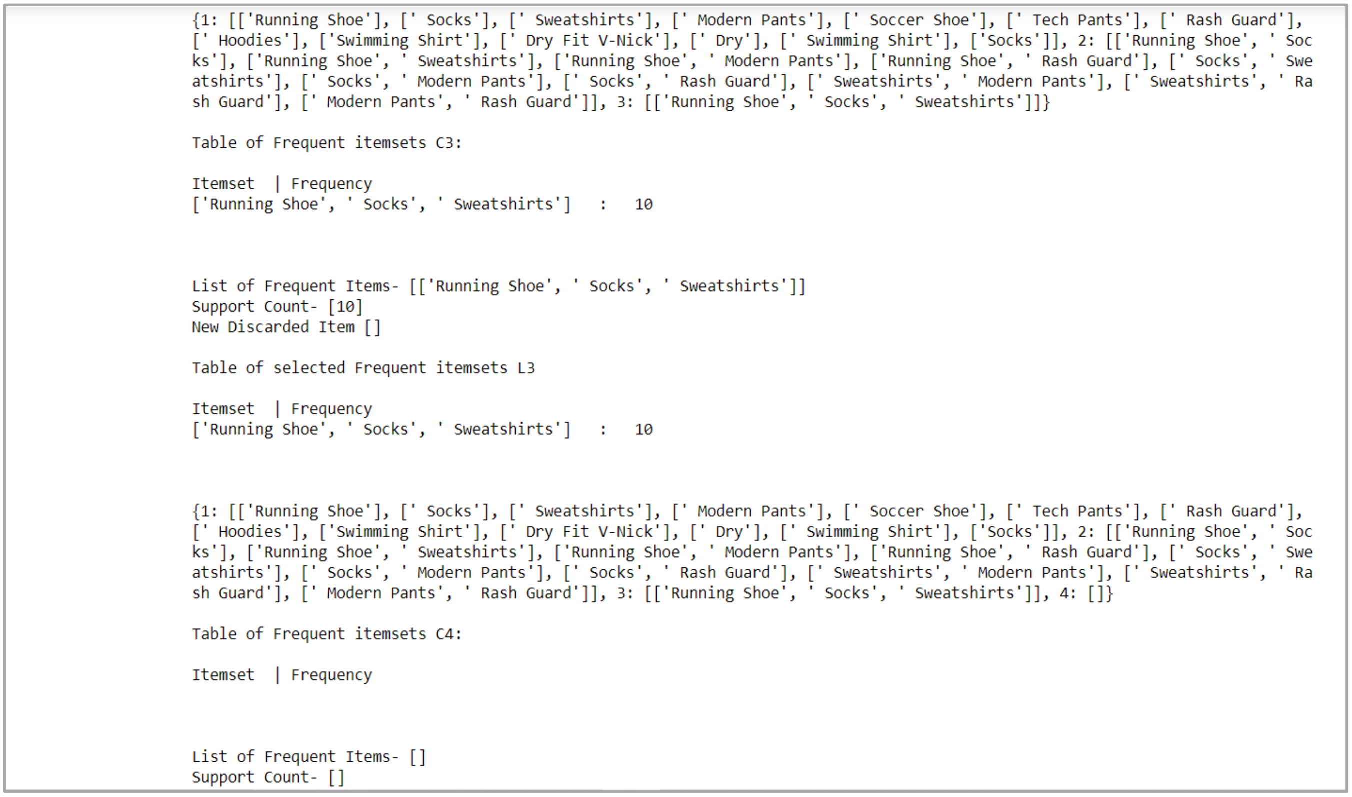
```
Itemset   | Frequency
['Running Shoe']   :   14
[' Socks']   :   12
[' Sweatshirts']   :   13
[' Modern Pants']   :   10
[' Rash Guard']   :   12
```

Figure 2 Nike Data Set Example Support 50, Confidence 30



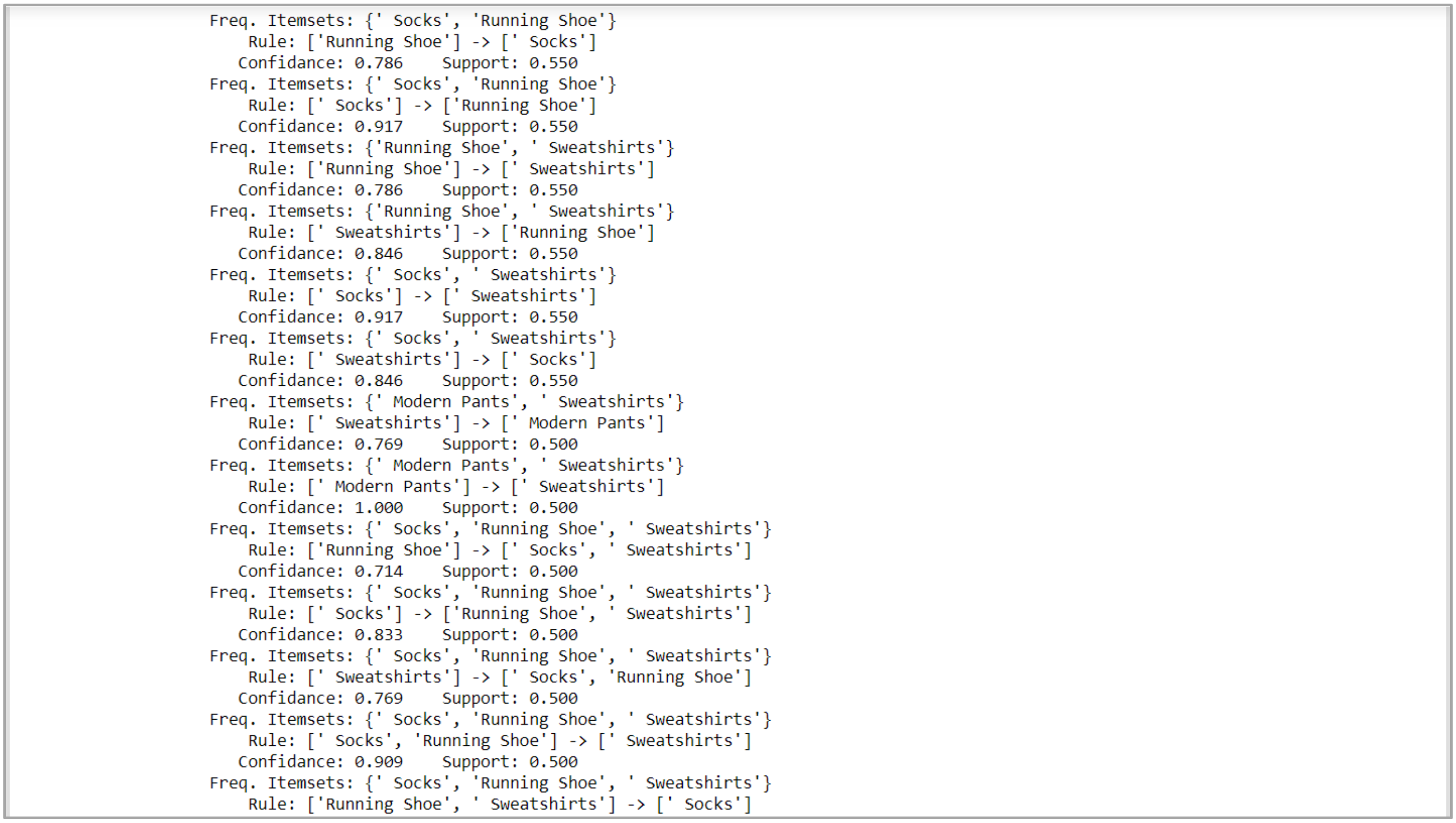
```
{1: [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Soccer Shoe'], [' Tech Pants'], [' Rash Guard'],
[' Hoodies'], ['Swimming Shirt'], [' Dry Fit V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']], 2: [['Running Shoe', ' Soc
ks'], ['Running Shoe', ' Sweatshirts'], ['Running Shoe', ' Modern Pants'], ['Running Shoe', ' Rash Guard'], [' Socks', ' Swe
atshirts'], [' Socks', ' Modern Pants'], [' Socks', ' Rash Guard'], [' Sweatshirts', ' Modern Pants'], [' Sweatshirts', ' Ra
sh Guard'], [' Modern Pants', ' Rash Guard']]}

Table of Frequent itemsets C2:

Itemset  | Frequency
['Running Shoe', ' Socks']   :   11
['Running Shoe', ' Sweatshirts']   :   11
['Running Shoe', ' Modern Pants']   :   9
['Running Shoe', ' Rash Guard']   :   7
[' Socks', ' Sweatshirts']   :   11
[' Socks', ' Modern Pants']   :   8
[' Socks', ' Rash Guard']   :   5
[' Sweatshirts', ' Modern Pants']   :   10
[' Sweatshirts', ' Rash Guard']   :   6
[' Modern Pants', ' Rash Guard']   :   6



List of Frequent Items- [['Running Shoe', ' Socks'], ['Running Shoe', ' Sweatshirts'], [' Socks', ' Sweatshirts'], [' Sweats
hirts', ' Modern Pants']]
Support Count- [11, 11, 11, 10]
New Discarded Item [['Running Shoe', ' Modern Pants'], ['Running Shoe', ' Rash Guard'], [' Socks', ' Modern Pants'], [' Sock
s', ' Rash Guard'], [' Sweatshirts', ' Rash Guard'], [' Modern Pants', ' Rash Guard']]

Table of selected Frequent itemsets L2

Itemset  | Frequency
['Running Shoe', ' Socks']   :   11
['Running Shoe', ' Sweatshirts']   :   11
[' Socks', ' Sweatshirts']   :   11
[' Sweatshirts', ' Modern Pants']   :   10
```

Figure 3 Nike Data Set Example Support 50, Confidence 30

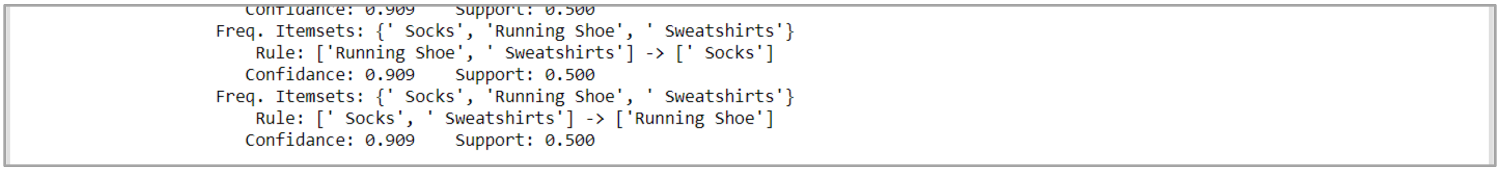


```
{1: [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Soccer Shoe'], [' Tech Pants'], [' Rash Guard'],
[' Hoodies'], ['Swimming Shirt'], [' Dry Fit V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']], 2: [['Running Shoe', ' Soc
ks'], ['Running Shoe', ' Sweatshirts'], ['Running Shoe', ' Modern Pants'], ['Running Shoe', ' Rash Guard'], [' Socks', ' Swe
atshirts'], [' Socks', ' Modern Pants'], [' Socks', ' Rash Guard'], [' Sweatshirts', ' Modern Pants'], [' Sweatshirts', ' Ra
sh Guard'], [' Modern Pants', ' Rash Guard']], 3: [['Running Shoe', ' Socks', ' Sweatshirts']]}

Table of Frequent itemsets C3:

Itemset  | Frequency
['Running Shoe', ' Socks', ' Sweatshirts']   :   10



List of Frequent Items- [['Running Shoe', ' Socks', ' Sweatshirts']]
Support Count- [10]
New Discarded Item []

Table of selected Frequent itemsets L3

Itemset  | Frequency
['Running Shoe', ' Socks', ' Sweatshirts']   :   10


{1: [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Soccer Shoe'], [' Tech Pants'], [' Rash Guard'],
[' Hoodies'], ['Swimming Shirt'], [' Dry Fit V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']], 2: [['Running Shoe', ' Soc
ks'], ['Running Shoe', ' Sweatshirts'], ['Running Shoe', ' Modern Pants'], ['Running Shoe', ' Rash Guard'], [' Socks', ' Swe
atshirts'], [' Socks', ' Modern Pants'], [' Socks', ' Rash Guard'], [' Sweatshirts', ' Modern Pants'], [' Sweatshirts', ' Ra
sh Guard'], [' Modern Pants', ' Rash Guard']], 3: [['Running Shoe', ' Socks', ' Sweatshirts']], 4: []}

Table of Frequent itemsets C4:

Itemset  | Frequency



List of Frequent Items- []
Support Count- []
```

Figure 4 Nike Data Set Example Support 50, Confidence 30



```
Freq. Itemsets: {' Socks', 'Running Shoe'}
    Rule: ['Running Shoe'] -> [' Socks']
   Confidance: 0.786    Support: 0.550
Freq. Itemsets: {' Socks', 'Running Shoe'}
    Rule: [' Socks'] -> ['Running Shoe']
   Confidance: 0.917    Support: 0.550
Freq. Itemsets: {'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe'] -> [' Sweatshirts']
   Confidance: 0.786    Support: 0.550
Freq. Itemsets: {'Running Shoe', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> ['Running Shoe']
   Confidance: 0.846    Support: 0.550
Freq. Itemsets: {' Socks', ' Sweatshirts'}
    Rule: [' Socks'] -> [' Sweatshirts']
   Confidance: 0.917    Support: 0.550
Freq. Itemsets: {' Socks', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> [' Socks']
   Confidance: 0.846    Support: 0.550
Freq. Itemsets: {' Modern Pants', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> [' Modern Pants']
   Confidance: 0.769    Support: 0.500
Freq. Itemsets: {' Modern Pants', ' Sweatshirts'}
    Rule: [' Modern Pants'] -> [' Sweatshirts']
   Confidance: 1.000    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe'] -> [' Socks', ' Sweatshirts']
   Confidance: 0.714    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Socks'] -> ['Running Shoe', ' Sweatshirts']
   Confidance: 0.833    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> [' Socks', 'Running Shoe']
   Confidance: 0.769    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Socks', 'Running Shoe'] -> [' Sweatshirts']
   Confidance: 0.909    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe', ' Sweatshirts'] -> [' Socks']
```

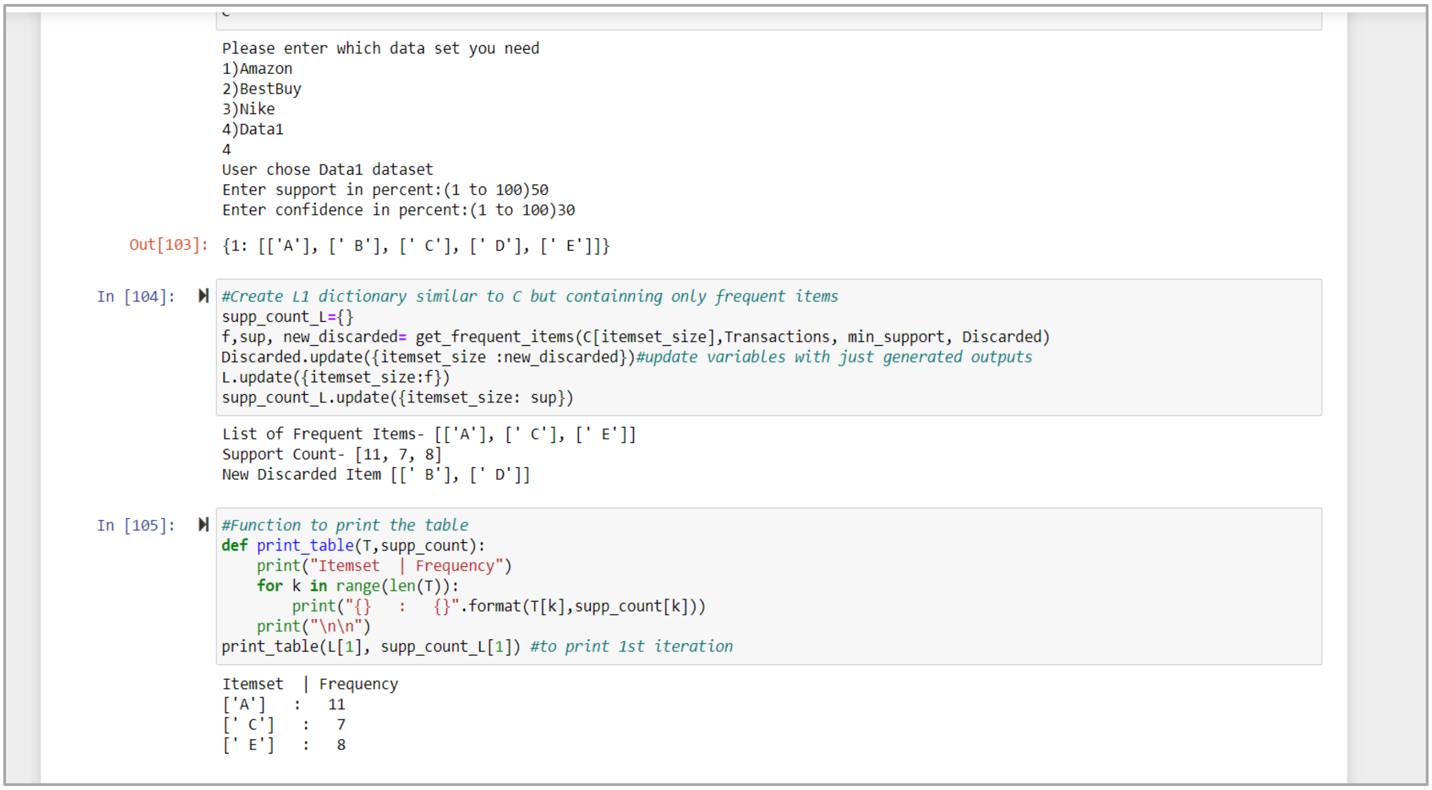


```
   Confidance: 0.909    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe', ' Sweatshirts'] -> [' Socks']
   Confidance: 0.909    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Socks', ' Sweatshirts'] -> ['Running Shoe']
   Confidance: 0.909    Support: 0.500
```

Figure 5,6 Nike Data Set Example Support 50, Confidence 30 Final Association Rules

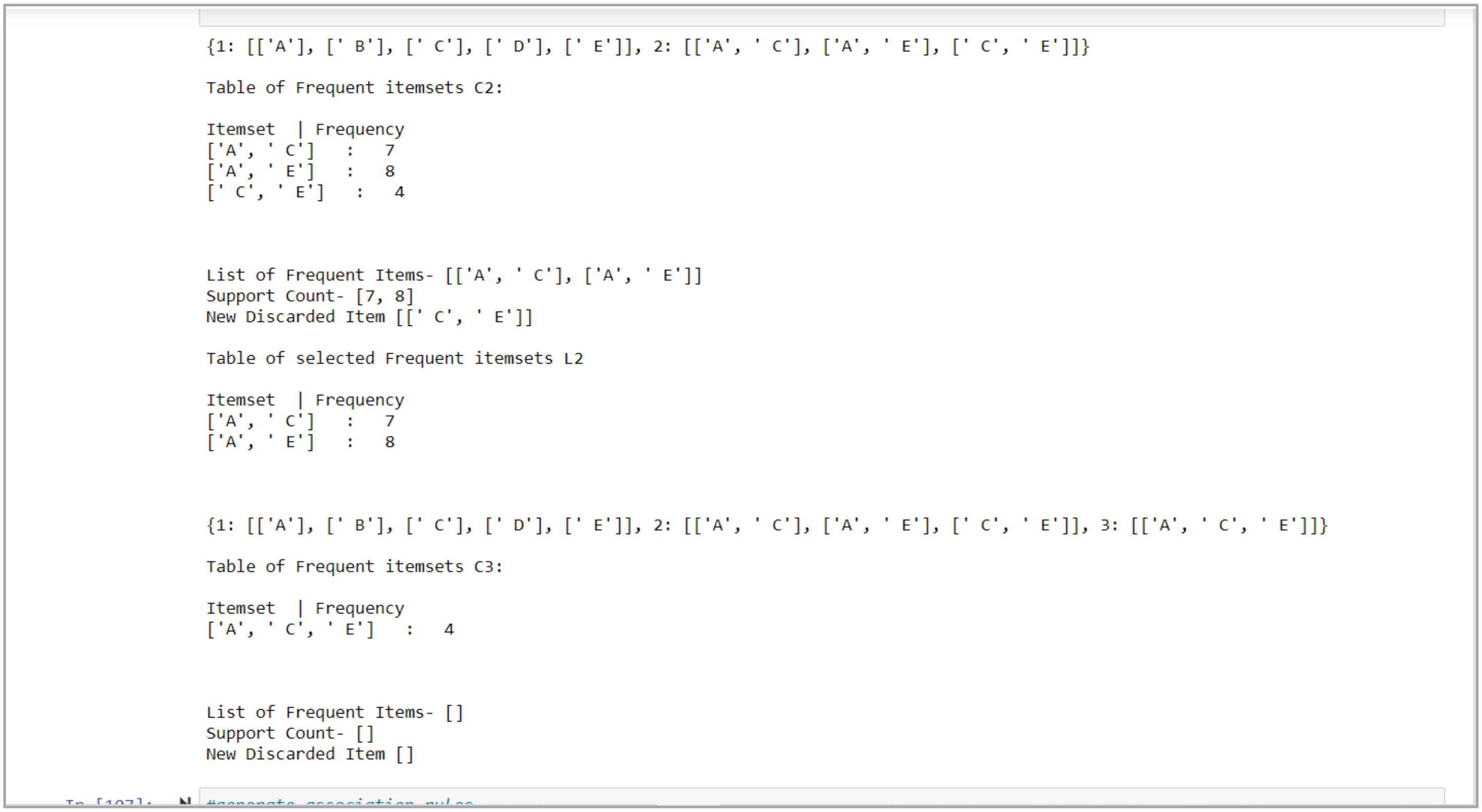
4)Data1 – Generic Dataset

Running Data1 Data Set with Support 50 and Confidence 30



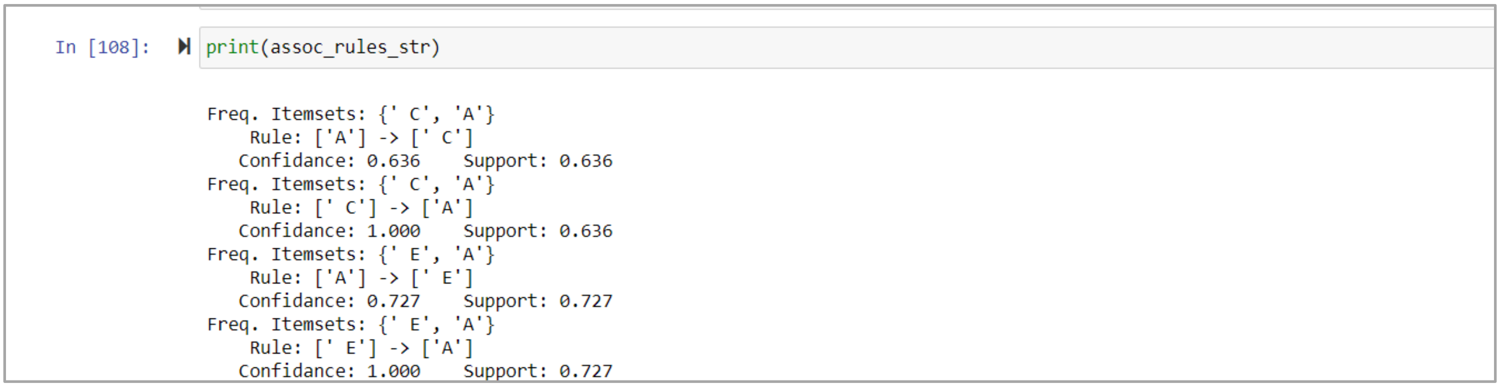
```
Please enter which data set you need
1)Amazon
2)BestBuy
3)Nike
4)Data1
4
User chose Data1 dataset
Enter support in percent:(1 to 100)50
Enter confidence in percent:(1 to 100)30

Out[103]: {1: [['A'], [' B'], [' C'], [' D'], [' E']]}

In [104]:  #Create L1 dictionary similar to C but containning only frequent items
           supp_count_L={}
           f,sup, new_discarded= get_frequent_items(C[itemset_size],Transactions, min_support, Discarded)
           Discarded.update({itemset_size :new_discarded})#update variables with just generated outputs
           L.update({itemset_size:f})
           supp_count_L.update({itemset_size: sup})

           List of Frequent Items- [['A'], [' C'], [' E']]
           Support Count- [11, 7, 8]
           New Discarded Item [[' B'], [' D']]

In [105]:  #Function to print the table
           def print_table(T,supp_count):
               print("Itemset   | Frequency")
               for k in range(len(T)):
                   print("{}   :   {}".format(T[k],supp_count[k]))
               print("\n\n")
           print_table(L[1], supp_count_L[1]) #to print 1st iteration

           Itemset   | Frequency
           ['A']   :   11
           [' C']   :   7
           [' E']   :   8
```

Figure 1 Data1 Data Set Support 50, Confidence 30



```
{1: [['A'], [' B'], [' C'], [' D'], [' E']], 2: [['A', ' C'], ['A', ' E'], [' C', ' E']]}

Table of Frequent itemsets C2:

Itemset   | Frequency
['A', ' C']   :   7
['A', ' E']   :   8
[' C', ' E']   :   4



List of Frequent Items- [['A', ' C'], ['A', ' E']]
Support Count- [7, 8]
New Discarded Item [[' C', ' E']]

Table of selected Frequent itemsets L2

Itemset   | Frequency
['A', ' C']   :   7
['A', ' E']   :   8



{1: [['A'], [' B'], [' C'], [' D'], [' E']], 2: [['A', ' C'], ['A', ' E'], [' C', ' E']], 3: [['A', ' C', ' E']]}

Table of Frequent itemsets C3:

Itemset   | Frequency
['A', ' C', ' E']   :   4



List of Frequent Items- []
Support Count- []
New Discarded Item []

In [107]:  #generate association rules
```

Figure 2 Data1 Data Set Support 50, Confidence 30



```
In [108]:  print(assoc_rules_str)

           Freq. Itemsets: {' C', 'A'}
               Rule: ['A'] -> [' C']
              Confidance: 0.636    Support: 0.636
           Freq. Itemsets: {' C', 'A'}
               Rule: [' C'] -> ['A']
              Confidance: 1.000    Support: 0.636
           Freq. Itemsets: {' E', 'A'}
               Rule: ['A'] -> [' E']
              Confidance: 0.727    Support: 0.727
           Freq. Itemsets: {' E', 'A'}
               Rule: [' E'] -> ['A']
              Confidance: 1.000    Support: 0.727
```

Figure 3 Data1 Data Set Support 50, Confidence 30 Final Association Rule

Implementation Logs and Execution Output

Log Example

```
Please enter which data set you need
1)Amazon
2)BestBuy
3)Nike
4)Data1
3
User chose Nike dataset
Enter support in percent:(1 to 100)50
Enter confidence in percent:(1 to 100)30
Out[217]:
{1: [['Running Shoe'],
  [' Socks'],
  [' Sweatshirts'],
  [' Modern Pants'],
  [' Soccer Shoe'],
  [' Tech Pants'],
  [' Rash Guard'],
  [' Hoodies'],
  ['Swimming Shirt'],
  [' Dry Fit V-Nick'],
  [' Dry'],
  [' Swimming Shirt'],
  ['Socks']]}
List of Frequent Items- [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern
Pants'], [' Rash Guard']]
Support Count- [14, 12, 13, 10, 12]
New Discarded Item [[' Soccer Shoe'], [' Tech Pants'], [' Hoodies'], ['Swimming Shi
rt'], [' Dry Fit V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']]

Itemset  | Frequency
['Running Shoe']   :   14
[' Socks']   :   12
[' Sweatshirts']   :   13
[' Modern Pants']   :   10
[' Rash Guard']   :   12

{1: [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Soccer S
hoe'], [' Tech Pants'], [' Rash Guard'], [' Hoodies'], ['Swimming Shirt'], [' Dry F
it V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']], 2: [['Running Shoe', ' Sock
s'], ['Running Shoe', ' Sweatshirts'], ['Running Shoe', ' Modern Pants'], ['Running
Shoe', ' Rash Guard'], [' Socks', ' Sweatshirts'], [' Socks', ' Modern Pants'], ['
Socks', ' Rash Guard'], [' Sweatshirts', ' Modern Pants'], [' Sweatshirts', ' Rash
Guard'], [' Modern Pants', ' Rash Guard']]}

Table of Frequent itemsets C2:

Itemset  | Frequency
```

```
['Running Shoe', ' Socks']   :   11
['Running Shoe', ' Sweatshirts']   :   11
['Running Shoe', ' Modern Pants']   :   9
['Running Shoe', ' Rash Guard']   :   7
[' Socks', ' Sweatshirts']   :   11
[' Socks', ' Modern Pants']   :   8
[' Socks', ' Rash Guard']   :   5
[' Sweatshirts', ' Modern Pants']   :   10
[' Sweatshirts', ' Rash Guard']   :   6
[' Modern Pants', ' Rash Guard']   :   6



List of Frequent Items- [['Running Shoe', ' Socks'], ['Running Shoe', ' Sweatshirts
'], [' Socks', ' Sweatshirts'], [' Sweatshirts', ' Modern Pants']]
Support Count- [11, 11, 11, 10]
New Discarded Item [['Running Shoe', ' Modern Pants'], ['Running Shoe', ' Rash Guar
d'], [' Socks', ' Modern Pants'], [' Socks', ' Rash Guard'], [' Sweatshirts', ' Ras
h Guard'], [' Modern Pants', ' Rash Guard']]

Table of selected Frequent itemsets L2

Itemset  | Frequency
['Running Shoe', ' Socks']   :   11
['Running Shoe', ' Sweatshirts']   :   11
[' Socks', ' Sweatshirts']   :   11
[' Sweatshirts', ' Modern Pants']   :   10



{1: [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Soccer S
hoe'], [' Tech Pants'], [' Rash Guard'], [' Hoodies'], ['Swimming Shirt'], [' Dry F
it V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']], 2: [['Running Shoe', ' Sock
s'], ['Running Shoe', ' Sweatshirts'], ['Running Shoe', ' Modern Pants'], ['Running
Shoe', ' Rash Guard'], [' Socks', ' Sweatshirts'], [' Socks', ' Modern Pants'], ['
Socks', ' Rash Guard'], [' Sweatshirts', ' Modern Pants'], [' Sweatshirts', ' Rash
Guard'], [' Modern Pants', ' Rash Guard']], 3: [['Running Shoe', ' Socks', ' Sweats
hirts']]}

Table of Frequent itemsets C3:

Itemset  | Frequency
['Running Shoe', ' Socks', ' Sweatshirts']   :   10



List of Frequent Items- [['Running Shoe', ' Socks', ' Sweatshirts']]
Support Count- [10]
New Discarded Item []

Table of selected Frequent itemsets L3

Itemset  | Frequency
['Running Shoe', ' Socks', ' Sweatshirts']   :   10



{1: [['Running Shoe'], [' Socks'], [' Sweatshirts'], [' Modern Pants'], [' Soccer S
hoe'], [' Tech Pants'], [' Rash Guard'], [' Hoodies'], ['Swimming Shirt'], [' Dry F
it V-Nick'], [' Dry'], [' Swimming Shirt'], ['Socks']], 2: [['Running Shoe', ' Sock
s'], ['Running Shoe', ' Sweatshirts'], ['Running Shoe', ' Modern Pants'], ['Running
Shoe', ' Rash Guard'], [' Socks', ' Sweatshirts'], [' Socks', ' Modern Pants'], ['
Socks', ' Rash Guard'], [' Sweatshirts', ' Modern Pants'], [' Sweatshirts', ' Rash
Guard'], [' Modern Pants', ' Rash Guard']], 3: [['Running Shoe', ' Socks', ' Sweats
hirts']], 4: []}

Table of Frequent itemsets C4:

Itemset  | Frequency



List of Frequent Items- []
Support Count- []
New Discarded Item []
Freq. Itemsets: {' Socks', 'Running Shoe'}
    Rule: ['Running Shoe'] -> [' Socks']
   Confidance: 0.786    Support: 0.550
Freq. Itemsets: {' Socks', 'Running Shoe'}
    Rule: [' Socks'] -> ['Running Shoe']
   Confidance: 0.917    Support: 0.550
Freq. Itemsets: {'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe'] -> [' Sweatshirts']
   Confidance: 0.786    Support: 0.550
Freq. Itemsets: {'Running Shoe', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> ['Running Shoe']
   Confidance: 0.846    Support: 0.550
Freq. Itemsets: {' Socks', ' Sweatshirts'}
    Rule: [' Socks'] -> [' Sweatshirts']
   Confidance: 0.917    Support: 0.550
Freq. Itemsets: {' Socks', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> [' Socks']
   Confidance: 0.846    Support: 0.550
Freq. Itemsets: {' Modern Pants', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> [' Modern Pants']
   Confidance: 0.769    Support: 0.500
Freq. Itemsets: {' Modern Pants', ' Sweatshirts'}
    Rule: [' Modern Pants'] -> [' Sweatshirts']
   Confidance: 1.000    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe'] -> [' Socks', ' Sweatshirts']
   Confidance: 0.714    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Socks'] -> ['Running Shoe', ' Sweatshirts']
   Confidance: 0.833    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Sweatshirts'] -> [' Socks', 'Running Shoe']
   Confidance: 0.769    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Socks', 'Running Shoe'] -> [' Sweatshirts']
   Confidance: 0.909    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: ['Running Shoe', ' Sweatshirts'] -> [' Socks']
   Confidance: 0.909    Support: 0.500
Freq. Itemsets: {' Socks', 'Running Shoe', ' Sweatshirts'}
    Rule: [' Socks', ' Sweatshirts'] -> ['Running Shoe']
   Confidance: 0.909    Support: 0.500
```