Mid Term Project Report

Apriori Algorithm

Data Mining

CS-634-104

FALL-21

Professor: Yasser Abduallah

Department of Computer Science

Kavya Basavalingappa Umashankar

UCID – kbu2

Contents

Introduction:

Apriori Algorithm Overview:

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets if those item sets appear sufficiently often in the database.

It consists of two steps.

1. Self-Join

2. Pruning

Repeating these steps N times, where N is the number of items, in the last iteration you get frequent item sets containing N items.

- **Support:**

The support of an association pattern is the percentage of task-relevant data transactions for which the pattern is true.

To calculate the support of X -> Z:

Calculate the support of itemset {X, Z}, i.e., the number of transactions buying X, Z divided by the total number of transactions.

- **Confidence:**

Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern.

To calculate the confidence of X -> Z

Calculate the support of itemset {X} (Support of {X, Z})/(Support of {X})

Implementation overview:

The implementation is built from scratch. It is done in python programming using the basic libraries such as Pandas, NumPy and combinations. The algorithm is divided into segments and these segments are coded into nine functions.

The first function is read_file_and_prepare_data(filename="BestBuy.csv",colname = "Transaction"). This function is used to read the csv file. Once all the transactions are read, this functions finds the unique values of the entire transaction and assigns each unique item to a unique value such as 0,1,2 and so on. All the transactions are converted to a list of numbers. By default the file is BestBuy.csv and colname refers to the column name where we can find the list of products purchased.

The second function is create_1_itemset(X). It creates all the unique items from the transactions. Similarly, The third function create_k_itemset(freq_item, k) is used to create combinations between the item such as 2-itemset, 3-itemset and so on.

The fourth function is create_freq_item(X, ck, min_support) is used to create frequent itemset. It calculates the support for each item or itemset and eliminates the values below the minimum support value provided by the user. The function return a list of items called as "Frequent Itemset".

The fifth function is apriori(X, min_support=0.5). In this function, all the above functions are called in necessary looping format to return the frequent itemset.

The sixth function is create_rules(freq_items, item_support_dict, min_confidence). All the itemset confidence value is calculated and rules are generated.

The seventh function is compute_conf(freq_items, item_support_dict, freq_set, subsets, min_confidence=0.5). The confidence value is computed and if it is below minimum confidence value it is discarded.

The eight function is convert_numbers_to_names(ck) is to convert numbers back which was done in first function to the items name for displaying back to the users.

The ninth function is run(). In this function, The inputs are collected such as filename, support value and confidence value and provide the parameters in function call.

Assumptions:

The implementation has some assumption to make sure the running of the application flawless and without exception.

The assumptions are:

1. Filename with its extension are provided correctly by the user.
2. If the support value and the confidence value are out of range or error in typecasting, then the default value of support value and confidence value are considered.

Requirements:

- Modern Operating System:
- Windows 7, 8 or 10 / Mac OS X 10.11 or higher, 64-bit / Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space
- Python IDE: PyCharm/Spyder with Numpy, Pandas, Itertools installed.

Compile and Run the Source code:

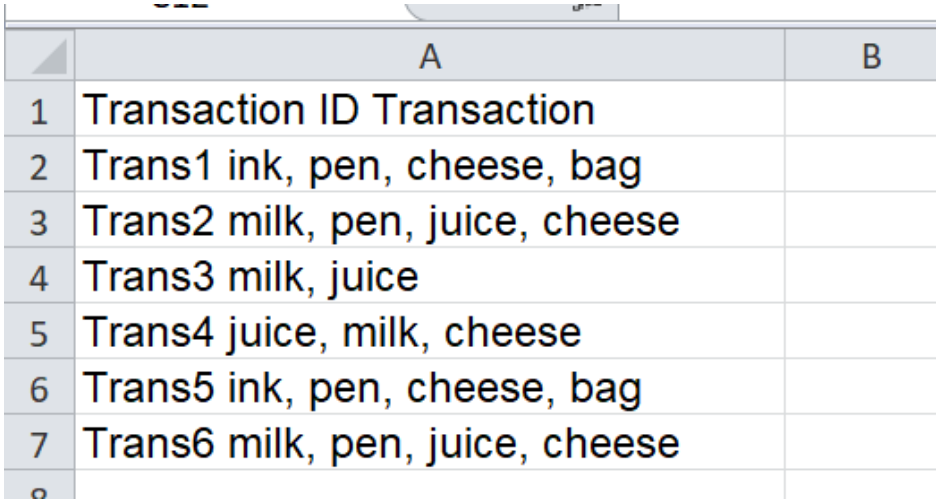
Python automatically compiles the code before running it. The file is saved as “Apriori.py” and all required files are present in the directory. In the terminal, the command **python Apriori.py** in pycharm command prompt and **runfile(“address of the file, wfile=”address of execution file of sypder”)** in spyder will compile and execute the code. All the assumptions are in place as explained in the “Assumptions” Section.

List of Datasets:

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.

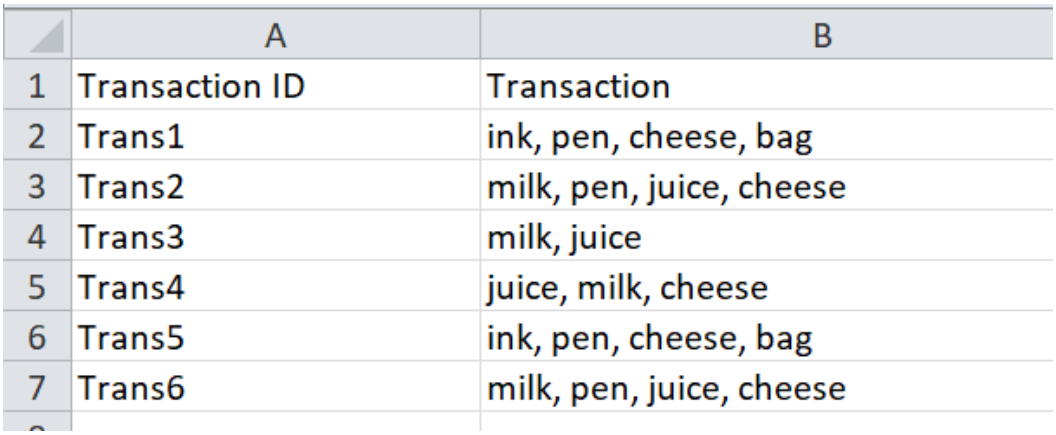
The user can simply paste all the data into a CSV file in below format:



	A	B
1	Transaction ID Transaction	
2	Trans1 ink, pen, cheese, bag	
3	Trans2 milk, pen, juice, cheese	
4	Trans3 milk, juice	
5	Trans4 juice, milk, cheese	
6	Trans5 ink, pen, cheese, bag	
7	Trans6 milk, pen, juice, cheese	

Figure 1: Unformatted csv file

A function file_format(filename) is written to format the file and restore it back. After formatting the file, the same csv file will look like:



	A	B
1	Transaction ID	Transaction
2	Trans1	ink, pen, cheese, bag
3	Trans2	milk, pen, juice, cheese
4	Trans3	milk, juice
5	Trans4	juice, milk, cheese
6	Trans5	ink, pen, cheese, bag
7	Trans6	milk, pen, juice, cheese

Figure 2: Formatted csv file after running the fuction file_format(filename)

Data sets are included with the project package in the folder data. There are 6 data sets in total: Nike, K-Mart, BestBuy, Amazon, Generic, Custom_dataset. They are in CSV format

Testing Implementation and Data Set

Command Prompt:

This section describes several test for each data sets type with different support and confidence values.

It tests each data set respectively.

1. **BestBuy:**

   Running BestBuy data set with support 50% and confidence 40%.

```
In [6]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/BestBuy.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID ', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.5

Enter the CONFIDENCE value : 0.4
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.5
The Minimum Confidence value is :  0.4
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['Anti-Virus'], ['Flash Drive'], ['Microsoft Office'], ['Printer'], ['Speakers'], ['Lab Top'], ['Lab Top Case'], ['Anti-Virus', 'Flash Drive'], ['Flash Drive', 'Microsoft
Office'], ['Printer', 'Flash Drive'], ['Anti-Virus', 'Lab Top'], ['Anti-Virus', 'Lab Top Case'], ['Lab Top', 'Lab Top Case']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

            First_item buys         Second_item  confidence_value
0        [Flash Drive]   ->        [Anti-Virus]          0.769231
1        [Anti-Virus]   ->       [Flash Drive]          0.714286
2   [Microsoft Office]   ->       [Flash Drive]          1.000000
3       [Flash Drive]   ->  [Microsoft Office]          0.846154
4       [Flash Drive]   ->           [Printer]          0.769231
5           [Printer]   ->       [Flash Drive]          1.000000
6           [Lab Top]   ->        [Anti-Virus]          0.833333
7        [Anti-Virus]   ->           [Lab Top]          0.714286
8      [Lab Top Case]   ->        [Anti-Virus]          0.857143
9        [Anti-Virus]   ->      [Lab Top Case]          0.857143
10     [Lab Top Case]   ->           [Lab Top]          0.714286
11          [Lab Top]   ->      [Lab Top Case]          0.833333
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
```

Figure 3: BestBuy Data Set Example Support 50% , confidence 40%.

	A	B	C	D	E
1		First_item	buys	Second_item	confidence_value
2	0	['Flash Drive']	->	['Anti-Virus']	0.769230769
3	1	['Anti-Virus']	->	['Flash Drive']	0.714285714
4	2	['Microsoft Office']	->	['Flash Drive']	1
5	3	['Flash Drive']	->	['Microsoft Office']	0.846153846
6	4	['Flash Drive']	->	['Printer']	0.769230769
7	5	['Printer']	->	['Flash Drive']	1
8	6	['Lab Top']	->	['Anti-Virus']	0.833333333
9	7	['Anti-Virus']	->	['Lab Top']	0.714285714
10	8	['Lab Top Case']	->	['Anti-Virus']	0.857142857
11	9	['Anti-Virus']	->	['Lab Top Case']	0.857142857
12	10	['Lab Top Case']	->	['Lab Top']	0.714285714
13	11	['Lab Top']	->	['Lab Top Case']	0.833333333
14					

Figure 4: The rules are stored in output.csv file

Running BestBuy data set with support 30% and confidence 40%.

```
In [7]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/BestBuy.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID ', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.3

Enter the CONFIDENCE value : 0.4
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.3
The Minimum Confidence value is :  0.4
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['Anti-Virus'], ['Desk Top'], ['Flash Drive'], ['Microsoft Office'], ['Printer'], ['Speakers'], ['Lab Top'], ['Lab Top Case'], ['External Hard-Drive'], ['Digital
Camera'], ['Anti-Virus', 'Flash Drive'], ['Anti-Virus', 'Microsoft Office'], ['Anti-Virus', 'Printer'], ['Anti-Virus', 'Speakers'], ['Flash Drive', 'Microsoft Office'],
['Printer', 'Flash Drive'], ['Speakers', 'Flash Drive'], ['Printer', 'Microsoft Office'], ['Speakers', 'Microsoft Office'], ['Anti-Virus', 'Lab Top'], ['Anti-Virus', 'Lab
Top Case'], ['Flash Drive', 'Lab Top'], ['Flash Drive', 'Lab Top Case'], ['Lab Top Case', 'Microsoft Office'], ['Lab Top', 'Lab Top Case'], ['Anti-Virus', 'External Hard-
Drive'], ['External Hard-Drive', 'Flash Drive'], ['Printer', 'Lab Top Case'], ['External Hard-Drive', 'Lab Top'], ['External Hard-Drive', 'Lab Top Case'], ['Speakers',
'Lab Top Case'], ['Speakers', 'External Hard-Drive'], ['Speakers', 'Digital Camera'], ['Digital Camera', 'Lab Top Case'], ['Anti-Virus', 'Flash Drive', 'Microsoft
Office'], ['Anti-Virus', 'Printer', 'Flash Drive'], ['Anti-Virus', 'Speakers', 'Flash Drive'], ['Anti-Virus', 'Printer', 'Microsoft Office'], ['Anti-Virus', 'Speakers',
'Microsoft Office'], ['Printer', 'Flash Drive', 'Microsoft Office'], ['Speakers', 'Flash Drive', 'Microsoft Office'], ['Anti-Virus', 'Flash Drive', 'Lab Top'], ['Anti-
Virus', 'Flash Drive', 'Lab Top Case'], ['Anti-Virus', 'Lab Top Case', 'Microsoft Office'], ['Flash Drive', 'Lab Top Case', 'Microsoft Office'], ['Anti-Virus', 'Lab Top',
'Lab Top Case'], ['Flash Drive', 'Lab Top', 'Lab Top Case'], ['Anti-Virus', 'External Hard-Drive', 'Flash Drive'], ['Anti-Virus', 'Printer', 'Lab Top Case'], ['Printer',
'Flash Drive', 'Lab Top Case'], ['Anti-Virus', 'External Hard-Drive', 'Lab Top'], ['Anti-Virus', 'External Hard-Drive', 'Lab Top Case'], ['External Hard-Drive', 'Flash
Drive', 'Lab Top Case'], ['Anti-Virus', 'Speakers', 'Lab Top Case'], ['Anti-Virus', 'Speakers', 'External Hard-Drive'], ['Speakers', 'External Hard-Drive', 'Lab Top
Case'], ['Speakers', 'Digital Camera', 'Lab Top Case'], ['Anti-Virus', 'Flash Drive', 'Microsoft Office', 'Printer'], ['Anti-Virus', 'Flash Drive', 'Microsoft Office',
'Speakers'], ['Anti-Virus', 'Flash Drive', 'Lab Top Case', 'Microsoft Office'], ['Anti-Virus', 'Flash Drive', 'Lab Top', 'Lab Top Case'], ['Anti-Virus', 'Flash Drive',
'Lab Top Case', 'Printer'], ['Anti-Virus', 'External Hard-Drive', 'Flash Drive', 'Lab Top Case'], ['Anti-Virus', 'External Hard-Drive', 'Lab Top Case', 'Speakers']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

                         First_item  ... confidence_value
0                      [Flash Drive]  ...         0.769231
1                        [Anti-Virus]  ...         0.714286
2                  [Microsoft Office]  ...         0.727273
3                        [Anti-Virus]  ...         0.571429
4                          [Printer]  ...         0.700000
..                               ...  ...              ...
279  [Anti-Virus, External Hard-Drive]  ...         0.666667
280                        [Speakers]  ...         0.545455
281                    [Lab Top Case]  ...         0.428571
282             [External Hard-Drive]  ...         0.666667
283                      [Anti-Virus]  ...         0.428571

[284 rows x 4 columns]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
```

Figure 5: BestBuy Data Set Example Support 30% , confidence 40%.

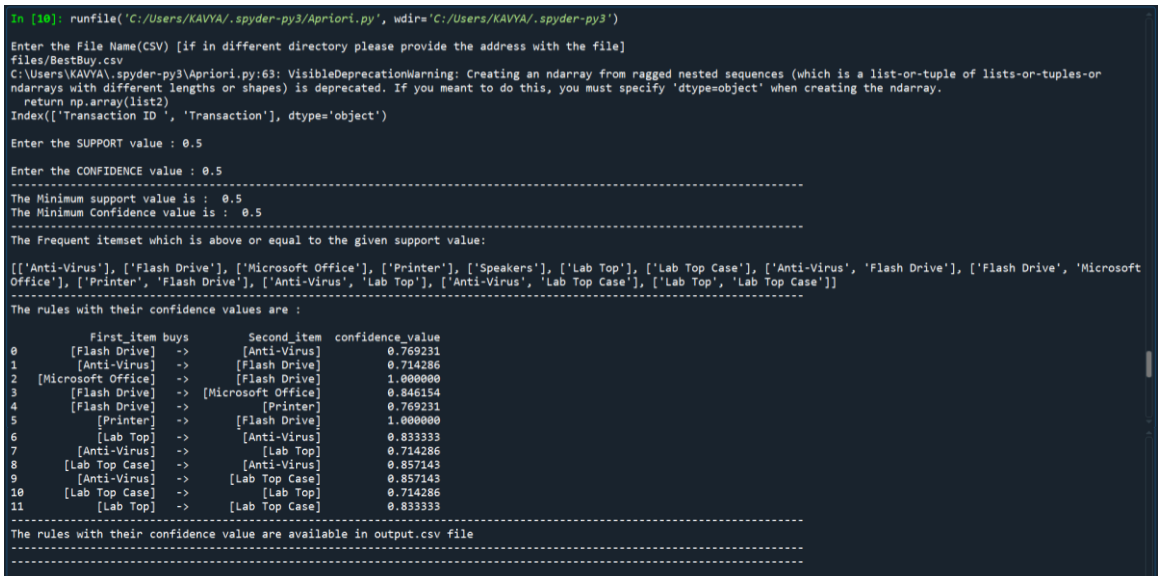
	A	First_item	buys	Second_item	confidence_value
2	0	['Flash Drive']	->	['Anti-Virus']	0.769230769
3	1	['Anti-Virus']	->	['Flash Drive']	0.714285714
4	2	['Microsoft Office']	->	['Anti-Virus']	0.727272727
5	3	['Anti-Virus']	->	['Microsoft Office']	0.571428571
6	4	['Printer']	->	['Anti-Virus']	0.7
7	5	['Anti-Virus']	->	['Printer']	0.5
8	6	['Speakers']	->	['Anti-Virus']	0.818181818
9	7	['Anti-Virus']	->	['Speakers']	0.642857143
10	8	['Microsoft Office']	->	['Flash Drive']	1
11	9	['Flash Drive']	->	['Microsoft Office']	0.846153846
12	10	['Flash Drive']	->	['Printer']	0.769230769
13	11	['Printer']	->	['Flash Drive']	1
14	12	['Flash Drive']	->	['Speakers']	0.461538462
15	13	['Speakers']	->	['Flash Drive']	0.545454545
16	14	['Microsoft Office']	->	['Printer']	0.818181818
17	15	['Printer']	->	['Microsoft Office']	0.9
18	16	['Microsoft Office']	->	['Speakers']	0.545454545
19	17	['Speakers']	->	['Microsoft Office']	0.545454545
20	18	['Lab Top']	->	['Anti-Virus']	0.833333333
21	19	['Anti-Virus']	->	['Lab Top']	0.714285714
22	20	['Lab Top Case']	->	['Anti-Virus']	0.857142857
23	21	['Anti-Virus']	->	['Lab Top Case']	0.857142857
24	22	['Lab Top']	->	['Flash Drive']	0.583333333

output

Ready

Figure 6: The rules formed with confidence value 40% for BestBuy are stored in output.csv

Running BestBuy data set with support 50% and confidence 50%.



```
In [10]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/BestBuy.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID ', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.5

Enter the CONFIDENCE value : 0.5
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.5
The Minimum Confidence value is :  0.5
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['Anti-Virus'], ['Flash Drive'], ['Microsoft Office'], ['Printer'], ['Speakers'], ['Lab Top'], ['Lab Top Case'], ['Anti-Virus', 'Flash Drive'], ['Flash Drive', 'Microsoft
Office'], ['Printer', 'Flash Drive'], ['Anti-Virus', 'Lab Top'], ['Anti-Virus', 'Lab Top Case'], ['Lab Top', 'Lab Top Case']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

           First_item buys         Second_item  confidence_value
0       [Flash Drive]   ->       [Anti-Virus]          0.769231
1        [Anti-Virus]   ->       [Flash Drive]          0.714286
2  [Microsoft Office]   ->       [Flash Drive]          1.000000
3       [Flash Drive]   ->  [Microsoft Office]          0.846154
4       [Flash Drive]   ->           [Printer]          0.769231
5           [Printer]   ->       [Flash Drive]          1.000000
6           [Lab Top]   ->       [Anti-Virus]          0.833333
7        [Anti-Virus]   ->          [Lab Top]          0.714286
8      [Lab Top Case]   ->       [Anti-Virus]          0.857143
9        [Anti-Virus]   ->     [Lab Top Case]          0.857143
10     [Lab Top Case]   ->          [Lab Top]          0.714286
11          [Lab Top]   ->     [Lab Top Case]          0.833333
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
```

Figure 7: BestBuy Data Set Example Support 50% , confidence 50%.

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['Flash Drive']	->	['Anti-Virus']	0.769230769	
3	1	['Anti-Virus']	->	['Flash Drive']	0.714285714	
4	2	['Microsoft Office']	->	['Flash Drive']	1	
5	3	['Flash Drive']	->	['Microsoft Office']	0.846153846	
6	4	['Flash Drive']	->	['Printer']	0.769230769	
7	5	['Printer']	->	['Flash Drive']	1	
8	6	['Lab Top']	->	['Anti-Virus']	0.833333333	
9	7	['Anti-Virus']	->	['Lab Top']	0.714285714	
10	8	['Lab Top Case']	->	['Anti-Virus']	0.857142857	
11	9	['Anti-Virus']	->	['Lab Top Case']	0.857142857	
12	10	['Lab Top Case']	->	['Lab Top']	0.714285714	
13	11	['Lab Top']	->	['Lab Top Case']	0.833333333	
14						

Figure 8: The rules formed with confidence value 50% for BestBuy are stored in output.csv

2. Amazon:

Running Amazon data set with support 30% and confidence 50%.

```
In [11]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Amazon.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID  ', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.3

Enter the CONFIDENCE value : 0.5
-----------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.3
The Minimum Confidence value is :  0.5
-----------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['A Beginner's Guide'], ['Android Programming: The Big Nerd Ranch'], ['Java For Dummies'], ['Java: The Complete Reference'], ['Head First Java 2nd Edition'], ['Beginning
Programming with Java'], ['A Beginner's Guide', 'Android Programming: The Big Nerd Ranch'], ['Java For Dummies', 'A Beginner's Guide'], ['A Beginner's Guide', 'Java:
The Complete Reference'], ['Java For Dummies', 'Android Programming: The Big Nerd Ranch'], ['Java: The Complete Reference', 'Android Programming: The Big Nerd Ranch'],
['Java For Dummies', 'Java: The Complete Reference'], ['Android Programming: The Big Nerd Ranch', 'Head First Java 2nd Edition'], ['Java For Dummies', 'A Beginner's
Guide', 'Java: The Complete Reference'], ['Java For Dummies', 'Java: The Complete Reference', 'Android Programming: The Big Nerd Ranch']]
-----------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

                                  First_item  ... confidence_value
0                              [A Beginner's Guide]  ...         0.545455
1                              [A Beginner's Guide]  ...         0.818182
2                                [Java For Dummies]  ...         0.692308
3                    [Java: The Complete Reference]  ...         0.900000
4                              [A Beginner's Guide]  ...         0.818182
5           [Android Programming: The Big Nerd Ranch]  ...         0.692308
6                                [Java For Dummies]  ...         0.692308
7                    [Java: The Complete Reference]  ...         0.600000
8                    [Java: The Complete Reference]  ...         1.000000
9                                [Java For Dummies]  ...         0.769231
10                    [Head First Java 2nd Edition]  ...         0.750000
11  [A Beginner's Guide, Java: The Complete Refere...  ...         1.000000
12   [Java For Dummies, Java: The Complete Reference]  ...         0.900000
13            [Java For Dummies, A Beginner's Guide]  ...         1.000000
14                   [Java: The Complete Reference]  ...         0.900000
15                             [A Beginner's Guide]  ...         0.818182
16                               [Java For Dummies]  ...         0.692308
17  [Java: The Complete Reference, Android Program...  ...         1.000000
18  [Java For Dummies, Android Programming: The Bi...  ...         0.666667
19   [Java For Dummies, Java: The Complete Reference]  ...         0.600000
20                   [Java: The Complete Reference]  ...         0.600000

[21 rows x 4 columns]
-----------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
```

Figure 9: Amazon Data Set Example Support 30% , confidence 50%.

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['A Beginnerï¿½s Guide']	->	['Android Programming: The Big Nerd Ranch']	0.545455	
3	1	['A Beginnerï¿½s Guide']	->	['Java For Dummies']	0.818182	
4	2	['Java For Dummies']	->	['A Beginnerï¿½s Guide']	0.692308	
5	3	['Java: The Complete Reference']	->	['A Beginnerï¿½s Guide']	0.9	
6	4	['A Beginnerï¿½s Guide']	->	['Java: The Complete Reference']	0.818182	
7	5	['Android Programming: The Big Nerd Ranch']	->	['Java For Dummies']	0.692308	
8	6	['Java For Dummies']	->	['Android Programming: The Big Nerd Ranch']	0.692308	
9	7	['Java: The Complete Reference']	->	['Android Programming: The Big Nerd Ranch']	0.6	
10	8	['Java: The Complete Reference']	->	['Java For Dummies']	1	
11	9	['Java For Dummies']	->	['Java: The Complete Reference']	0.769231	
12	10	['Head First Java 2nd Edition']	->	['Android Programming: The Big Nerd Ranch']	0.75	
13	11	['A Beginnerï¿½s Guide', 'Java: The Complete Refere	->	['Java For Dummies']	1	
14	12	['Java For Dummies', 'Java: The Complete Reference	->	['A Beginnerï¿½s Guide']	0.9	
15	13	['Java For Dummies', 'A Beginnerï¿½s Guide']	->	['Java: The Complete Reference']	1	
16	14	['Java: The Complete Reference']	->	['Java For Dummies', 'A Beginnerï¿½s Guide']	0.9	
17	15	['A Beginnerï¿½s Guide']	->	['Java For Dummies', 'Java: The Complete Reference']	0.818182	
18	16	['Java For Dummies']	->	['A Beginnerï¿½s Guide', 'Java: The Complete Reference'	0.692308	
19	17	['Java: The Complete Reference', 'Android Program	->	['Java For Dummies']	1	
20	18	['Java For Dummies', 'Android Programming: The Bi	->	['Java: The Complete Reference']	0.666667	
21	19	['Java For Dummies', 'Java: The Complete Reference	->	['Android Programming: The Big Nerd Ranch']	0.6	
22	20	['Java: The Complete Reference']	->	['Java For Dummies', 'Android Programming: The Big Ne	0.6	
23						

Figure 10: The rules formed with confidence value 50% for Amazon are stored in output.csv

Running Amazon data set with support 50% and confidence 60%.

```
In [15]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Amazon.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID  ', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.5

Enter the CONFIDENCE value : 0.6
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.5
The Minimum Confidence value is :  0.6
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['A Beginner's Guide'], ['Android Programming: The Big Nerd Ranch'], ['Java For Dummies'], ['Java: The Complete Reference'], ['Java For Dummies', 'Java: The Complete
Reference']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

                    First_item  ... confidence_value
0  [Java: The Complete Reference]  ...        1.000000
1             [Java For Dummies]  ...        0.769231

[2 rows x 4 columns]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
```

Figure 11: Amazon Data Set Example Support 50% , confidence 60%.

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['Java: The Complete Reference']	->	['Java For Dummies']	1	
3	1	['Java For Dummies']	->	['Java: The Complete Reference']	0.769231	
4						

Figure 12: The rules formed with confidence value 60% for Amazon are stored in output.csv

Running Amazon data set with support 35% and confidence 40%.

```
In [14]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Amazon.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID  ', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.35

Enter the CONFIDENCE value : 0.40
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.35
The Minimum Confidence value is :  0.4
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['A Beginner's Guide'], ['Android Programming: The Big Nerd Ranch'], ['Java For Dummies'], ['Java: The Complete Reference'], ['Head First Java 2nd Edition'], ['Java For
Dummies', 'A Beginner's Guide'], ['A Beginner's Guide', 'Java: The Complete Reference'], ['Java For Dummies', 'Android Programming: The Big Nerd Ranch'], ['Java For
Dummies', 'Java: The Complete Reference'], ['Java For Dummies', 'A Beginner's Guide', 'Java: The Complete Reference']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

                                        First_item  ... confidence_value
0                                [A Beginner's Guide]  ...         0.818182
1                                  [Java For Dummies]  ...         0.692308
2                    [Java: The Complete Reference]  ...         0.900000
3                                [A Beginner's Guide]  ...         0.818182
4          [Android Programming: The Big Nerd Ranch]  ...         0.692308
5                                  [Java For Dummies]  ...         0.692308
6                    [Java: The Complete Reference]  ...         1.000000
7                                  [Java For Dummies]  ...         0.769231
8   [A Beginner's Guide, Java: The Complete Refere...  ...         1.000000
9    [Java For Dummies, Java: The Complete Reference]  ...         0.900000
10            [Java For Dummies, A Beginner's Guide]  ...         1.000000
11                   [Java: The Complete Reference]  ...         0.900000
12                               [A Beginner's Guide]  ...         0.818182
13                                 [Java For Dummies]  ...         0.692308

[14 rows x 4 columns]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------------------
```

Figure 13: Amazon Data Set Example Support 35% , confidence 40%.

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['A Beginnerï¿½s Guide']	->	['Java For Dummies']	0.818182	
3	1	['Java For Dummies']	->	['A Beginnerï¿½s Guide']	0.692308	
4	2	['Java: The Complete Reference']	->	['A Beginnerï¿½s Guide']	0.9	
5	3	['A Beginnerï¿½s Guide']	->	['Java: The Complete Reference']	0.818182	
6	4	['Android Programming: The Big Nerd Ranch']	->	['Java For Dummies']	0.692308	
7	5	['Java For Dummies']	->	['Android Programming: The Big Nerd Ranch']	0.692308	
8	6	['Java: The Complete Reference']	->	['Java For Dummies']	1	
9	7	['Java For Dummies']	->	['Java: The Complete Reference']	0.769231	
10	8	['A Beginnerï¿½s Guide', 'Java: The Complete Reference']	->	['Java For Dummies']	1	
11	9	['Java For Dummies', 'Java: The Complete Reference']	->	['A Beginnerï¿½s Guide']	0.9	
12	10	['Java For Dummies', 'A Beginnerï¿½s Guide']	->	['Java: The Complete Reference']	1	
13	11	['Java: The Complete Reference']	->	['Java For Dummies', 'A Beginnerï¿½s Guide']	0.9	
14	12	['A Beginnerï¿½s Guide']	->	['Java For Dummies', 'Java: The Complete Reference']	0.818182	
15	13	['Java For Dummies']	->	['A Beginnerï¿½s Guide', 'Java: The Complete Reference']	0.692308	

Figure 14: The rules formed with confidence value 40% for Amazon are stored in output.csv

3. K-Mart:

Running Kmart data set with support 50% and confidence 50%.

```
In [2]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Kmart.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple
of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when
creating the ndarray.
  return np.array(list2)
Index(['Unnamed: 0', 'Transaction ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.5

Enter the CONFIDENCE value : 0.5
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.5
The Minimum Confidence value is :  0.5
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['Decorative Pillows'], ['Bed Skirts'], ['Kids Bedding'], ['Shams'], ['Sheets'], ['Bed Skirts', 'Kids Bedding'], ['Sheets', 'Kids Bedding']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

      First_item buys     Second_item  confidence_value
0  [Kids Bedding]   ->    [Bed Skirts]          0.833333
1    [Bed Skirts]   ->  [Kids Bedding]          0.909091
2  [Kids Bedding]   ->        [Sheets]          0.833333
3        [Sheets]   ->  [Kids Bedding]          1.000000
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
```

Figure 15: Kmart Data Set Example Support 50% , confidence 50%.

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['Kids Bedding']	->	['Bed Skirts']	0.833333	
3	1	['Bed Skirts']	->	['Kids Bedding']	0.909091	
4	2	['Kids Bedding']	->	['Sheets']	0.833333	
5	3	['Sheets']	->	['Kids Bedding']	1	
6						
7						

Figure 16: The rules formed with confidence value 50% for K-Mart are stored in output.csv

Running Kmart data set with support 50% and confidence 50%.

```
In [4]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Kmart.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Unnamed: 0', 'Transaction ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.4

Enter the CONFIDENCE value : 0.6
---------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.4
The Minimum Confidence value is :  0.6
---------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['Decorative Pillows'], ['Quilts'], ['Bed Skirts'], ['Kids Bedding'], ['Shams'], ['Sheets'], ['Bed Skirts', 'Kids Bedding'], ['Bed Skirts', 'Shams'], ['Bed Skirts',
'Sheets'], ['Kids Bedding', 'Shams'], ['Sheets', 'Kids Bedding'], ['Bed Skirts', 'Kids Bedding', 'Shams'], ['Bed Skirts', 'Sheets', 'Kids Bedding']]
---------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

                First_item  ... confidence_value
0             [Kids Bedding]  ...         0.833333
1               [Bed Skirts]  ...         0.909091
2                    [Shams]  ...         0.818182
3               [Bed Skirts]  ...         0.818182
4                   [Sheets]  ...         0.900000
5               [Bed Skirts]  ...         0.818182
6                    [Shams]  ...         0.818182
7             [Kids Bedding]  ...         0.750000
8             [Kids Bedding]  ...         0.833333
9                   [Sheets]  ...         1.000000
10      [Kids Bedding, Shams]  ...         0.888889
11        [Bed Skirts, Shams]  ...         0.888889
12  [Bed Skirts, Kids Bedding]  ...         0.800000
13                   [Shams]  ...         0.727273
14            [Kids Bedding]  ...         0.666667
15              [Bed Skirts]  ...         0.727273
16    [Sheets, Kids Bedding]  ...         0.900000
17  [Bed Skirts, Kids Bedding]  ...         0.900000
18       [Bed Skirts, Sheets]  ...         1.000000
19            [Kids Bedding]  ...         0.750000
20                  [Sheets]  ...         0.900000
21              [Bed Skirts]  ...         0.818182

[22 rows x 4 columns]
---------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
---------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------
```

Figure 17: Kmart Data Set Example Support 40% , confidence 60%.

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['Kids Bedding']	->	['Bed Skirts']	0.833333	
3	1	['Bed Skirts']	->	['Kids Bedding']	0.909091	
4	2	['Shams']	->	['Bed Skirts']	0.818182	
5	3	['Bed Skirts']	->	['Shams']	0.818182	
6	4	['Sheets']	->	['Bed Skirts']	0.9	
7	5	['Bed Skirts']	->	['Sheets']	0.818182	
8	6	['Shams']	->	['Kids Bedding']	0.818182	
9	7	['Kids Bedding']	->	['Shams']	0.75	
10	8	['Kids Bedding']	->	['Sheets']	0.833333	
11	9	['Sheets']	->	['Kids Bedding']	1	
12	10	['Kids Bedding', 'Shams']	->	['Bed Skirts']	0.888889	
13	11	['Bed Skirts', 'Shams']	->	['Kids Bedding']	0.888889	
14	12	['Bed Skirts', 'Kids Bedding']	->	['Shams']	0.8	
15	13	['Shams']	->	['Bed Skirts', 'Kids Bedding']	0.727273	
16	14	['Kids Bedding']	->	['Bed Skirts', 'Shams']	0.666667	
17	15	['Bed Skirts']	->	['Kids Bedding', 'Shams']	0.727273	
18	16	['Sheets', 'Kids Bedding']	->	['Bed Skirts']	0.9	
19	17	['Bed Skirts', 'Kids Bedding']	->	['Sheets']	0.9	
20	18	['Bed Skirts', 'Sheets']	->	['Kids Bedding']	1	
21	19	['Kids Bedding']	->	['Bed Skirts', 'Sheets']	0.75	
22	20	['Sheets']	->	['Bed Skirts', 'Kids Bedding']	0.9	
23	21	['Bed Skirts']	->	['Sheets', 'Kids Bedding']	0.818182	
24						

output

Figure 18: The rules formed with confidence value 60% for K-Mart are stored in output.csv

4. Nike:

Running Nike data set with support 50% and confidence 60%.

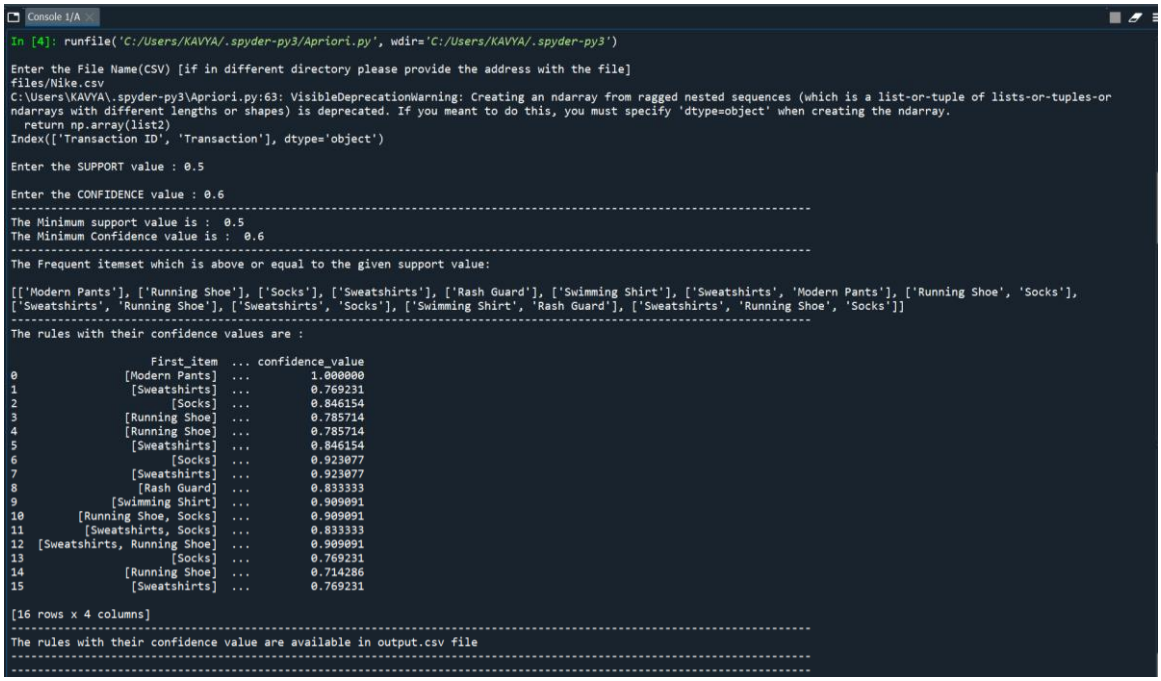


Figure 19: Nike Data Set Example Support 50% , confidence 60%

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['Modern Pants']	->	['Sweatshirts']	1	
3	1	['Sweatshirts']	->	['Modern Pants']	0.769231	
4	2	['Socks']	->	['Running Shoe']	0.846154	
5	3	['Running Shoe']	->	['Socks']	0.785714	
6	4	['Running Shoe']	->	['Sweatshirts']	0.785714	
7	5	['Sweatshirts']	->	['Running Shoe']	0.846154	
8	6	['Socks']	->	['Sweatshirts']	0.923077	
9	7	['Sweatshirts']	->	['Socks']	0.923077	
10	8	['Rash Guard']	->	['Swimming Shirt']	0.833333	
11	9	['Swimming Shirt']	->	['Rash Guard']	0.909091	
12	10	['Running Shoe', 'Socks']	->	['Sweatshirts']	0.909091	
13	11	['Sweatshirts', 'Socks']	->	['Running Shoe']	0.833333	
14	12	['Sweatshirts', 'Running Shoe']	->	['Socks']	0.909091	
15	13	['Socks']	->	['Sweatshirts', 'Running Shoe']	0.769231	
16	14	['Running Shoe']	->	['Sweatshirts', 'Socks']	0.714286	
17	15	['Sweatshirts']	->	['Running Shoe', 'Socks']	0.769231	
18						

Figure 20: The rules formed with confidence value 60% for Nike are stored in output.csv

Running Nike data set with support 60% and confidence 70%.

```
In [6]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Nike.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.6

Enter the CONFIDENCE value : 0.7
----------------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.6
The Minimum Confidence value is :  0.7
----------------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['Running Shoe'], ['Socks'], ['Sweatshirts'], ['Rash Guard'], ['Sweatshirts', 'Socks']]
----------------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

     First_item buys    Second_item  confidence_value
0       [Socks]   ->  [Sweatshirts]          0.923077
1  [Sweatshirts]   ->       [Socks]          0.923077
----------------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
```

Figure 21: Nike Data Set Example Support 60% , confidence 70%

G16

	A	B	C	D	E	F
1		First_item	buys	Second_item	confidence_value	
2	0	['Socks']	->	['Sweatshirts']	0.923077	
3	1	['Sweatshirts']	->	['Socks']	0.923077	
4						
5						
6						
7						

Figure 22: The rules formed with confidence value 70% for Nike are stored in output.csv

Running Nike data set with support 25% and confidence 35%.

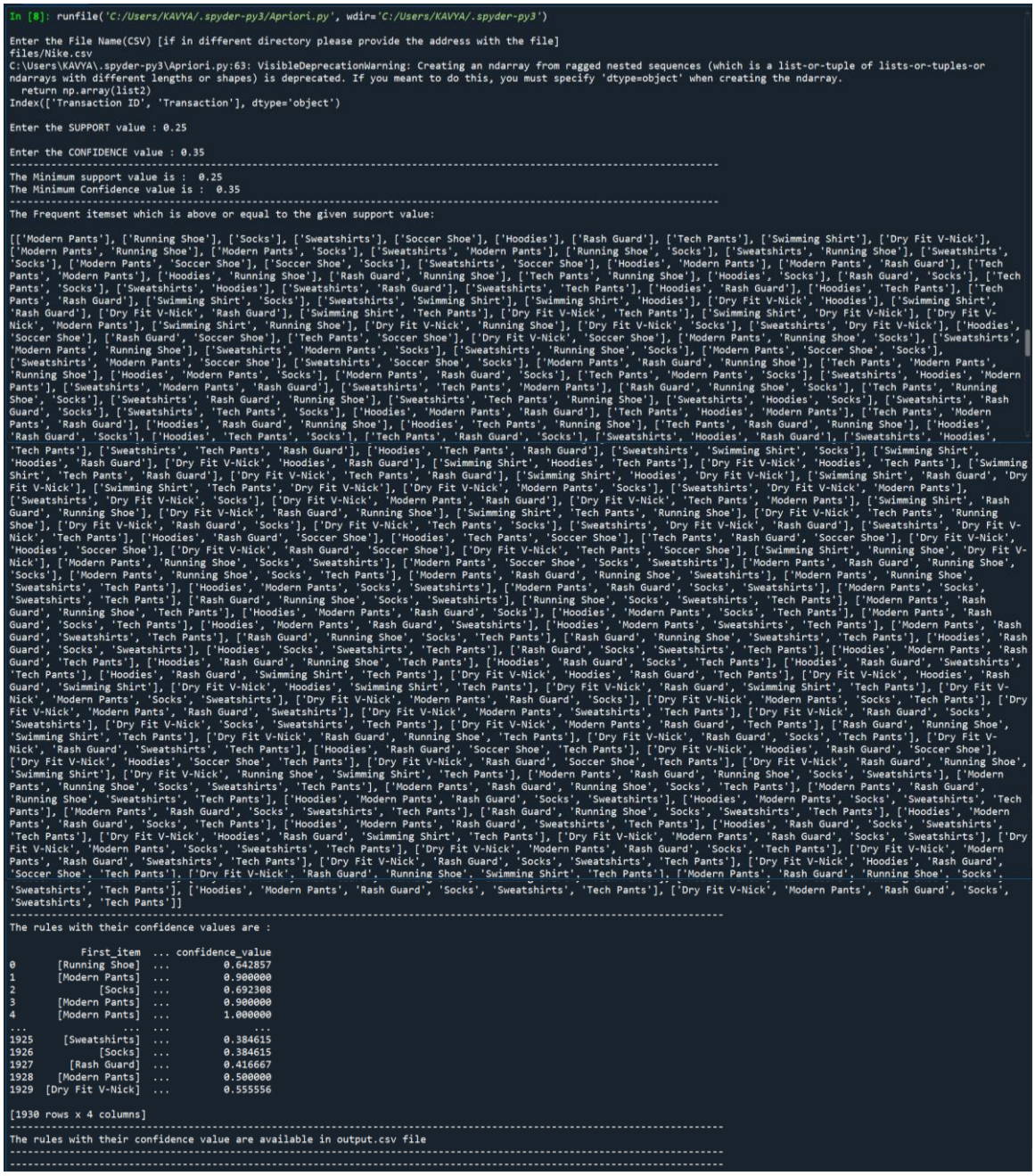


Figure 23: Nike Data Set Example Support 25% , confidence 35%

	A	B	C	D	E	F	G	H
1		First_item	buys	Second_item	confidence_value			
2	0	['Running	->	['Modern Pants']	0.642857			
3	1	['Modern	->	['Running Shoe']	0.9			
4	2	['Socks']	->	['Modern Pants']	0.692308			
5	3	['Modern	->	['Socks']	0.9			
6	4	['Modern	->	['Sweatshirts']	1			
7	5	['Sweatshi	->	['Modern Pants']	0.769231			
8	6	['Socks']	->	['Running Shoe']	0.846154			
9	7	['Running	->	['Socks']	0.785714			
10	8	['Running	->	['Sweatshirts']	0.785714			
11	9	['Sweatshi	->	['Running Shoe']	0.846154			
12	10	['Socks']	->	['Sweatshirts']	0.923077			
13	11	['Sweatshi	->	['Socks']	0.923077			
14	12	['Soccer Sh	->	['Modern Pants']	0.833333			
15	13	['Modern	->	['Soccer Shoe']	0.5			
16	14	['Socks']	->	['Soccer Shoe']	0.384615			
17	15	['Soccer Sh	->	['Socks']	0.833333			
18	16	['Soccer Sh	->	['Sweatshirts']	0.833333			
19	17	['Sweatshi	->	['Soccer Shoe']	0.384615			
20	18	['Modern	->	['Hoodies']	0.5			
21	19	['Hoodies'	->	['Modern Pants']	0.625			
22	20	['Rash Gua	->	['Modern Pants']	0.5			
23	21	['Modern	->	['Rash Guard']	0.6			
24	22	['Modern	->	['Tech Pants']	0.6			

output

Figure 24: The rules formed with confidence value 35% for Nike are stored in output.csv

5. Generic:

Running Generic data set with support 50% and confidence 60%.

```
In [10]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Generic.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.5

Enter the CONFIDENCE value : 0.6
----------------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.5
The Minimum Confidence value is :  0.6
----------------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['A'], ['C'], ['E'], ['A', 'C'], ['A', 'E']]
----------------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

  First_item buys Second_item  confidence_value
0        [C]   ->         [A]          1.000000
1        [A]   ->         [C]          0.636364
2        [E]   ->         [A]          1.000000
3        [A]   ->         [E]          0.727273
----------------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
----------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------
```

Figure 23: Generic Data Set Example Support 50% , confidence 60%

	A	B	C	D	E	F	G
1		First_item	buys	Second_it	confidence_value		
2	0	['C']	->	['A']	1		
3	1	['A']	->	['C']	0.636364		
4	2	['E']	->	['A']	1		
5	3	['A']	->	['E']	0.727273		
6							
7							
8							

Figure 24: The rules formed with confidence value 60% for Generic are stored in output.csv

Running Generic data set with support 30% and confidence 35%.

```
In [16]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/Generic.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.3

Enter the CONFIDENCE value : 0.35
-----------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.3
The Minimum Confidence value is :  0.35
-----------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['A'], ['B'], ['C'], ['D'], ['E'], ['A', 'B'], ['A', 'C'], ['B', 'C'], ['A', 'D'], ['A', 'E'], ['C', 'E'], ['A', 'B', 'C'], ['A', 'C', 'E']]
-----------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

   First_item buys Second_item  confidence_value
0         [B]   ->         [A]          1.000000
1         [A]   ->         [B]          0.454545
2         [C]   ->         [A]          1.000000
3         [A]   ->         [C]          0.636364
4         [C]   ->         [B]          0.571429
5         [B]   ->         [C]          0.800000
6         [D]   ->         [A]          1.000000
7         [A]   ->         [D]          0.363636
8         [E]   ->         [A]          1.000000
9         [A]   ->         [E]          0.727273
10        [E]   ->         [C]          0.500000
11        [C]   ->         [E]          0.571429
12     [B, C]   ->         [A]          1.000000
13     [A, C]   ->         [B]          0.571429
14     [A, B]   ->         [C]          0.800000
15        [C]   ->      [A, B]          0.571429
16        [B]   ->      [A, C]          0.800000
17        [A]   ->      [B, C]          0.363636
18     [C, E]   ->         [A]          1.000000
19     [A, E]   ->         [C]          0.500000
20     [A, C]   ->         [E]          0.571429
21        [E]   ->      [A, C]          0.500000
22        [C]   ->      [A, E]          0.571429
23        [A]   ->      [C, E]          0.363636
-----------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
```

Figure 25: Generic Data Set Example Support 30% , confidence 35%

	A	B	C	D	E	F	G
1		First_item	buys	Second_it	confidence_value		
2	0	['B']	->	['A']	1		
3	1	['A']	->	['B']	0.454545		
4	2	['C']	->	['A']	1		
5	3	['A']	->	['C']	0.636364		
6	4	['C']	->	['B']	0.571429		
7	5	['B']	->	['C']	0.8		
8	6	['D']	->	['A']	1		
9	7	['A']	->	['D']	0.363636		
10	8	['E']	->	['A']	1		
11	9	['A']	->	['E']	0.727273		
12	10	['E']	->	['C']	0.5		
13	11	['C']	->	['E']	0.571429		
14	12	['B', 'C']	->	['A']	1		
15	13	['A', 'C']	->	['B']	0.571429		
16	14	['A', 'B']	->	['C']	0.8		
17	15	['C']	->	['A', 'B']	0.571429		
18	16	['B']	->	['A', 'C']	0.8		
19	17	['A']	->	['B', 'C']	0.363636		
20	18	['C', 'E']	->	['A']	1		
21	19	['A', 'E']	->	['C']	0.5		
22	20	['A', 'C']	->	['E']	0.571429		
23	21	['E']	->	['A', 'C']	0.5		
24	22	['C']	->	['A', 'E']	0.571429		

output

Figure 26: The rules formed with confidence value 35% for Generic are stored in output.csv

6. Custom Dataset:

Running Custom data set with support 50% and confidence 60%.

```
In [17]: runfile('C:/Users/KAVYA/.spyder-py3/Apriori.py', wdir='C:/Users/KAVYA/.spyder-py3')

Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/custom_dataset.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.5

Enter the CONFIDENCE value : 0.6
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.5
The Minimum Confidence value is :  0.6
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['cheese'], ['pen'], ['juice'], ['milk'], ['cheese', 'pen'], ['cheese', 'juice'], ['cheese', 'milk'], ['juice', 'milk'], ['cheese', 'juice', 'milk']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

         First_item buys      Second_item  confidence_value
0             [pen]   ->         [cheese]              1.00
1          [cheese]   ->            [pen]              0.80
2           [juice]   ->         [cheese]              0.75
3          [cheese]   ->          [juice]              0.60
4            [milk]   ->         [cheese]              0.75
5          [cheese]   ->           [milk]              0.60
6            [milk]   ->          [juice]              1.00
7           [juice]   ->           [milk]              1.00
8     [juice, milk]   ->         [cheese]              0.75
9    [cheese, milk]   ->          [juice]              1.00
10 [cheese, juice]   ->           [milk]              1.00
11           [milk]   ->  [cheese, juice]              0.75
12          [juice]   ->   [cheese, milk]              0.75
13         [cheese]   ->    [juice, milk]              0.60
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
```

Figure 27: Custom Data Set Example Support 50% , confidence 60%

	A	B	C	D	E	F	G
1		First_item	buys	Second_item	confidence_value		
2	0	['pen']	->	['cheese']	1		
3	1	['cheese']	->	['pen']	0.8		
4	2	['juice']	->	['cheese']	0.75		
5	3	['cheese']	->	['juice']	0.6		
6	4	['milk']	->	['cheese']	0.75		
7	5	['cheese']	->	['milk']	0.6		
8	6	['milk']	->	['juice']	1		
9	7	['juice']	->	['milk']	1		
10	8	['juice', 'm	->	['cheese']	0.75		
11	9	['cheese', '	->	['juice']	1		
12	10	['cheese', '	->	['milk']	1		
13	11	['milk']	->	['cheese', 'juice']	0.75		
14	12	['juice']	->	['cheese', 'milk']	0.75		
15	13	['cheese']	->	['juice', 'milk']	0.6		

Figure 28: The rules formed with confidence value 35% for Custom are stored in output.csv

Running Custom data set with support 30% and confidence 45%.

```
Enter the File Name(CSV) [if in different directory please provide the address with the file]
files/custom_dataset.csv
C:\Users\KAVYA\.spyder-py3\Apriori.py:63: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  return np.array(list2)
Index(['Transaction ID', 'Transaction'], dtype='object')

Enter the SUPPORT value : 0.3

Enter the CONFIDENCE value : 0.45
-----------------------------------------------------------------------------------------------------------------------
The Minimum support value is :  0.3
The Minimum Confidence value is :  0.45
-----------------------------------------------------------------------------------------------------------------------
The Frequent itemset which is above or equal to the given support value:

[['bag'], ['cheese'], ['ink'], ['pen'], ['juice'], ['milk'], ['bag', 'cheese'], ['bag', 'ink'], ['bag', 'pen'], ['cheese', 'ink'], ['cheese', 'pen'], ['ink', 'pen'],
['cheese', 'juice'], ['cheese', 'milk'], ['juice', 'pen'], ['milk', 'pen'], ['juice', 'milk'], ['bag', 'cheese', 'ink'], ['bag', 'cheese', 'pen'], ['bag', 'ink', 'pen'],
['cheese', 'ink', 'pen'], ['cheese', 'juice', 'pen'], ['cheese', 'milk', 'pen'], ['cheese', 'juice', 'milk'], ['juice', 'milk', 'pen'], ['bag', 'cheese', 'ink', 'pen'],
['cheese', 'juice', 'milk', 'pen']]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence values are :

         First_item buys            Second_item  confidence_value
0             [bag]   ->               [cheese]          1.000000
1             [ink]   ->                  [bag]          1.000000
2             [bag]   ->                  [ink]          1.000000
3             [pen]   ->                  [bag]          0.500000
4             [bag]   ->                  [pen]          1.000000
..              ...  ...                    ...               ...
84   [cheese, milk]   ->           [juice, pen]          0.666667
85  [cheese, juice]   ->            [milk, pen]          0.666667
86            [pen]   ->  [cheese, juice, milk]          0.500000
87           [milk]   ->   [cheese, juice, pen]          0.500000
88         [juice]   ->    [cheese, milk, pen]          0.500000

[89 rows x 4 columns]
-----------------------------------------------------------------------------------------------------------------------
The rules with their confidence value are available in output.csv file
-----------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------
```

Figure 27: Custom Data Set Example Support 30% , confidence 45%

	A	B	C	D	E	F	G
1		First_item	buys	Second_it	confidence_value		
2	0	['bag']	->	['cheese']	1		
3	1	['ink']	->	['bag']	1		
4	2	['bag']	->	['ink']	1		
5	3	['pen']	->	['bag']	0.5		
6	4	['bag']	->	['pen']	1		
7	5	['ink']	->	['cheese']	1		
8	6	['pen']	->	['cheese']	1		
9	7	['cheese']	->	['pen']	0.8		
10	8	['pen']	->	['ink']	0.5		
11	9	['ink']	->	['pen']	1		
12	10	['juice']	->	['cheese']	0.75		
13	11	['cheese']	->	['juice']	0.6		
14	12	['milk']	->	['cheese']	0.75		
15	13	['cheese']	->	['milk']	0.6		
16	14	['pen']	->	['juice']	0.5		
17	15	['juice']	->	['pen']	0.5		
18	16	['pen']	->	['milk']	0.5		
19	17	['milk']	->	['pen']	0.5		
20	18	['milk']	->	['juice']	1		
21	19	['juice']	->	['milk']	1		
22	20	['cheese', 'ink']	->	['bag']	1		
23	21	['bag', 'ink']	->	['cheese']	1		
24	22	['bag', 'cheese']	->	['ink']	1		

output

Figure 28: The rules formed with confidence value 45% for Custom are stored in output.csv