Project Statement

1. Create 10 items usually seen in Amazon, K-mart, or any other supermarkets (e.g. diapers, clothes, etc.).

2. Create a database of 20 transactions each containing some of these items. The information can be stored in a file, or a DBMS (e.g. NJIT ORACLE or MySQL).

3. Repeat (1) by creating 4 additional, different databases each containing 20 transactions.

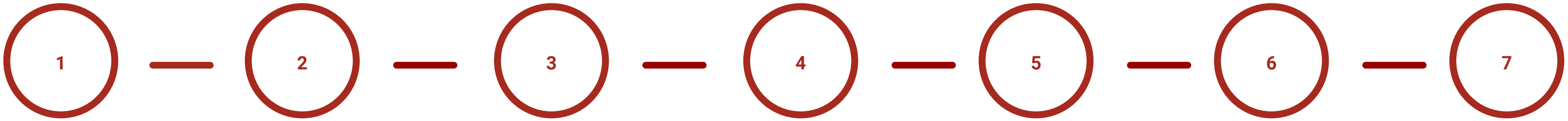
Using the Apriori algorithm, generate and print out all the association rules and the input transactions for each of the 5 transactional databases you created. The support and confidence *must* be user-specified parameters, so the output should show different rules with respect to different databases and different support/confidence.

Make sure to show multiple support and confidence results.

Project Description

1. There are 5 .xlsx files have been created with the data available in the Midterm_Project_Items_Datasets_Examples.pdf file in Canvas
2. All the files are saved in the **input Folder** inside the Project Folder
3. The complete generation of Association Rules consists of 7 functions excluding the main()
4. The Process Flow is describe as follows:

Process Flow



1. Get Input Parameters & Read Data

2. Create All Possible Combinations And Calculate Support for all single Items and combinations. Keep On Filtering Based on the Minimum Support and until all the combinations are covered

3. Generate Sets of Low Frequency Items and Combinations and High Frequency Items and Combinations

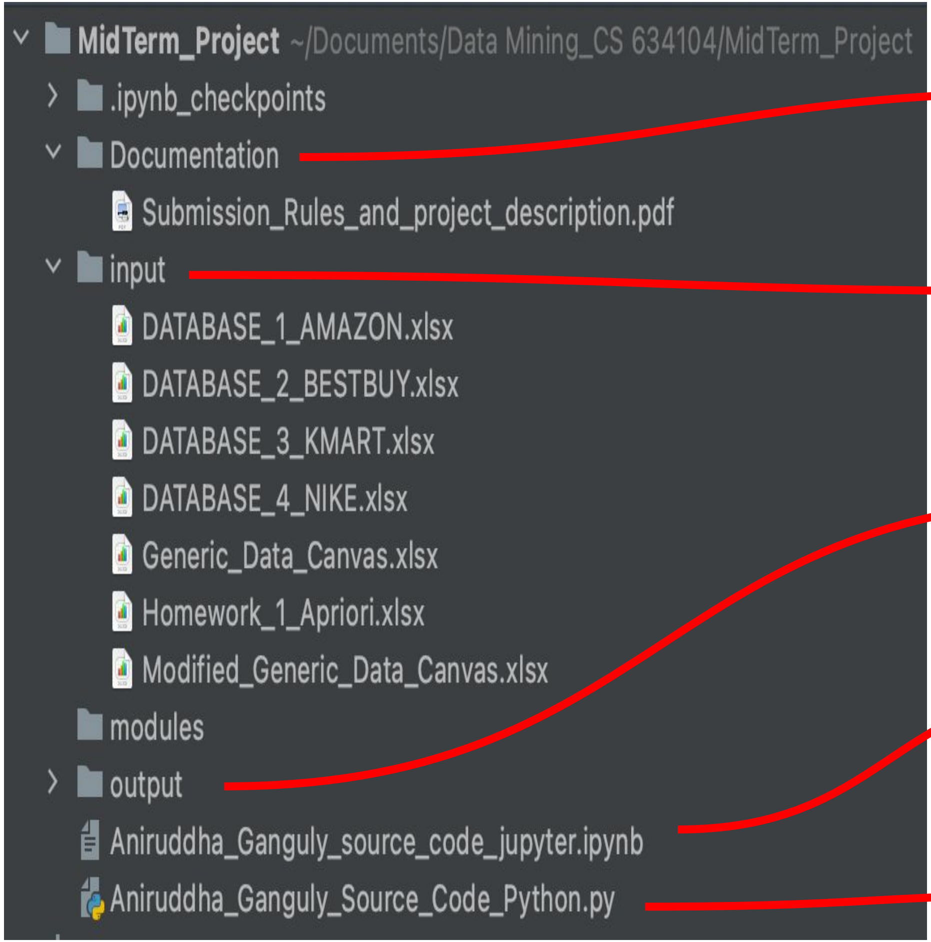
4. Prune the High Frequency Item set

5. Get All the Item combinations which have high frequency of occurring

6. Generate All possible Permutations from the High Frequency Item combinations And Calculate Confidence

7. Filter All Rules generated with Minimum Confidence and print the Rules

Folder Structure Description



```
v MidTerm_Project ~/Documents/Data Mining_CS 634104/MidTerm_Project
    > .ipynb_checkpoints
    v Documentation
        Submission_Rules_and_project_description.pdf
    v input
        DATABASE_1_AMAZON.xlsx
        DATABASE_2_BESTBUY.xlsx
        DATABASE_3_KMART.xlsx
        DATABASE_4_NIKE.xlsx
        Generic_Data_Canvas.xlsx
        Homework_1_Apriori.xlsx
        Modified_Generic_Data_Canvas.xlsx
      modules
    > output
      Aniruddha_Ganguly_source_code_jupyter.ipynb
      Aniruddha_Ganguly_Source_Code_Python.py
```

Documentation Folder:

- Contains All the Project Related Documentation

input Folder:

- Contains All the Input Files
- Each File containing two tabs named Items and Transactions

output Folder:

- Contains all the generated output files from the Apriori algorithm
- All output files are in .csv format

Source_code (Type : Jupyter Notebook (.ipynb)):

- Describes all the functions with the headers. This os the ipython version of the code

Source_code (Type : Python File):

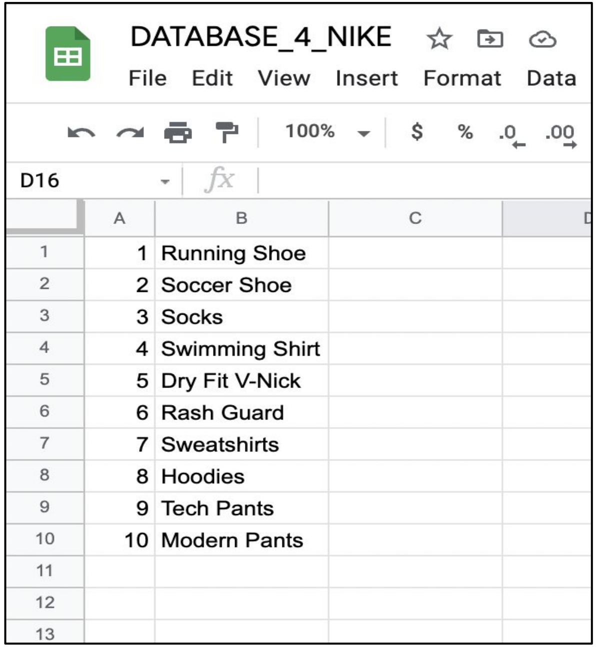
- All functions are written here with the __main__.

******* Please execute the file to generate and print the association rules**

Input Sample Structure (Type : .xlsx)

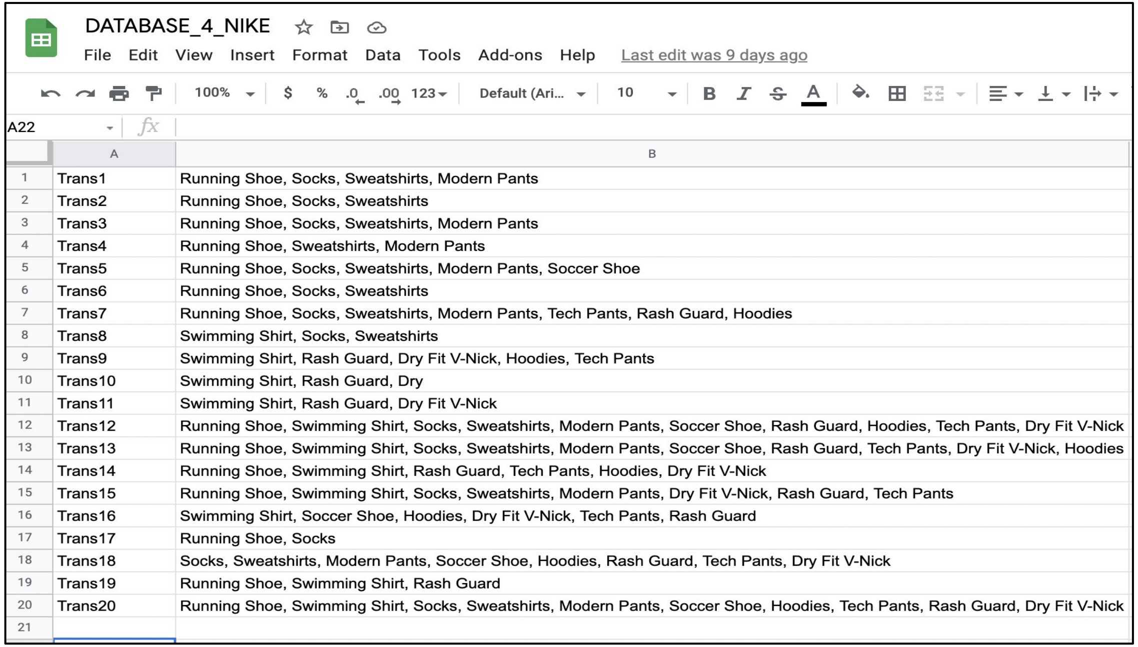
The Input file consists of two tabs named (1. Items & 2.Transactions). Below are the sample structure of the file.

Items



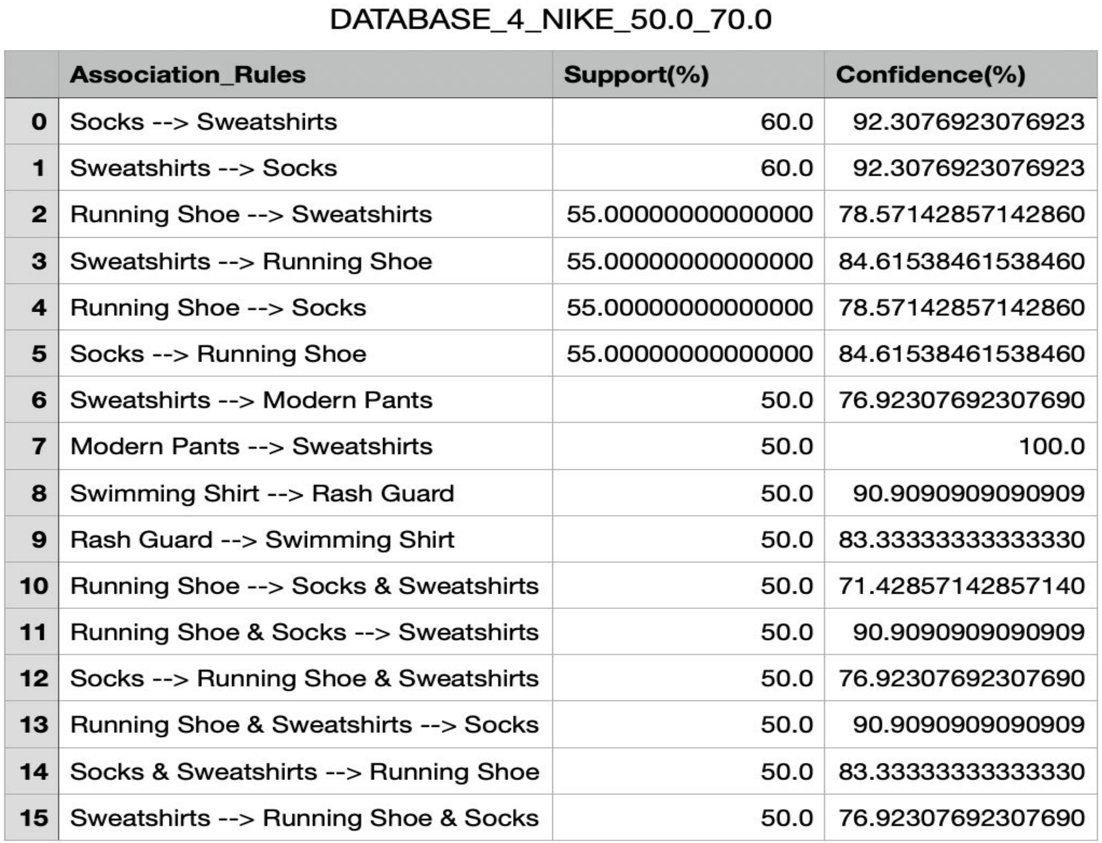
	A	B	C	D
1	1	Running Shoe		
2	2	Soccer Shoe		
3	3	Socks		
4	4	Swimming Shirt		
5	5	Dry Fit V-Nick		
6	6	Rash Guard		
7	7	Sweatshirts		
8	8	Hoodies		
9	9	Tech Pants		
10	10	Modern Pants		
11				
12				
13				

Transactions



	A	B
1	Trans1	Running Shoe, Socks, Sweatshirts, Modern Pants
2	Trans2	Running Shoe, Socks, Sweatshirts
3	Trans3	Running Shoe, Socks, Sweatshirts, Modern Pants
4	Trans4	Running Shoe, Sweatshirts, Modern Pants
5	Trans5	Running Shoe, Socks, Sweatshirts, Modern Pants, Soccer Shoe
6	Trans6	Running Shoe, Socks, Sweatshirts
7	Trans7	Running Shoe, Socks, Sweatshirts, Modern Pants, Tech Pants, Rash Guard, Hoodies
8	Trans8	Swimming Shirt, Socks, Sweatshirts
9	Trans9	Swimming Shirt, Rash Guard, Dry Fit V-Nick, Hoodies, Tech Pants
10	Trans10	Swimming Shirt, Rash Guard, Dry
11	Trans11	Swimming Shirt, Rash Guard, Dry Fit V-Nick
12	Trans12	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Soccer Shoe, Rash Guard, Hoodies, Tech Pants, Dry Fit V-Nick
13	Trans13	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Soccer Shoe, Rash Guard, Tech Pants, Dry Fit V-Nick, Hoodies
14	Trans14	Running Shoe, Swimming Shirt, Rash Guard, Tech Pants, Hoodies, Dry Fit V-Nick
15	Trans15	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Dry Fit V-Nick, Rash Guard, Tech Pants
16	Trans16	Swimming Shirt, Soccer Shoe, Hoodies, Dry Fit V-Nick, Tech Pants, Rash Guard
17	Trans17	Running Shoe, Socks
18	Trans18	Socks, Sweatshirts, Modern Pants, Soccer Shoe, Hoodies, Rash Guard, Tech Pants, Dry Fit V-Nick
19	Trans19	Running Shoe, Swimming Shirt, Rash Guard
20	Trans20	Running Shoe, Swimming Shirt, Socks, Sweatshirts, Modern Pants, Soccer Shoe, Hoodies, Tech Pants, Rash Guard, Dry Fit V-Nick
21		

Output Sample Structure (Type : .csv)



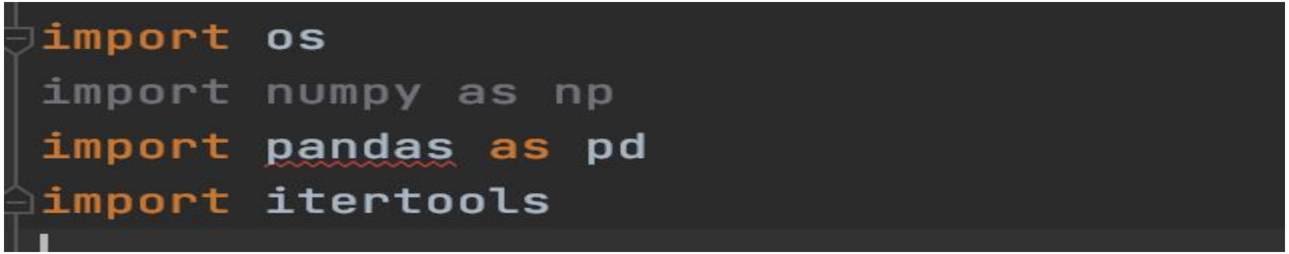
DATABASE_4_NIKE_50.0_70.0

	Association_Rules	Support(%)	Confidence(%)
0	Socks --> Sweatshirts	60.0	92.3076923076923
1	Sweatshirts --> Socks	60.0	92.3076923076923
2	Running Shoe --> Sweatshirts	55.00000000000000	78.57142857142860
3	Sweatshirts --> Running Shoe	55.00000000000000	84.61538461538460
4	Running Shoe --> Socks	55.00000000000000	78.57142857142860
5	Socks --> Running Shoe	55.00000000000000	84.61538461538460
6	Sweatshirts --> Modern Pants	50.0	76.92307692307690
7	Modern Pants --> Sweatshirts	50.0	100.0
8	Swimming Shirt --> Rash Guard	50.0	90.9090909090909
9	Rash Guard --> Swimming Shirt	50.0	83.33333333333330
10	Running Shoe --> Socks & Sweatshirts	50.0	71.42857142857140
11	Running Shoe & Socks --> Sweatshirts	50.0	90.9090909090909
12	Socks --> Running Shoe & Sweatshirts	50.0	76.92307692307690
13	Running Shoe & Sweatshirts --> Socks	50.0	90.9090909090909
14	Socks & Sweatshirts --> Running Shoe	50.0	83.33333333333330
15	Sweatshirts --> Running Shoe & Socks	50.0	76.92307692307690

Code Description

Python Version : 3.8
Os : MacOS Big Sur

Modules Used:



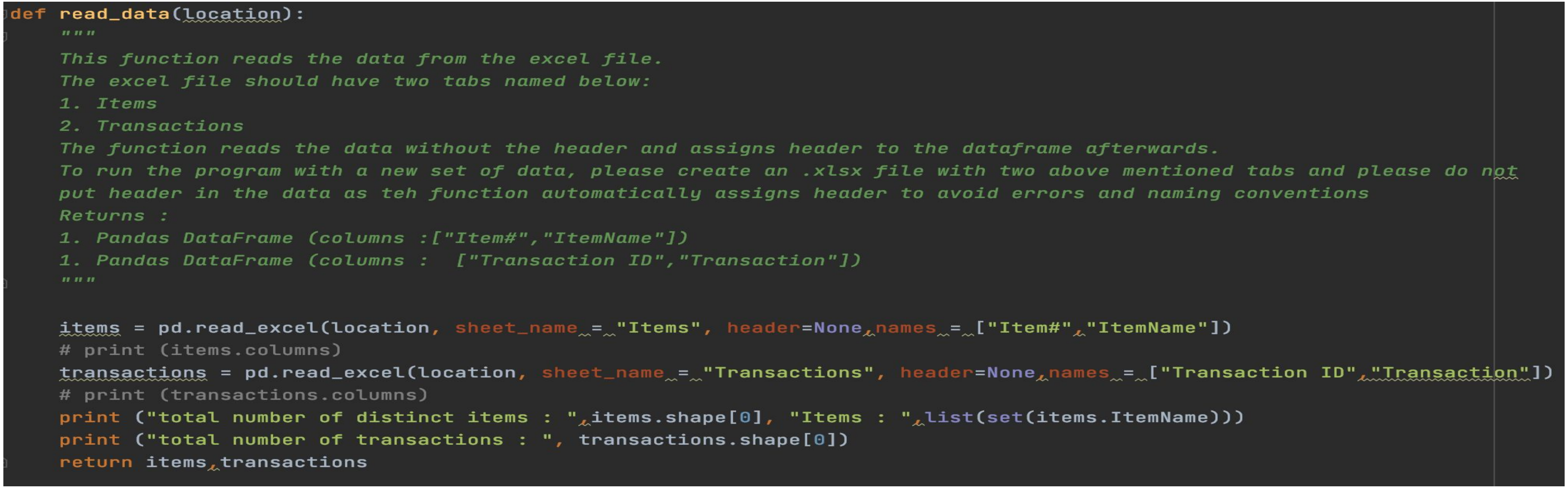
```
import os
import numpy as np
import pandas as pd
import itertools
```

** Please Run the below lines in your terminal if any of the following modules are missing

```
pip3 install pandas
pip3 install itertools
```

Function 1: read_data(location)

This function reads the data from the excel file with the location as input using the existing python modules (os, pandas) and assigns header afterwards. Please make sure that the input excel file does not have any header while running the function.
The excel file should have two tabs named
1. Items : Distinct Items)
2. Transactions : Transaction Related to the Items



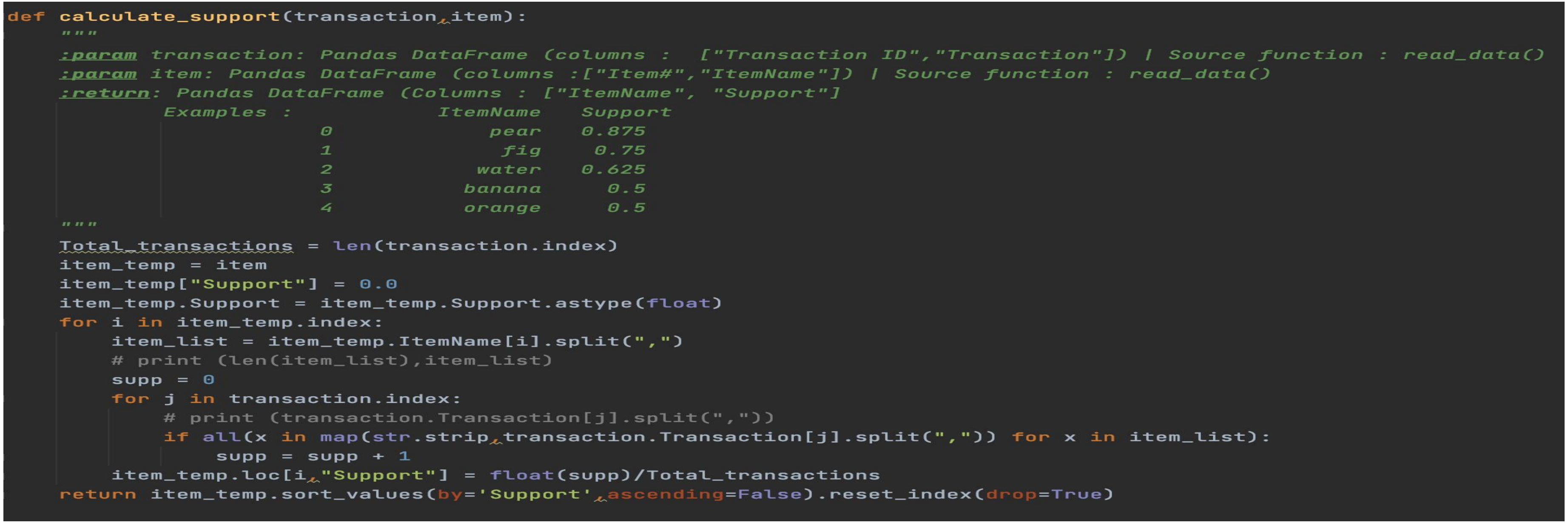
```python
def read_data(location):
    """
    This function reads the data from the excel file.
    The excel file should have two tabs named below:
    1. Items
    2. Transactions
    The function reads the data without the header and assigns header to the dataframe afterwards.
    To run the program with a new set of data, please create an .xlsx file with two above mentioned tabs and please do not
    put header in the data as teh function automatically assigns header to avoid errors and naming conventions
    Returns :
    1. Pandas DataFrame (columns :["Item#","ItemName"])
    1. Pandas DataFrame (columns :  ["Transaction ID","Transaction"])
    """

    items = pd.read_excel(location, sheet_name = "Items", header=None,names = ["Item#","ItemName"])
    # print (items.columns)
    transactions = pd.read_excel(location, sheet_name = "Transactions", header=None,names = ["Transaction ID","Transaction"])
    # print (transactions.columns)
    print ("total number of distinct items : ",items.shape[0], "Items : ",list(set(items.ItemName)))
    print ("total number of transactions : ", transactions.shape[0])
    return items,transactions
```

Function 2: calculate_support(transaction,item)

This function takes two dataframes (Items and Transactions) as input and calculate support of each individual Items as well as combinations of items.

This is a standalone function which is used to generate support.

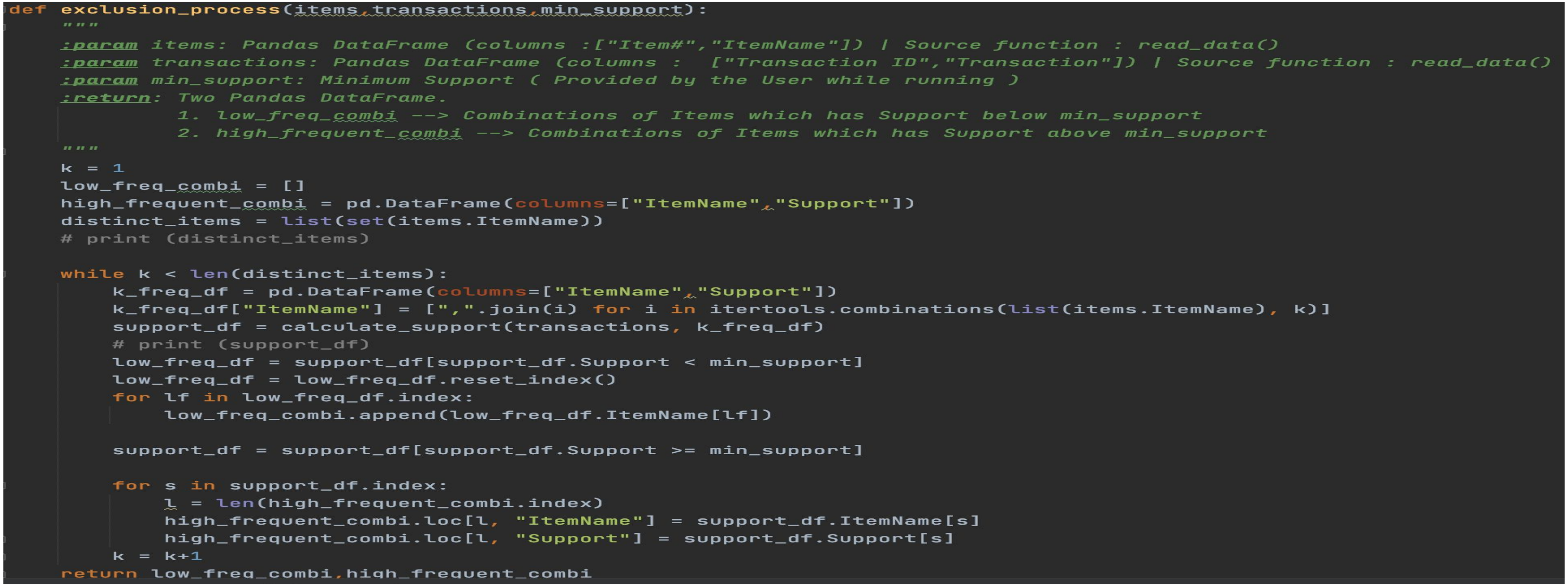
Returns a sorted pandas DataFrame.



```python
def calculate_support(transaction,item):
    """
    :param transaction: Pandas DataFrame (columns :  ["Transaction ID","Transaction"]) | Source function : read_data()
    :param item: Pandas DataFrame (columns :["Item#","ItemName"]) | Source function : read_data()
    :return: Pandas DataFrame (Columns : ["ItemName", "Support"]
            Examples :           ItemName   Support
                        0            pear   0.875
                        1             fig    0.75
                        2           water   0.625
                        3          banana     0.5
                        4          orange     0.5
    """
    Total_transactions = len(transaction.index)
    item_temp = item
    item_temp["Support"] = 0.0
    item_temp.Support = item_temp.Support.astype(float)
    for i in item_temp.index:
        item_list = item_temp.ItemName[i].split(",")
        # print (len(item_list),item_list)
        supp = 0
        for j in transaction.index:
            # print (transaction.Transaction[j].split(","))
            if all(x in map(str.strip,transaction.Transaction[j].split(",")) for x in item_list):
                supp = supp + 1
        item_temp.loc[i,"Support"] = float(supp)/Total_transactions
    return item_temp.sort_values(by='Support',ascending=False).reset_index(drop=True)
```

Function 3: exclusion_process(items,transactions,min_support)

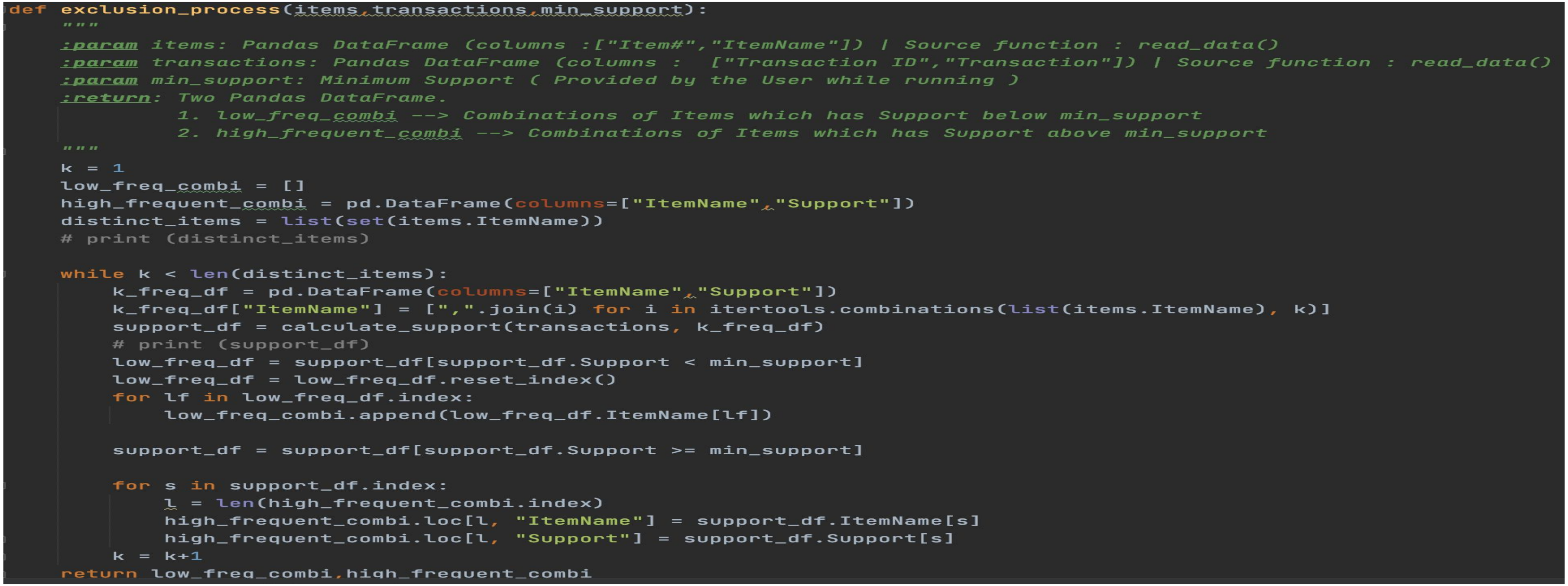
This function takes items, transactions and the minimum support (user input) as input and uses the calculate_support() function to generate the support and divide the items combinations into two separate dataFrames based on low/high frequency of occurrence in the transactions set.



```
def exclusion_process(items,transactions,min_support):
    """
    :param items: Pandas DataFrame (columns :["Item#","ItemName"]) | Source function : read_data()
    :param transactions: Pandas DataFrame (columns :  ["Transaction ID","Transaction"]) | Source function : read_data()
    :param min_support: Minimum Support ( Provided by the User while running )
    :return: Two Pandas DataFrame.
             1. low_freq_combi --> Combinations of Items which has Support below min_support
             2. high_frequent_combi --> Combinations of Items which has Support above min_support
    """
    k = 1
    low_freq_combi = []
    high_frequent_combi = pd.DataFrame(columns=["ItemName","Support"])
    distinct_items = list(set(items.ItemName))
    # print (distinct_items)

    while k < len(distinct_items):
        k_freq_df = pd.DataFrame(columns=["ItemName","Support"])
        k_freq_df["ItemName"] = [",".join(i) for i in itertools.combinations(list(items.ItemName), k)]
        support_df = calculate_support(transactions, k_freq_df)
        # print (support_df)
        low_freq_df = support_df[support_df.Support < min_support]
        low_freq_df = low_freq_df.reset_index()
        for lf in low_freq_df.index:
            low_freq_combi.append(low_freq_df.ItemName[lf])

        support_df = support_df[support_df.Support >= min_support]

        for s in support_df.index:
            l = len(high_frequent_combi.index)
            high_frequent_combi.loc[l, "ItemName"] = support_df.ItemName[s]
            high_frequent_combi.loc[l, "Support"] = support_df.Support[s]
        k = k+1
    return low_freq_combi,high_frequent_combi
```

Function 4: pruning(low_frequency_combi,high_frequency_combi)

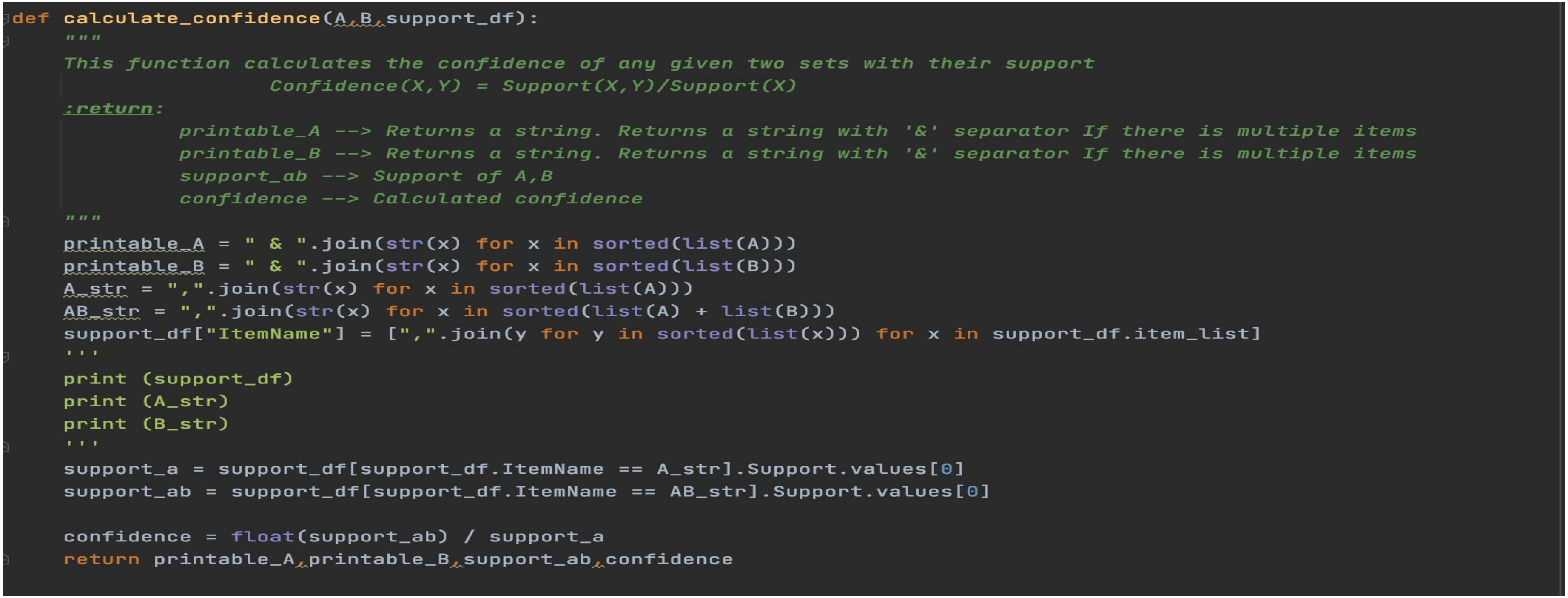
This function takes the item combination of low and high frequency and prune the process for optimized results and avoid the redundancy.
Theory : “The superset of a low frequency item would have low frequency”



```
def exclusion_process(items,transactions,min_support):
    """
    :param items: Pandas DataFrame (columns :["Item#","ItemName"]) | Source function : read_data()
    :param transactions: Pandas DataFrame (columns :  ["Transaction ID","Transaction"]) | Source function : read_data()
    :param min_support: Minimum Support ( Provided by the User while running )
    :return: Two Pandas DataFrame.
             1. low_freq_combi --> Combinations of Items which has Support below min_support
             2. high_frequent_combi --> Combinations of Items which has Support above min_support
    """
    k = 1
    low_freq_combi = []
    high_frequent_combi = pd.DataFrame(columns=["ItemName","Support"])
    distinct_items = list(set(items.ItemName))
    # print (distinct_items)

    while k < len(distinct_items):
        k_freq_df = pd.DataFrame(columns=["ItemName","Support"])
        k_freq_df["ItemName"] = [",".join(i) for i in itertools.combinations(list(items.ItemName), k)]
        support_df = calculate_support(transactions, k_freq_df)
        # print (support_df)
        low_freq_df = support_df[support_df.Support < min_support]
        low_freq_df = low_freq_df.reset_index()
        for lf in low_freq_df.index:
            low_freq_combi.append(low_freq_df.ItemName[lf])

        support_df = support_df[support_df.Support >= min_support]

        for s in support_df.index:
            l = len(high_frequent_combi.index)
            high_frequent_combi.loc[l, "ItemName"] = support_df.ItemName[s]
            high_frequent_combi.loc[l, "Support"] = support_df.Support[s]
        k = k+1
    return low_freq_combi,high_frequent_combi
```

Function 5: calculate_confidence(list1, list2, support_df)

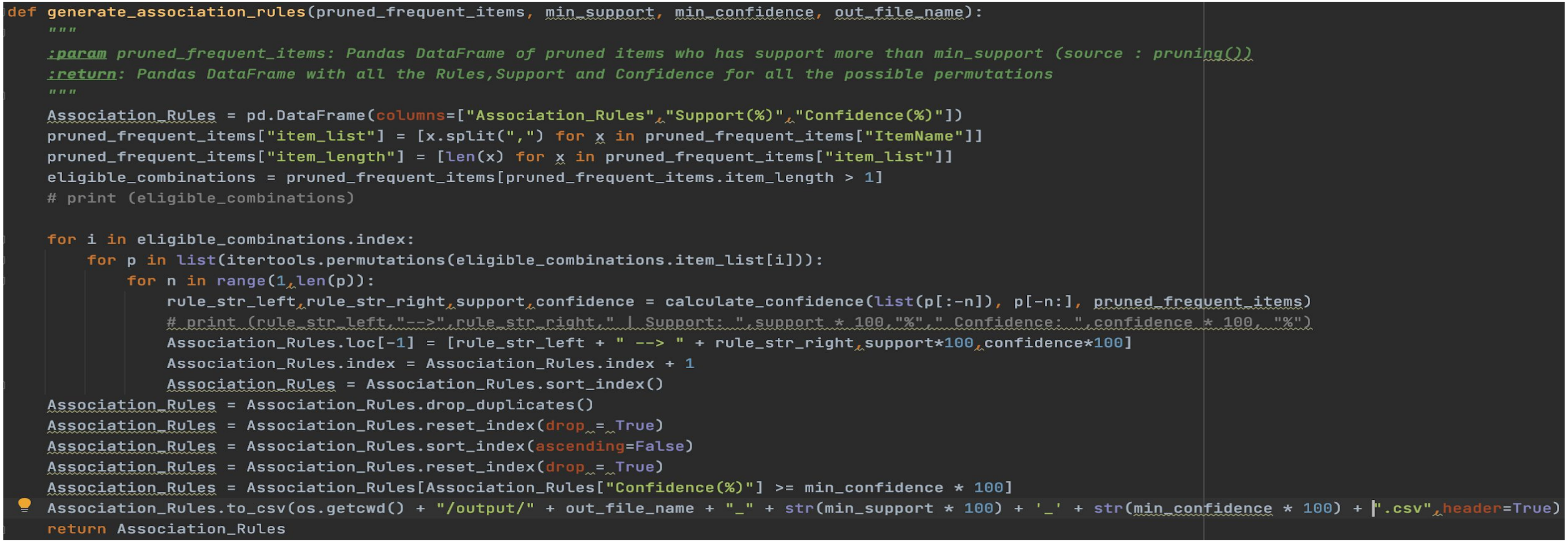
This function takes a list items and the support as input and calculate confidence of all permutations of the items. This would help to generate association rules given minimum confidence as user input.



```
def calculate_confidence(A,B,support_df):
    """
    This function calculates the confidence of any given two sets with their support
                    Confidence(X,Y) = Support(X,Y)/Support(X)
    :return:
             printable_A --> Returns a string. Returns a string with '&' separator If there is multiple items
             printable_B --> Returns a string. Returns a string with '&' separator If there is multiple items
             support_ab --> Support of A,B
             confidence --> Calculated confidence
    """
    printable_A = " & ".join(str(x) for x in sorted(list(A)))
    printable_B = " & ".join(str(x) for x in sorted(list(B)))
    A_str = ",".join(str(x) for x in sorted(list(A)))
    AB_str = ",".join(str(x) for x in sorted(list(A) + list(B)))
    support_df["ItemName"] = [",".join(y for y in sorted(list(x))) for x in support_df.item_list]
    '''
    print (support_df)
    print (A_str)
    print (B_str)
    '''
    support_a = support_df[support_df.ItemName == A_str].Support.values[0]
    support_ab = support_df[support_df.ItemName == AB_str].Support.values[0]

    confidence = float(support_ab) / support_a
    return printable_A,printable_B,support_ab,confidence
```

Function 6: generate_association_rule(pruned_frequent_items, min_confidence)

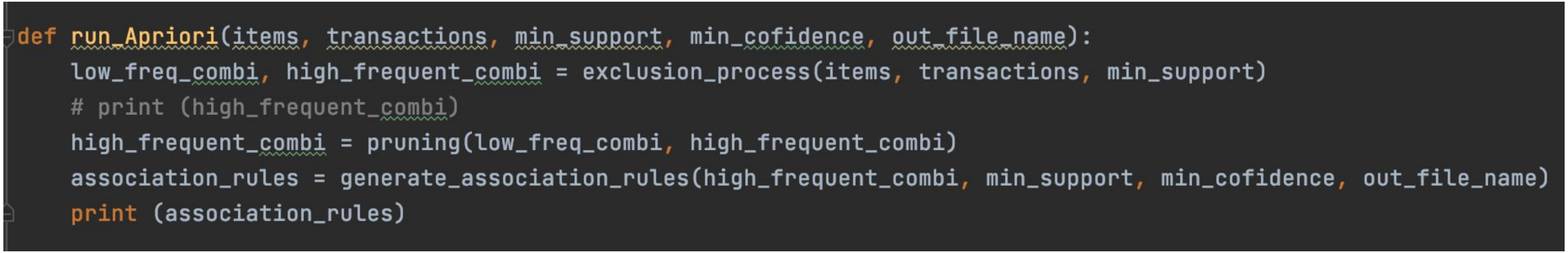
This function takes a list items and the support as input and calculate confidence of all permutations of the items. This would help to generate association rules given minimum confidence as user input.



```python
def generate_association_rules(pruned_frequent_items, min_support, min_confidence, out_file_name):
    """
    :param pruned_frequent_items: Pandas DataFrame of pruned items who has support more than min_support (source : pruning())
    :return: Pandas DataFrame with all the Rules,Support and Confidence for all the possible permutations
    """
    Association_Rules = pd.DataFrame(columns=["Association_Rules","Support(%)","Confidence(%)"])
    pruned_frequent_items["item_list"] = [x.split(",") for x in pruned_frequent_items["ItemName"]]
    pruned_frequent_items["item_length"] = [len(x) for x in pruned_frequent_items["item_list"]]
    eligible_combinations = pruned_frequent_items[pruned_frequent_items.item_length > 1]
    # print (eligible_combinations)

    for i in eligible_combinations.index:
        for p in list(itertools.permutations(eligible_combinations.item_list[i])):
            for n in range(1,len(p)):
                rule_str_left,rule_str_right,support,confidence = calculate_confidence(list(p[:-n]), p[-n:], pruned_frequent_items)
                # print (rule_str_left,"-->",rule_str_right," | Support: ",support * 100,"%"," Confidence: ",confidence * 100, "%")
                Association_Rules.loc[-1] = [rule_str_left + " --> " + rule_str_right,support*100,confidence*100]
                Association_Rules.index = Association_Rules.index + 1
                Association_Rules = Association_Rules.sort_index()
    Association_Rules = Association_Rules.drop_duplicates()
    Association_Rules = Association_Rules.reset_index(drop = True)
    Association_Rules = Association_Rules.sort_index(ascending=False)
    Association_Rules = Association_Rules.reset_index(drop = True)
    Association_Rules = Association_Rules[Association_Rules["Confidence(%)"] >= min_confidence * 100]
    Association_Rules.to_csv(os.getcwd() + "/output/" + out_file_name + "_" + str(min_support * 100) + '_' + str(min_confidence * 100) + ".csv",header=True)
    return Association_Rules
```

Function 7: run_Apriori(items, transactions, min_support, min_confidence)

This is a wrapper function which triggers all the above functions in a step by step manner to generate the association rules.



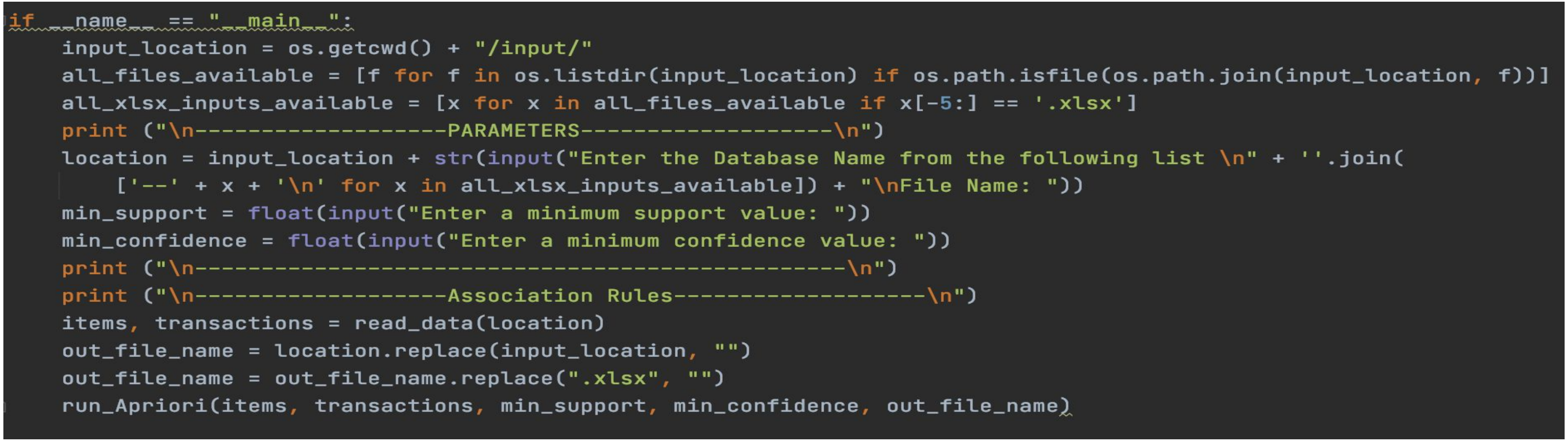
```
def run_Apriori(items, transactions, min_support, min_cofidence, out_file_name):
    low_freq_combi, high_frequent_combi = exclusion_process(items, transactions, min_support)
    # print (high_frequent_combi)
    high_frequent_combi = pruning(low_freq_combi, high_frequent_combi)
    association_rules = generate_association_rules(high_frequent_combi, min_support, min_cofidence, out_file_name)
    print (association_rules)
```

Function MAIN:

The main function takes three inline inputs:

1. DataBase Name (The user has to choose from the printed list)
2. Minimum Support
3. Minimum Confidence

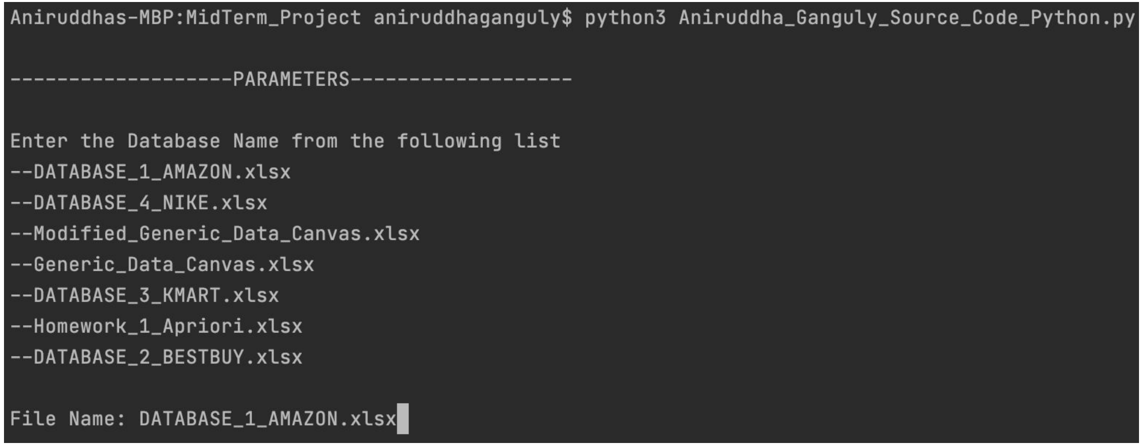
And it prints out the Association Rules as well as saves the Rules in output folder.



```
if __name__ == "__main__":
    input_location = os.getcwd() + "/input/"
    all_files_available = [f for f in os.listdir(input_location) if os.path.isfile(os.path.join(input_location, f))]
    all_xlsx_inputs_available = [x for x in all_files_available if x[-5:] == '.xlsx']
    print ("\n--------------------PARAMETERS--------------------\n")
    location = input_location + str(input("Enter the Database Name from the following list \n" + ''.join(
        ['--' + x + '\n' for x in all_xlsx_inputs_available]) + "\nFile Name: "))
    min_support = float(input("Enter a minimum support value: "))
    min_confidence = float(input("Enter a minimum confidence value: "))
    print ("\n--------------------------------------------------\n")
    print ("\n-------------------Association Rules-------------------\n")
    items, transactions = read_data(location)
    out_file_name = location.replace(input_location, "")
    out_file_name = out_file_name.replace(".xlsx", "")
    run_Apriori(items, transactions, min_support, min_confidence, out_file_name)
```

How to Run the code?

1. Open a Terminal and go to the Project Folder
2. Run “Python3 Aniruddha_Ganguly_Source_Code_Python.py”
3. The Program would start running and ask you parameters in the terminal:
   Parameter 1: Database Name



```
Aniruddhas-MBP:MidTerm_Project aniruddhaganguly$ python3 Aniruddha_Ganguly_Source_Code_Python.py

-------------------PARAMETERS-------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Modified_Generic_Data_Canvas.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_1_AMAZON.xlsx
```

Please enter any file name from the above prompt list as those are the data available.

Parameter 2&3: Minimum Support & Minimum Confidence - Please Enter minimum support and minimum confidence in decimal form

Example:

If required minimum support is 53% → Please enter 0.53

If required minimum confidence is 71% → Please enter 0.71

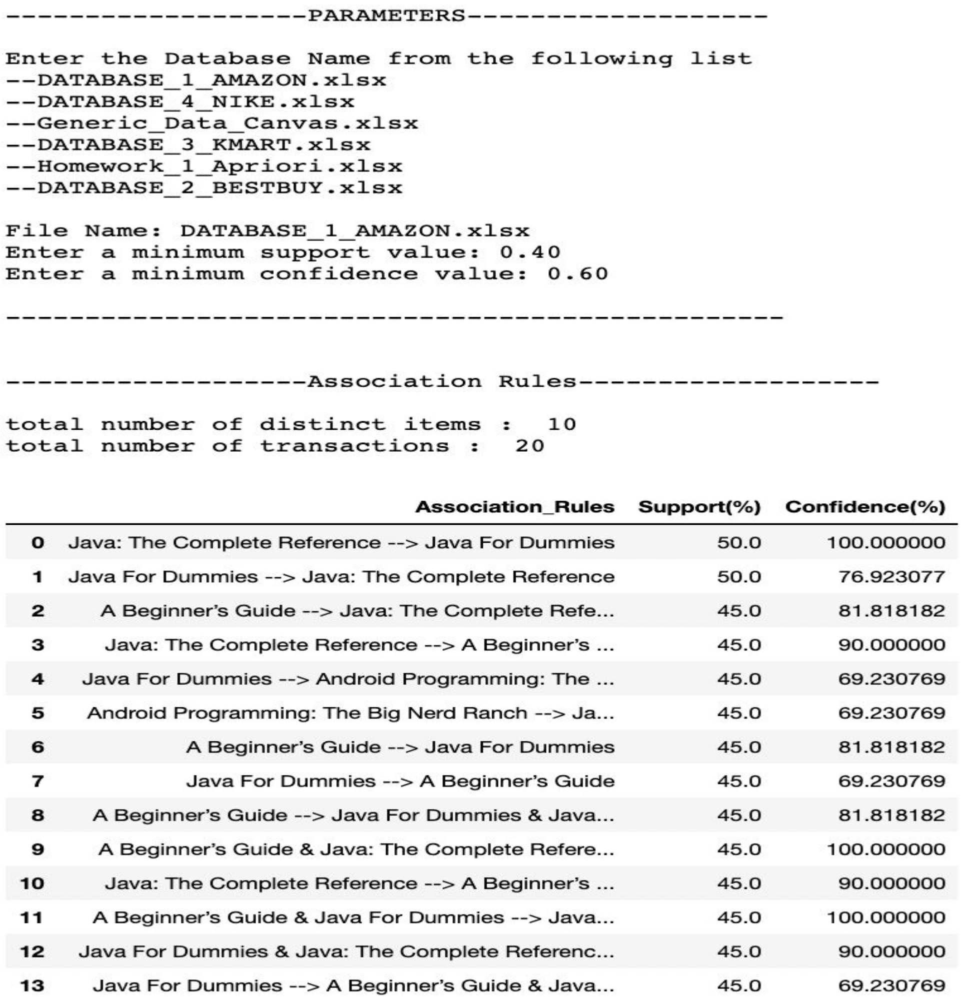
Results

DATABASE 1: AMAZON

Minimum Support : **0.40** (40%)
Minimum Confidence : **0.60** (60%)

Dataset	**Amazon**
Source	**Canvas**
File Name	**DATABASE_1_AMAZON.xlsx**
Minimum Support	**0.40**
Minimum Confidence	**0.60**

Input Parameters



```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_1_AMAZON.xlsx
Enter a minimum support value: 0.40
Enter a minimum confidence value: 0.60

-------------------------------------------------


-------------------Association Rules-------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	Java: The Complete Reference --> Java For Dummies	50.0	100.000000
1	Java For Dummies --> Java: The Complete Reference	50.0	76.923077
2	A Beginner's Guide --> Java: The Complete Refe...	45.0	81.818182
3	Java: The Complete Reference --> A Beginner's ...	45.0	90.000000
4	Java For Dummies --> Android Programming: The ...	45.0	69.230769
5	Android Programming: The Big Nerd Ranch --> Ja...	45.0	69.230769
6	A Beginner's Guide --> Java For Dummies	45.0	81.818182
7	Java For Dummies --> A Beginner's Guide	45.0	69.230769
8	A Beginner's Guide --> Java For Dummies & Java...	45.0	81.818182
9	A Beginner's Guide & Java: The Complete Refere...	45.0	100.000000
10	Java: The Complete Reference --> A Beginner's ...	45.0	90.000000
11	A Beginner's Guide & Java For Dummies --> Java...	45.0	100.000000
12	Java For Dummies & Java: The Complete Referenc...	45.0	90.000000
13	Java For Dummies --> A Beginner's Guide & Java...	45.0	69.230769

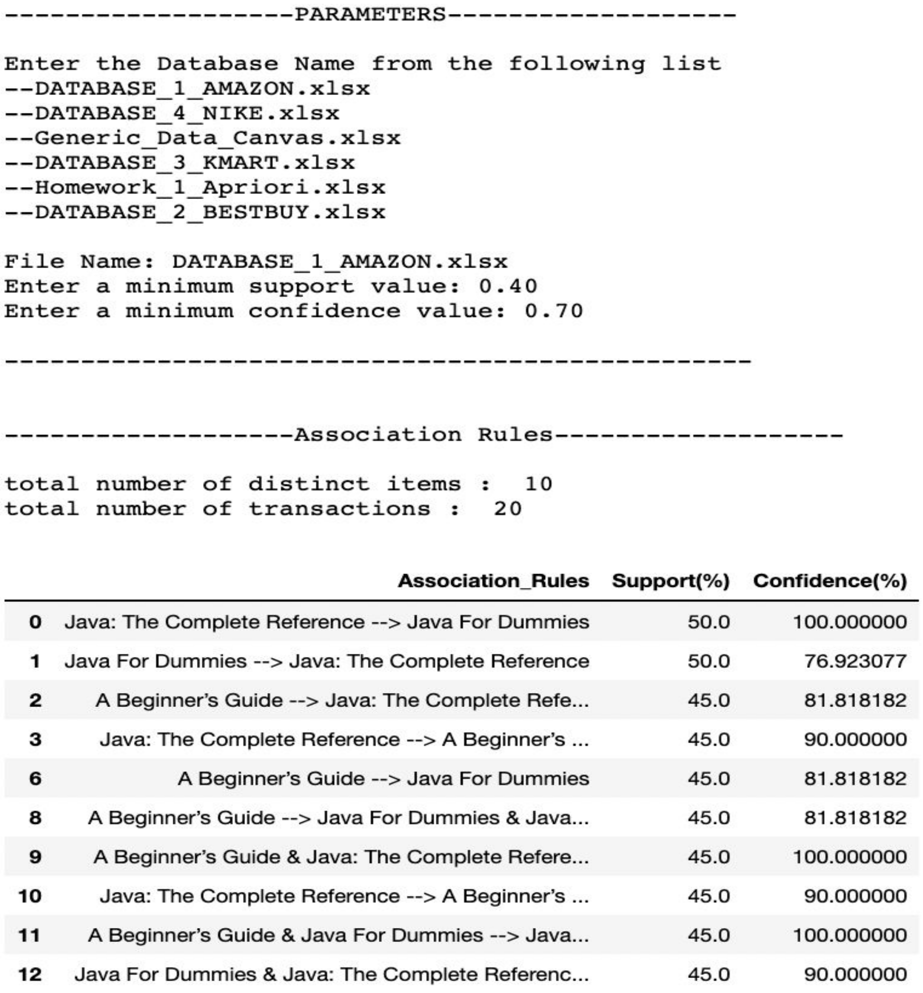
OUTPUT

Minimum Support : **0.40** (40%)
Minimum Confidence : **0.70** (70%)

Dataset	**Amazon**
Source	**Canvas**
File Name	**DATABASE_1_AMAZON.xlsx**
Minimum Support	**0.40**
Minimum Confidence	**0.70**

Input Parameters

OUTPUT



```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_1_AMAZON.xlsx
Enter a minimum support value: 0.40
Enter a minimum confidence value: 0.70

-------------------------------------------------


------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

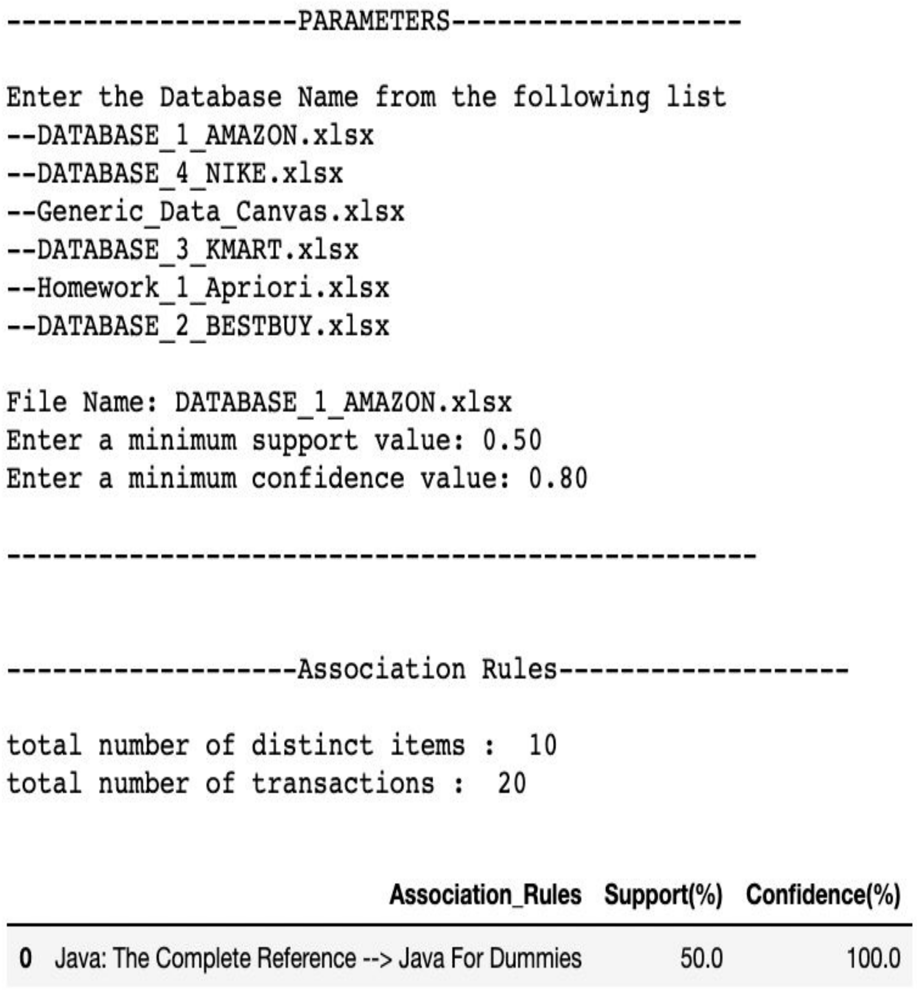
	Association_Rules	Support(%)	Confidence(%)
0	Java: The Complete Reference --> Java For Dummies	50.0	100.000000
1	Java For Dummies --> Java: The Complete Reference	50.0	76.923077
2	A Beginner's Guide --> Java: The Complete Refe...	45.0	81.818182
3	Java: The Complete Reference --> A Beginner's ...	45.0	90.000000
6	A Beginner's Guide --> Java For Dummies	45.0	81.818182
8	A Beginner's Guide --> Java For Dummies & Java...	45.0	81.818182
9	A Beginner's Guide & Java: The Complete Refere...	45.0	100.000000
10	Java: The Complete Reference --> A Beginner's ...	45.0	90.000000
11	A Beginner's Guide & Java For Dummies --> Java...	45.0	100.000000
12	Java For Dummies & Java: The Complete Referenc...	45.0	90.000000

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.80** (80%)

Dataset	**Amazon**
Source	**Canvas**
File Name	**DATABASE_1_AMAZON.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.80**

Input Parameters

OUTPUT



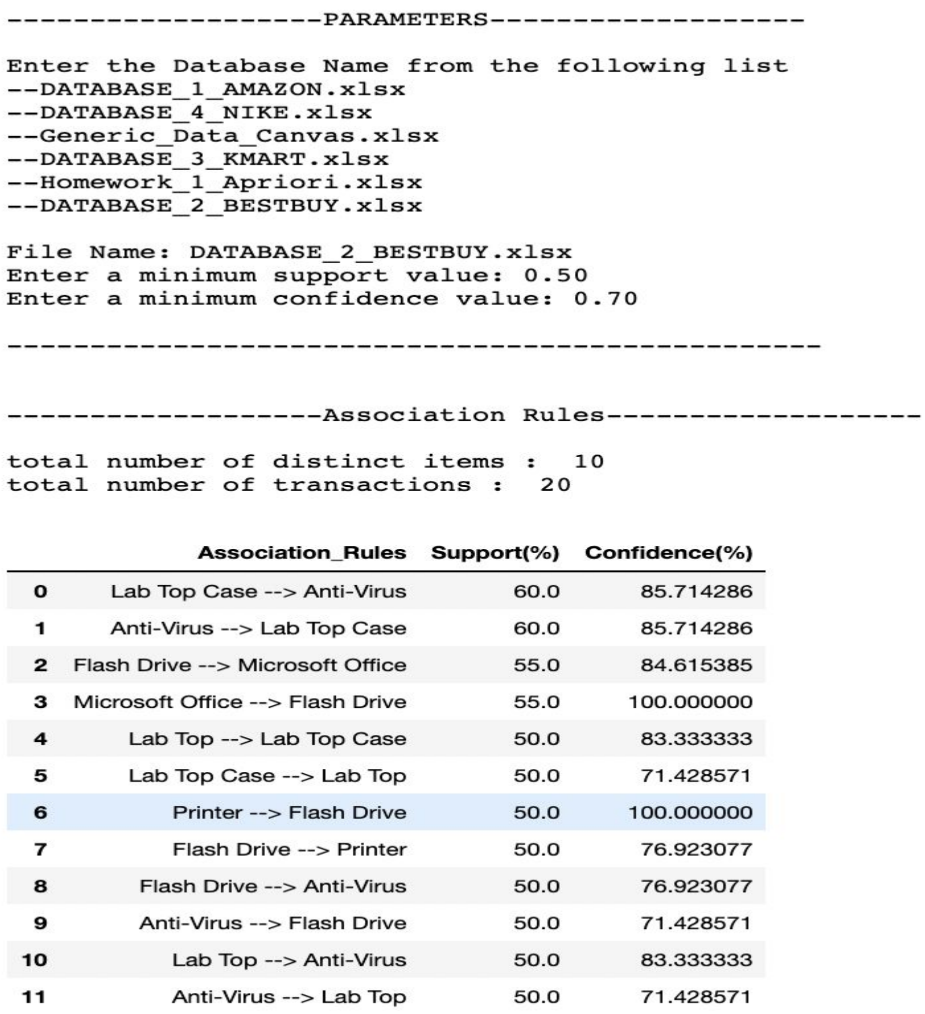
```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_1_AMAZON.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.80

-------------------------------------------------


-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20


                                      Association_Rules  Support(%)  Confidence(%)
0  Java: The Complete Reference --> Java For Dummies          50.0          100.0
```

DATABASE 2: Best Buy

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.70** (70%)

Dataset	**Best Buy**
Source	**Canvas**
File Name	**DATABASE_2_BESTBUY.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.70**

Input Parameters



```
------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_2_BESTBUY.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.70

-------------------------------------------------


------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	Lab Top Case --> Anti-Virus	60.0	85.714286
1	Anti-Virus --> Lab Top Case	60.0	85.714286
2	Flash Drive --> Microsoft Office	55.0	84.615385
3	Microsoft Office --> Flash Drive	55.0	100.000000
4	Lab Top --> Lab Top Case	50.0	83.333333
5	Lab Top Case --> Lab Top	50.0	71.428571
6	Printer --> Flash Drive	50.0	100.000000
7	Flash Drive --> Printer	50.0	76.923077
8	Flash Drive --> Anti-Virus	50.0	76.923077
9	Anti-Virus --> Flash Drive	50.0	71.428571
10	Lab Top --> Anti-Virus	50.0	83.333333
11	Anti-Virus --> Lab Top	50.0	71.428571

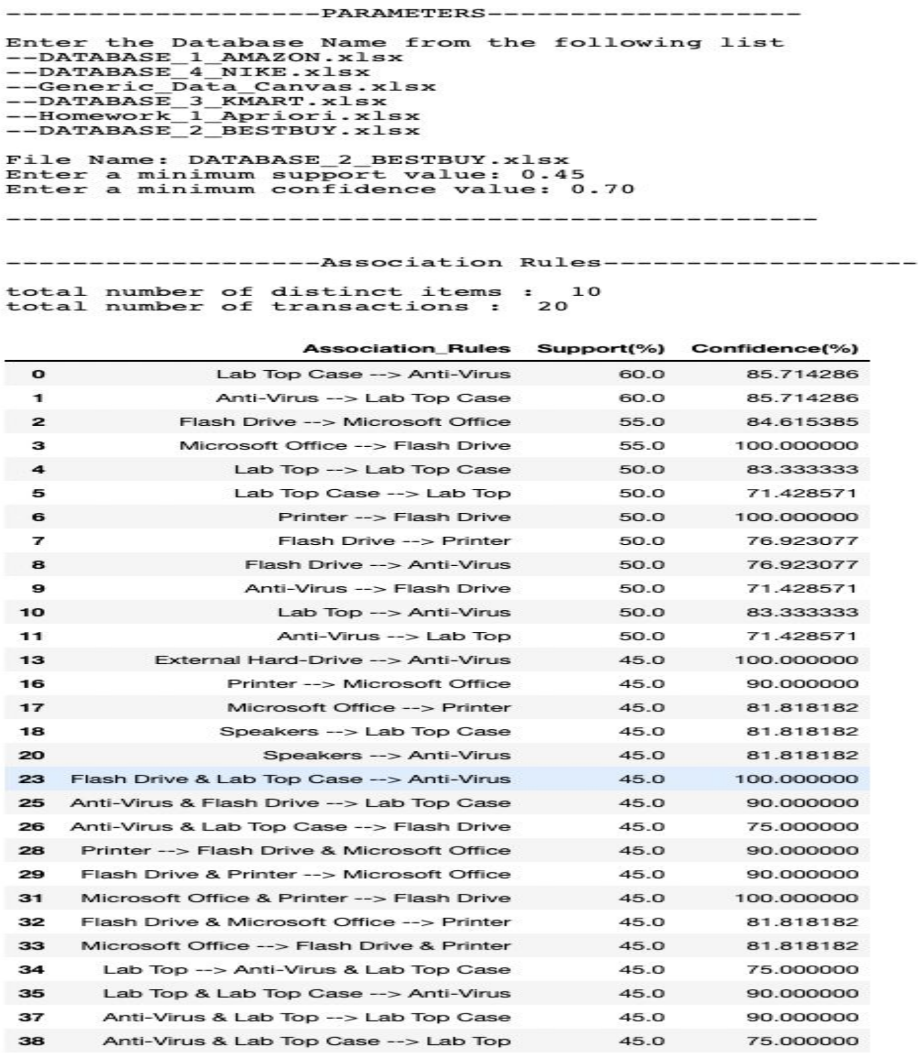
OUTPUT

Minimum Support : **0.45** (45%)
Minimum Confidence : **0.70** (70%)

Dataset	**Best Buy**
Source	**Canvas**
File Name	**DATABASE_2_BESTBUY.xlsx**
Minimum Support	**0.45**
Minimum Confidence	**0.70**

Input Parameters

OUTPUT



```
-------------------PARAMETERS------------------
Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_2_BESTBUY.xlsx
Enter a minimum support value: 0.45
Enter a minimum confidence value: 0.70

-------------------------------------------------

-------------------Association Rules-------------------

total number of distinct items :  10
total number of transactions :  20
```

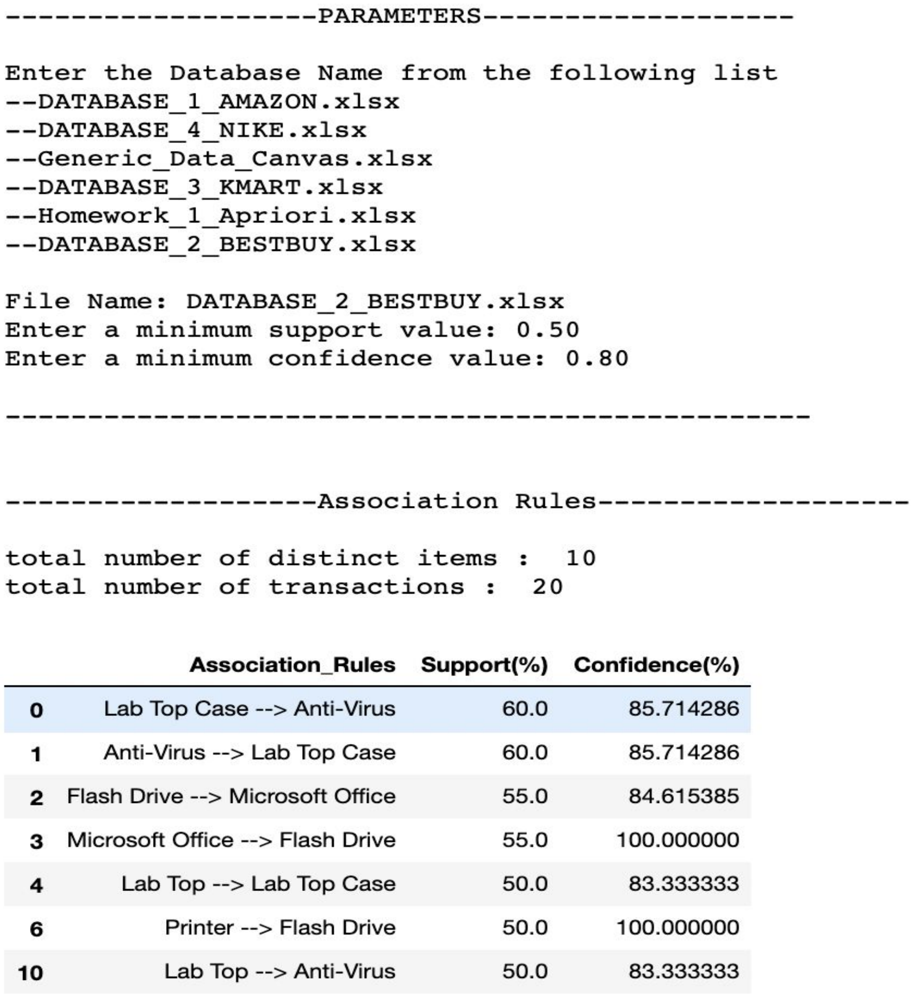
	Association_Rules	Support(%)	Confidence(%)
0	Lab Top Case --> Anti-Virus	60.0	85.714286
1	Anti-Virus --> Lab Top Case	60.0	85.714286
2	Flash Drive --> Microsoft Office	55.0	84.615385
3	Microsoft Office --> Flash Drive	55.0	100.000000
4	Lab Top --> Lab Top Case	50.0	83.333333
5	Lab Top Case --> Lab Top	50.0	71.428571
6	Printer --> Flash Drive	50.0	100.000000
7	Flash Drive --> Printer	50.0	76.923077
8	Flash Drive --> Anti-Virus	50.0	76.923077
9	Anti-Virus --> Flash Drive	50.0	71.428571
10	Lab Top --> Anti-Virus	50.0	83.333333
11	Anti-Virus --> Lab Top	50.0	71.428571
13	External Hard-Drive --> Anti-Virus	45.0	100.000000
16	Printer --> Microsoft Office	45.0	90.000000
17	Microsoft Office --> Printer	45.0	81.818182
18	Speakers --> Lab Top Case	45.0	81.818182
20	Speakers --> Anti-Virus	45.0	81.818182
23	Flash Drive & Lab Top Case --> Anti-Virus	45.0	100.000000
25	Anti-Virus & Flash Drive --> Lab Top Case	45.0	90.000000
26	Anti-Virus & Lab Top Case --> Flash Drive	45.0	75.000000
28	Printer --> Flash Drive & Microsoft Office	45.0	90.000000
29	Flash Drive & Printer --> Microsoft Office	45.0	90.000000
31	Microsoft Office & Printer --> Flash Drive	45.0	100.000000
32	Flash Drive & Microsoft Office --> Printer	45.0	81.818182
33	Microsoft Office --> Flash Drive & Printer	45.0	81.818182
34	Lab Top --> Anti-Virus & Lab Top Case	45.0	75.000000
35	Lab Top & Lab Top Case --> Anti-Virus	45.0	90.000000
37	Anti-Virus & Lab Top --> Lab Top Case	45.0	90.000000
38	Anti-Virus & Lab Top Case --> Lab Top	45.0	75.000000

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.80** (80%)

Dataset	**Best Buy**
Source	**Canvas**
File Name	**DATABASE_2_BESTBUY.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.80**

Input Parameters

OUTPUT



```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_2_BESTBUY.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.80

-------------------------------------------------


-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

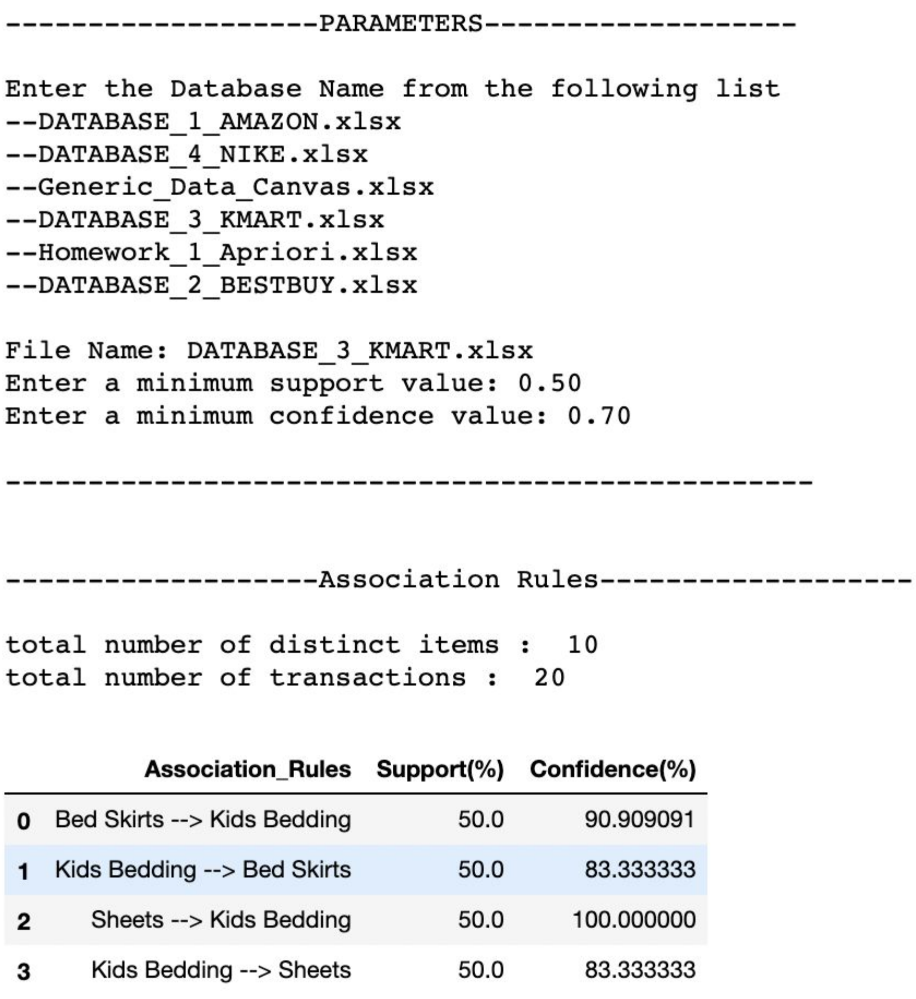
	Association_Rules	Support(%)	Confidence(%)
0	Lab Top Case --> Anti-Virus	60.0	85.714286
1	Anti-Virus --> Lab Top Case	60.0	85.714286
2	Flash Drive --> Microsoft Office	55.0	84.615385
3	Microsoft Office --> Flash Drive	55.0	100.000000
4	Lab Top --> Lab Top Case	50.0	83.333333
6	Printer --> Flash Drive	50.0	100.000000
10	Lab Top --> Anti-Virus	50.0	83.333333

DATABASE 3: KMART

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.70** (70%)

Dataset	**KMART**
Source	**Canvas**
File Name	**DATABASE_3_KMART.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.70**

Input Parameters



```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_3_KMART.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.70

-------------------------------------------------



-------------------Association Rules-------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	Bed Skirts --> Kids Bedding	50.0	90.909091
1	Kids Bedding --> Bed Skirts	50.0	83.333333
2	Sheets --> Kids Bedding	50.0	100.000000
3	Kids Bedding --> Sheets	50.0	83.333333

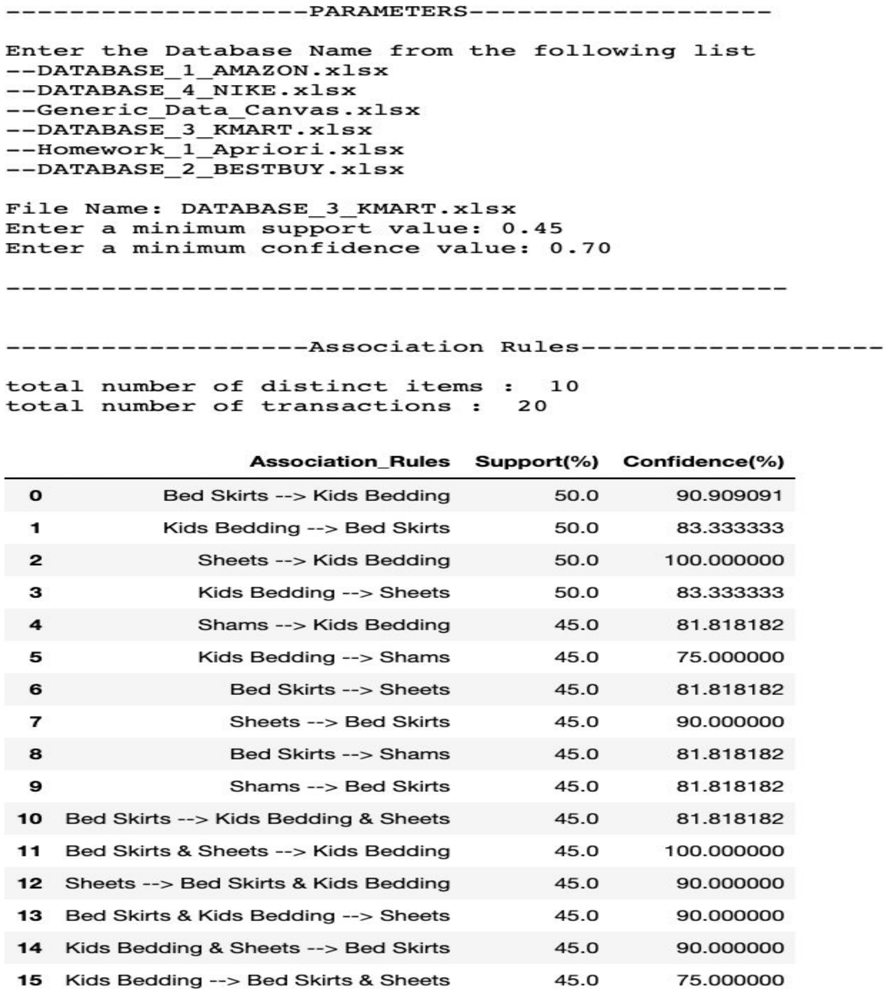
OUTPUT

Minimum Support : **0.45** (45%)
Minimum Confidence : **0.70** (70%)

Dataset	**KMART**
Source	**Canvas**
File Name	**DATABASE_3_KMART.xlsx**
Minimum Support	**0.45**
Minimum Confidence	**0.70**

Input Parameters

OUTPUT



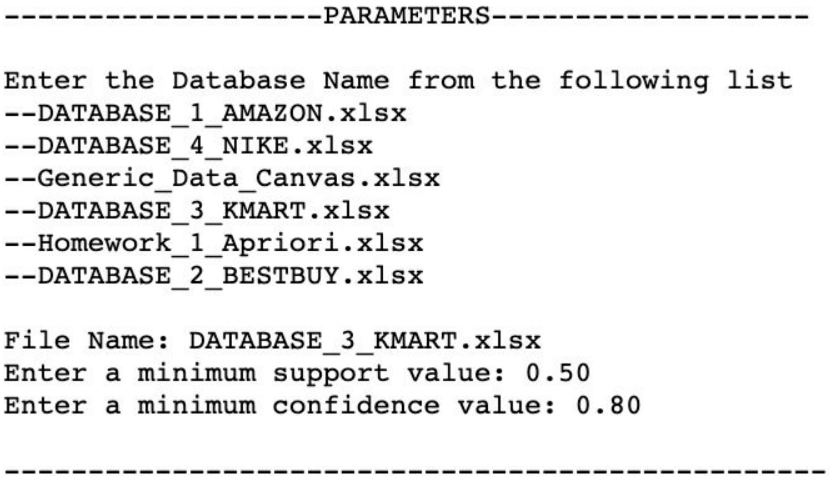
```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_3_KMART.xlsx
Enter a minimum support value: 0.45
Enter a minimum confidence value: 0.70

-------------------------------------------------


-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	Bed Skirts --> Kids Bedding	50.0	90.909091
1	Kids Bedding --> Bed Skirts	50.0	83.333333
2	Sheets --> Kids Bedding	50.0	100.000000
3	Kids Bedding --> Sheets	50.0	83.333333
4	Shams --> Kids Bedding	45.0	81.818182
5	Kids Bedding --> Shams	45.0	75.000000
6	Bed Skirts --> Sheets	45.0	81.818182
7	Sheets --> Bed Skirts	45.0	90.000000
8	Bed Skirts --> Shams	45.0	81.818182
9	Shams --> Bed Skirts	45.0	81.818182
10	Bed Skirts --> Kids Bedding & Sheets	45.0	81.818182
11	Bed Skirts & Sheets --> Kids Bedding	45.0	100.000000
12	Sheets --> Bed Skirts & Kids Bedding	45.0	90.000000
13	Bed Skirts & Kids Bedding --> Sheets	45.0	90.000000
14	Kids Bedding & Sheets --> Bed Skirts	45.0	90.000000
15	Kids Bedding --> Bed Skirts & Sheets	45.0	75.000000

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.80** (80%)

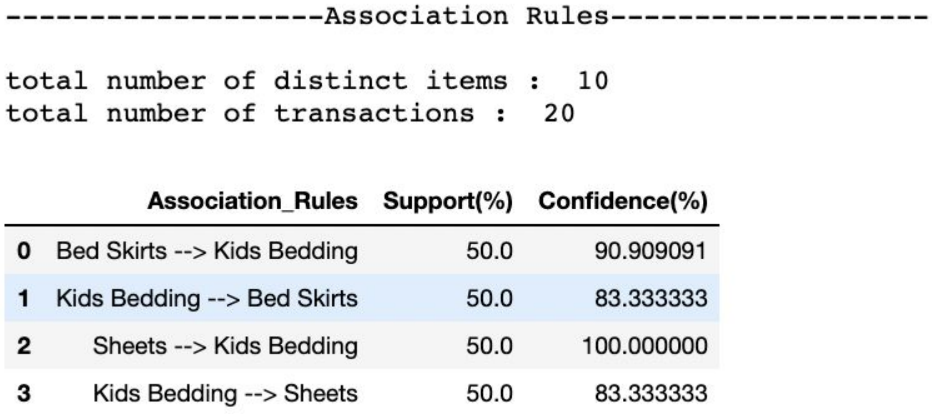
Dataset	**KMART**
Source	**Canvas**
File Name	**DATABASE_3_KMART.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.80**

Input Parameters



```
--------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_3_KMART.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.80

------------------------------------------------
```

OUTPUT



```
-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	Bed Skirts --> Kids Bedding	50.0	90.909091
1	Kids Bedding --> Bed Skirts	50.0	83.333333
2	Sheets --> Kids Bedding	50.0	100.000000
3	Kids Bedding --> Sheets	50.0	83.333333

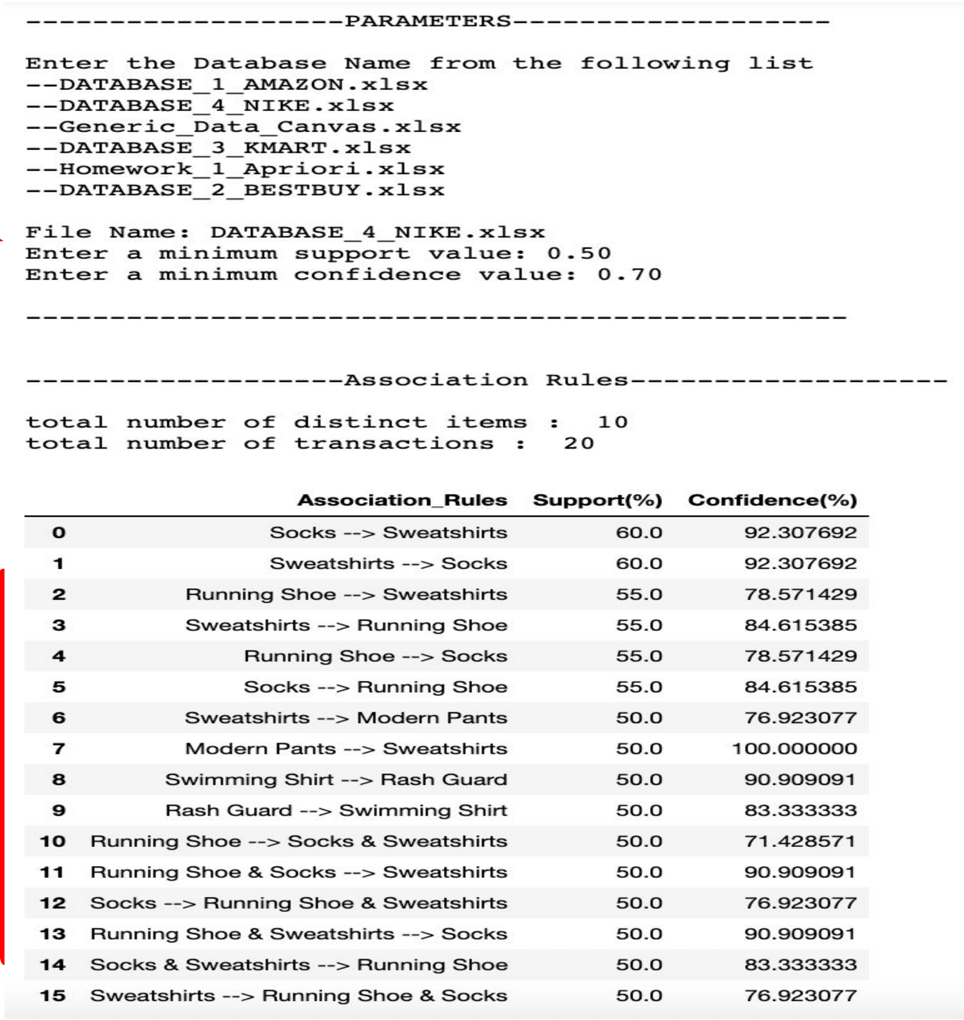
DATABASE 4: NIKE

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.70** (70%)

Dataset	**NIKE**
Source	**Canvas**
File Name	**DATABASE_4_NIKE.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.70**

Input Parameters

OUTPUT



```
-------------------PARAMETERS-------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_4_NIKE.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.70

-------------------------------------------------


-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

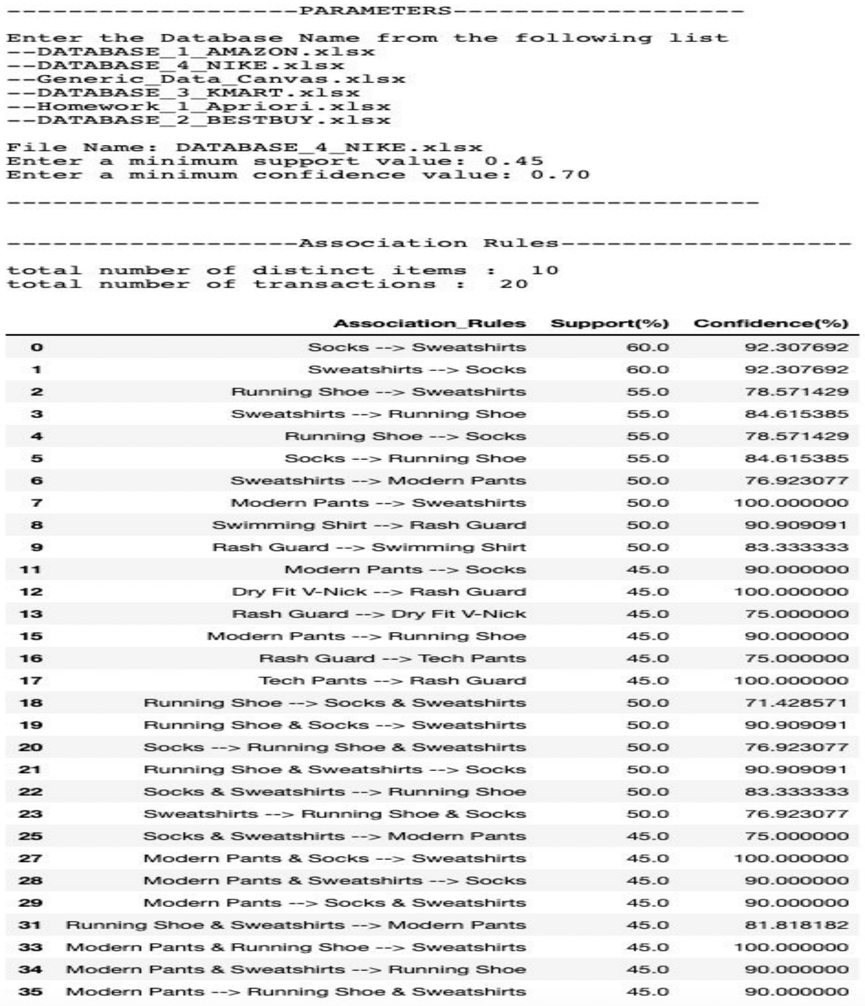
	Association_Rules	Support(%)	Confidence(%)
0	Socks --> Sweatshirts	60.0	92.307692
1	Sweatshirts --> Socks	60.0	92.307692
2	Running Shoe --> Sweatshirts	55.0	78.571429
3	Sweatshirts --> Running Shoe	55.0	84.615385
4	Running Shoe --> Socks	55.0	78.571429
5	Socks --> Running Shoe	55.0	84.615385
6	Sweatshirts --> Modern Pants	50.0	76.923077
7	Modern Pants --> Sweatshirts	50.0	100.000000
8	Swimming Shirt --> Rash Guard	50.0	90.909091
9	Rash Guard --> Swimming Shirt	50.0	83.333333
10	Running Shoe --> Socks & Sweatshirts	50.0	71.428571
11	Running Shoe & Socks --> Sweatshirts	50.0	90.909091
12	Socks --> Running Shoe & Sweatshirts	50.0	76.923077
13	Running Shoe & Sweatshirts --> Socks	50.0	90.909091
14	Socks & Sweatshirts --> Running Shoe	50.0	83.333333
15	Sweatshirts --> Running Shoe & Socks	50.0	76.923077

Minimum Support : **0.45** (45%)
Minimum Confidence : **0.70** (70%)

Dataset	**NIKE**
Source	**Canvas**
File Name	**DATABASE_4_NIKE.xlsx**
Minimum Support	**0.45**
Minimum Confidence	**0.70**

Input Parameters

OUTPUT



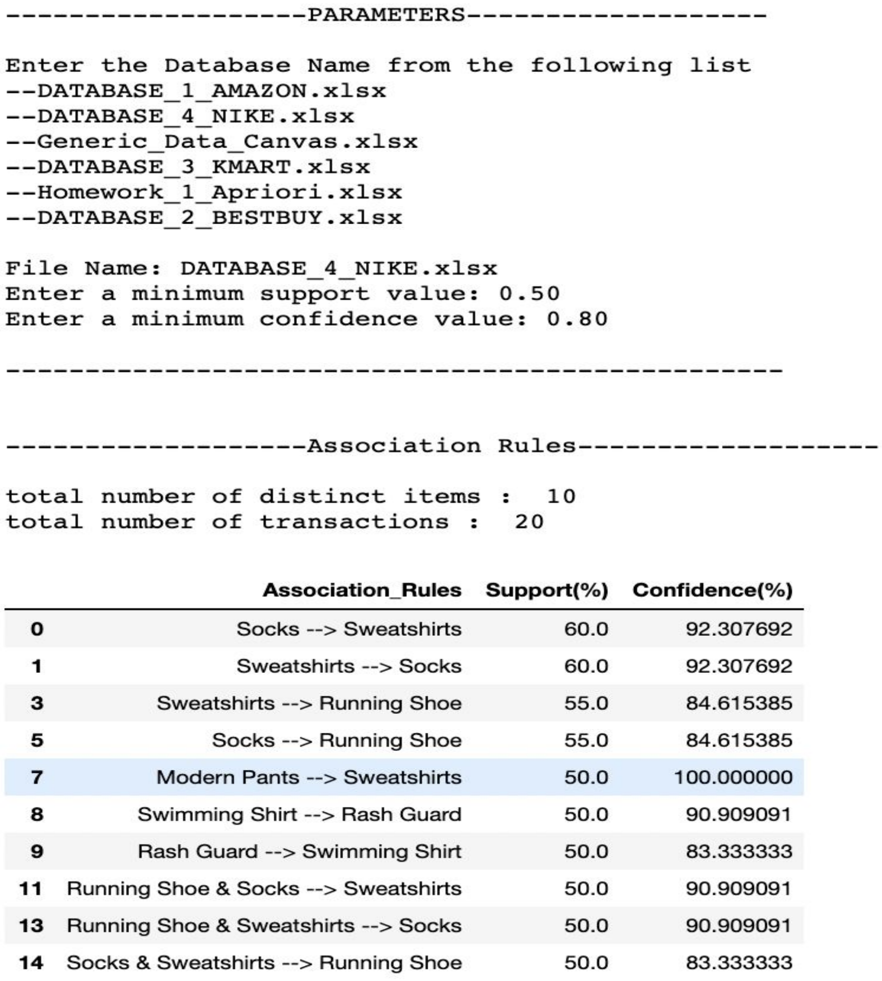
```
--------------------PARAMETERS-------------------
Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_4_NIKE.xlsx
Enter a minimum support value: 0.45
Enter a minimum confidence value: 0.70

-------------------------------------------------

-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20

                            Association_Rules  Support(%)  Confidence(%)
 0                       Socks --> Sweatshirts        60.0      92.307692
 1                       Sweatshirts --> Socks        60.0      92.307692
 2                Running Shoe --> Sweatshirts        55.0      78.571429
 3                Sweatshirts --> Running Shoe        55.0      84.615385
 4                      Running Shoe --> Socks        55.0      78.571429
 5                      Socks --> Running Shoe        55.0      84.615385
 6               Sweatshirts --> Modern Pants         50.0      76.923077
 7               Modern Pants --> Sweatshirts         50.0     100.000000
 8              Swimming Shirt --> Rash Guard         50.0      90.909091
 9              Rash Guard --> Swimming Shirt         50.0      83.333333
11                     Modern Pants --> Socks         45.0      90.000000
12              Dry Fit V-Nick --> Rash Guard         45.0     100.000000
13              Rash Guard --> Dry Fit V-Nick         45.0      75.000000
15              Modern Pants --> Running Shoe         45.0      90.000000
16                  Rash Guard --> Tech Pants         45.0      75.000000
17                  Tech Pants --> Rash Guard         45.0     100.000000
18      Running Shoe --> Socks & Sweatshirts          50.0      71.428571
19      Running Shoe & Socks --> Sweatshirts          50.0      90.909091
20      Socks --> Running Shoe & Sweatshirts          50.0      76.923077
21      Running Shoe & Sweatshirts --> Socks          50.0      90.909091
22      Socks & Sweatshirts --> Running Shoe          50.0      83.333333
23      Sweatshirts --> Running Shoe & Socks          50.0      76.923077
25      Socks & Sweatshirts --> Modern Pants          45.0      75.000000
27      Modern Pants & Socks --> Sweatshirts          45.0     100.000000
28      Modern Pants & Sweatshirts --> Socks          45.0      90.000000
29      Modern Pants --> Socks & Sweatshirts          45.0      90.000000
31  Running Shoe & Sweatshirts --> Modern Pants       45.0      81.818182
33  Modern Pants & Running Shoe --> Sweatshirts       45.0     100.000000
34  Modern Pants & Sweatshirts --> Running Shoe       45.0      90.000000
35  Modern Pants --> Running Shoe & Sweatshirts       45.0      90.000000
```

Minimum Support : **0.50** (50%)
Minimum Confidence : **0.80** (80%)

Dataset	**NIKE**
Source	**Canvas**
File Name	**DATABASE_4_NIKE.xlsx**
Minimum Support	**0.50**
Minimum Confidence	**0.80**

Input Parameters

OUTPUT



```
-------------------PARAMETERS------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: DATABASE_4_NIKE.xlsx
Enter a minimum support value: 0.50
Enter a minimum confidence value: 0.80

-------------------------------------------------


-------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	Socks --> Sweatshirts	60.0	92.307692
1	Sweatshirts --> Socks	60.0	92.307692
3	Sweatshirts --> Running Shoe	55.0	84.615385
5	Socks --> Running Shoe	55.0	84.615385
7	Modern Pants --> Sweatshirts	50.0	100.000000
8	Swimming Shirt --> Rash Guard	50.0	90.909091
9	Rash Guard --> Swimming Shirt	50.0	83.333333
11	Running Shoe & Socks --> Sweatshirts	50.0	90.909091
13	Running Shoe & Sweatshirts --> Socks	50.0	90.909091
14	Socks & Sweatshirts --> Running Shoe	50.0	83.333333

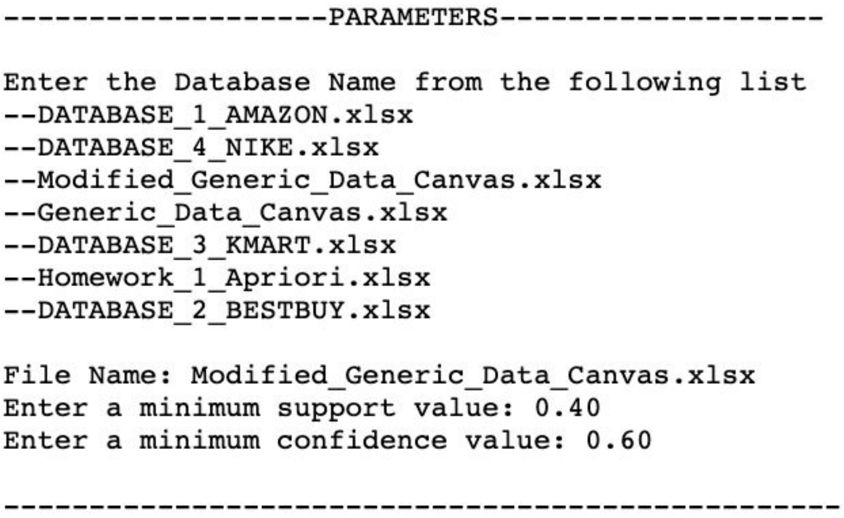
DATABASE 5: Canvas Generic Data

(Modified to make 10 Items and 20 Transactions)

Minimum Support : **0.40** (40%)
Minimum Confidence : **0.60** (60%)

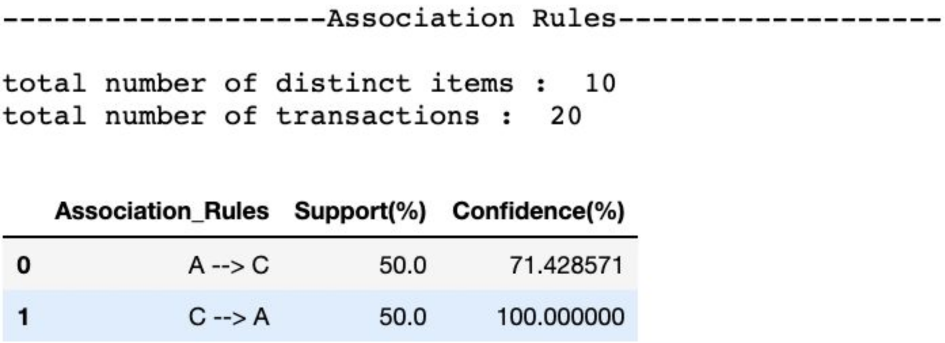
Dataset	**Canvas Generic Data**
Source	**Canvas**
File Name	**Modified_Generic_Data_Canvas.xlsx**
Minimum Support	**0.40**
Minimum Confidence	**0.60**

Input Parameters



```
-------------------PARAMETERS-------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Modified_Generic_Data_Canvas.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: Modified_Generic_Data_Canvas.xlsx
Enter a minimum support value: 0.40
Enter a minimum confidence value: 0.60

-------------------------------------------------
```

OUTPUT



```
-------------------Association Rules-------------------

total number of distinct items :  10
total number of transactions :  20
```

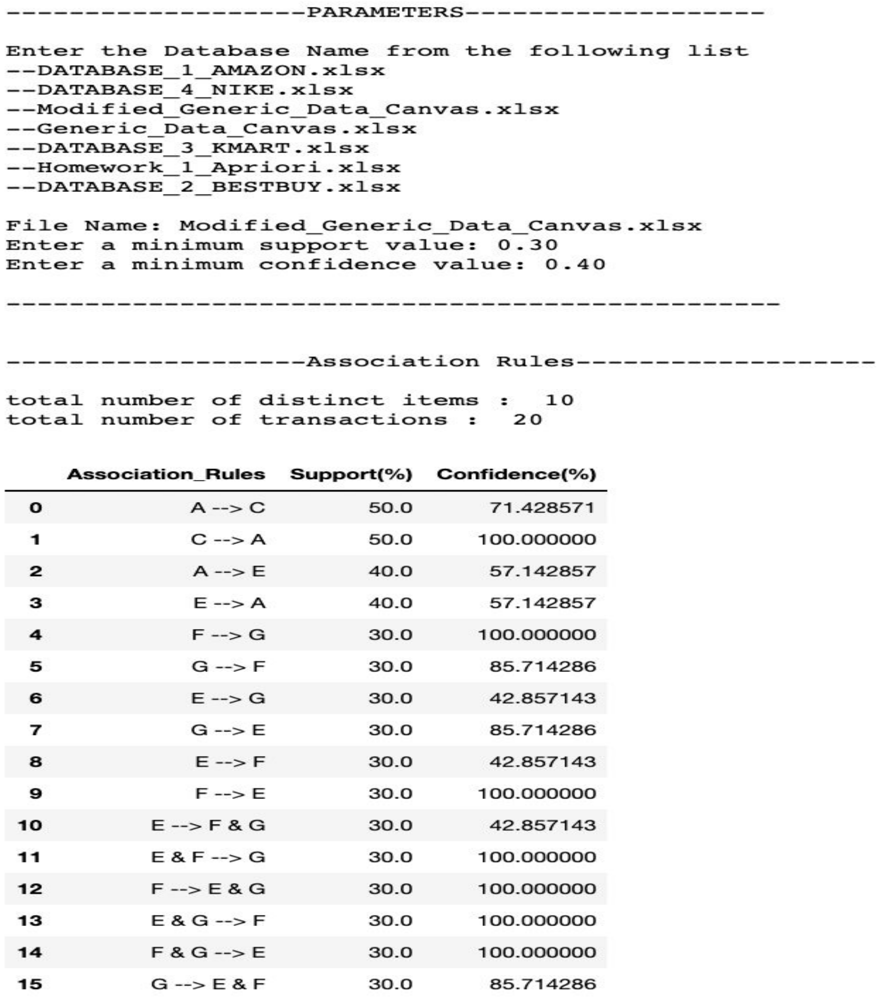
	Association_Rules	Support(%)	Confidence(%)
0	A --> C	50.0	71.428571
1	C --> A	50.0	100.000000

Minimum Support : **0.30** (30%)
Minimum Confidence : **0.40** (40%)

Dataset	**Canvas Generic Data**
Source	**Canvas**
File Name	**Modified_Generic_Data_Canvas.xlsx**
Minimum Support	**0.30**
Minimum Confidence	**0.40**

Input Parameters

OUTPUT



```
-------------------PARAMETERS-------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Modified_Generic_Data_Canvas.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: Modified_Generic_Data_Canvas.xlsx
Enter a minimum support value: 0.30
Enter a minimum confidence value: 0.40

--------------------------------------------------


------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

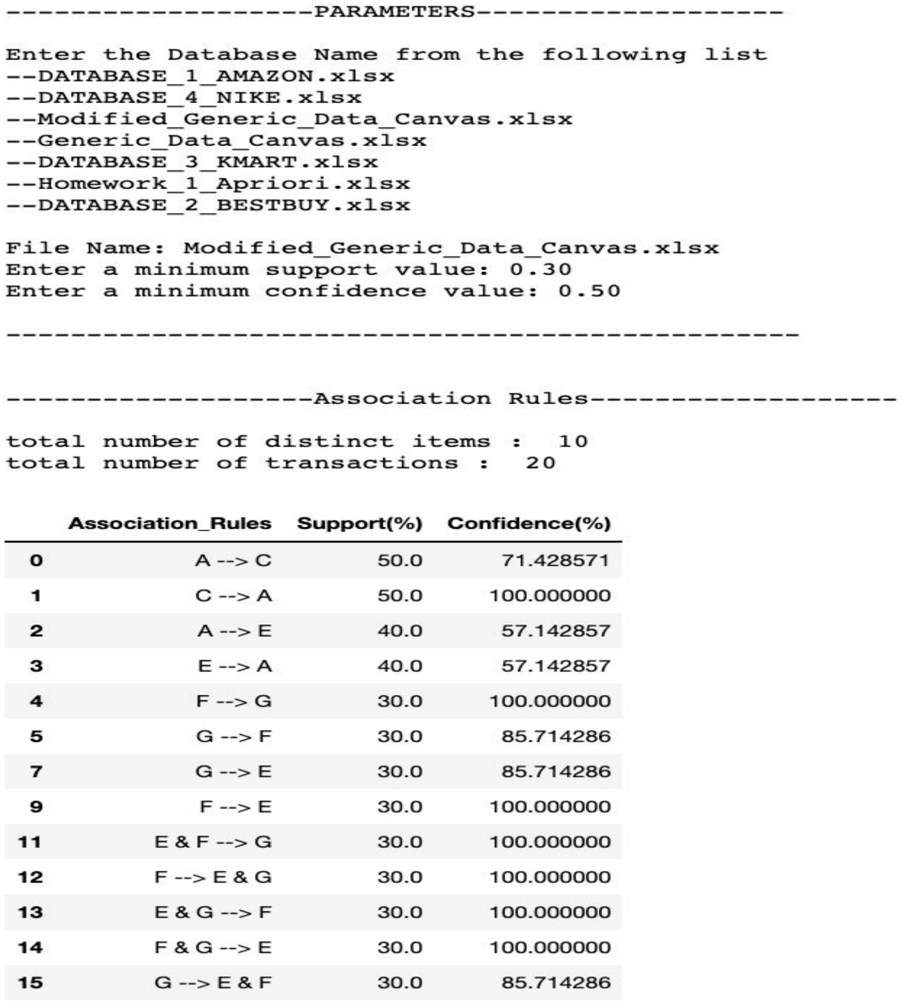
	Association_Rules	Support(%)	Confidence(%)
0	A --> C	50.0	71.428571
1	C --> A	50.0	100.000000
2	A --> E	40.0	57.142857
3	E --> A	40.0	57.142857
4	F --> G	30.0	100.000000
5	G --> F	30.0	85.714286
6	E --> G	30.0	42.857143
7	G --> E	30.0	85.714286
8	E --> F	30.0	42.857143
9	F --> E	30.0	100.000000
10	E --> F & G	30.0	42.857143
11	E & F --> G	30.0	100.000000
12	F --> E & G	30.0	100.000000
13	E & G --> F	30.0	100.000000
14	F & G --> E	30.0	100.000000
15	G --> E & F	30.0	85.714286

Minimum Support : **0.30** (30%)
Minimum Confidence : **0.50** (50%)

Dataset	**Canvas Generic Data**
Source	**Canvas**
File Name	**Modified_Generic_Data_Canvas.xlsx**
Minimum Support	**0.30**
Minimum Confidence	**0.50**

Input Parameters

OUTPUT



```
-------------------PARAMETERS-------------------

Enter the Database Name from the following list
--DATABASE_1_AMAZON.xlsx
--DATABASE_4_NIKE.xlsx
--Modified_Generic_Data_Canvas.xlsx
--Generic_Data_Canvas.xlsx
--DATABASE_3_KMART.xlsx
--Homework_1_Apriori.xlsx
--DATABASE_2_BESTBUY.xlsx

File Name: Modified_Generic_Data_Canvas.xlsx
Enter a minimum support value: 0.30
Enter a minimum confidence value: 0.50

-------------------------------------------------


------------------Association Rules------------------

total number of distinct items :  10
total number of transactions :  20
```

	Association_Rules	Support(%)	Confidence(%)
0	A --> C	50.0	71.428571
1	C --> A	50.0	100.000000
2	A --> E	40.0	57.142857
3	E --> A	40.0	57.142857
4	F --> G	30.0	100.000000
5	G --> F	30.0	85.714286
7	G --> E	30.0	85.714286
9	F --> E	30.0	100.000000
11	E & F --> G	30.0	100.000000
12	F --> E & G	30.0	100.000000
13	E & G --> F	30.0	100.000000
14	F & G --> E	30.0	100.000000
15	G --> E & F	30.0	85.714286

End