Data Mining

Midterm Project

Apriori Algorithm

Submitted by - Parth Sharma (UCID - 31543562) Mail - ps75@njit.edu

GitHub Link - https://github.com/parth426/Data-Mining.git

How to run: - Jupyter Notebook will be the best way to run this program. This program uses the name and the address of the CSV file which is present in my current system. To avoid error, make sure you have CSV file with same name and path, or you can change the attributes in the program accordingly.

About the Program: This program shows the comparison between the 2 approaches which are used to find out the frequent item set. This program takes the input from user at the run time input such as

- **Minimum support**
- **Minimum Confidence**

Code and the output is given below.

Importing Libraries

In [1]:

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

In [2]:

```
i = int(input('Enter any desired value '))
if i == 1:
    data = pd.read_csv('GroceryStoreDataSet.csv')
if i ==2:
    data = pd.read_csv('nike.csv')
else:
    data = pd.read_csv('flipkart.csv')
```

```
Enter any desired value 1
```

Dataset 1

Note- (i=1 for dataset 1)

In [3]:

```
data.head()
```

Out[3]:

Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9
0	Desk Top	Flash Drive	Microsoft Office	Speakers	Anti-Virus	NaN	NaN	NaN	NaN
1	Lab Top	Flash Drive	Microsoft Office	Anti-Virus	Lab Top Case	External Hard-Drive	NaN	NaN	NaN
2	Lab Top	Flash Drive	Anti-Virus	External Hard-Drive	Lab Top Case	NaN	NaN	NaN	NaN
3	Lab Top	Lab Top Case	Anti-Virus	NaN	NaN	NaN	NaN	NaN	NaN
4	Lab Top	Flash Drive	Microsoft Office	NaN	NaN	NaN	NaN	NaN	NaN

In [4]:

```
data.tail()
```

Out[4]:

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9
14	Digital Camera	Microsoft Office	Anti-Virus	Lab Top Case	External Hard-Drive	Speakers	NaN	NaN	NaN
15	Digital Camera	Lab Top Case	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	Digital Camera	Speakers	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	Digital Camera	Printer	Flash Drive	Microsoft Office	Speakers	Lab Top Case	Anti-Virus	NaN	NaN
18	Digital Camera	Speakers	Anti-Virus	Lab Top Case	NaN	NaN	NaN	NaN	NaN

In [5]:

```
data.shape
```

Out[5]:

```
(19, 9)
```

List of all unique items in each transaction

In [6]:

```
for i in range(data.shape[0]):
    print(f'List of Unique Items in Transaction : {i} ')
    print(data.iloc[i].dropna().unique(),' \n')
```

```
List of Unique Items in Transaction : 0
[' Desk Top' ' Flash Drive' ' Microsoft Office' ' Speakers' ' Anti-Virus']

List of Unique Items in Transaction : 1
[' Lab Top' ' Flash Drive' ' Microsoft Office' ' Anti-Virus'
 ' Lab Top Case' ' External Hard-Drive']

List of Unique Items in Transaction : 2
[' Lab Top' ' Flash Drive' ' Anti-Virus' ' External Hard-Drive'
 ' Lab Top Case']

List of Unique Items in Transaction : 3
[' Lab Top' ' Lab Top Case' ' Anti-Virus']

List of Unique Items in Transaction : 4
[' Lab Top' ' Flash Drive' ' Microsoft Office']

List of Unique Items in Transaction : 5
[' Desk Top' ' Flash Drive' ' Microsoft Office']

List of Unique Items in Transaction : 6
[' Lab Top' ' Anti-Virus']

List of Unique Items in Transaction : 7
[' Desk Top' ' Flash Drive' ' Microsoft Office' ' Lab Top Case'
```

```
[' Desk Top' ' Flash Drive' ' Microsoft Office' ' Lab Top Case'
 ' Anti-Virus' ' Speakers' ' External Hard-Drive']

List of Unique Items in Transaction : 8
[' Digital Camera ' ' Desk Top' ' Printer' ' Flash Drive'
 ' Microsoft Office' ' Lab Top Case' ' Anti-Virus' ' External Hard-Drive'
 ' Speakers']

List of Unique Items in Transaction : 9
[' Lab Top' ' Lab Top Case' ' External Hard-Drive' ' Speakers'
 ' Anti-Virus']

List of Unique Items in Transaction : 10
[' Digital Camera ' ' Lab Top Case' ' External Hard-Drive' ' Anti-Virus'
 ' Speakers']

List of Unique Items in Transaction : 11
[' Digital Camera ']

List of Unique Items in Transaction : 12
[' Digital Camera ' ' Printer' ' Flash Drive' ' Microsoft Office']

List of Unique Items in Transaction : 13
[' Printer' ' Microsoft Office' ' Anti-Virus' ' Lab Top Case' ' Speakers'
 ' External Hard-Drive']

List of Unique Items in Transaction : 14
[' Digital Camera' ' Microsoft Office' ' Anti-Virus' ' Lab Top Case'
 ' External Hard-Drive' ' Speakers']

List of Unique Items in Transaction : 15
[' Digital Camera ' ' Lab Top Case']

List of Unique Items in Transaction : 16
[' Digital Camera ' ' Speakers']

List of Unique Items in Transaction : 17
[' Digital Camera ' ' Printer' ' Flash Drive' ' Microsoft Office'
 ' Speakers' ' Lab Top Case' ' Anti-Virus']

List of Unique Items in Transaction : 18
[' Digital Camera ' ' Speakers' ' Anti-Virus' ' Lab Top Case']
```

In [7]:

```
print(f'Total no. of transactions in this Nike set {data.shape[0]}')
```

```
Total no. of transactions in this Nike set 19
```

In [8]:

```
minimum_s_count = float(input('enter any value for min support count '))
```

```
enter any value for min support count 5
```

In [9]:

```
minimum_confi = float(input('Enter any desired value for confidence '))
```

```
Enter any desired value for confidence 60
```

In [10]:

```
records = []
for i in range(0, data.shape[0]):
    records.append([str(data.values[i,j]) for j in range(0, data.shape[1])])

items = sorted([item for sublist in records for item in sublist if item != 'nan'])
```

List of Unique itemsets in this dataset

In [11]:

```
set(items)
```

Out[11]:

```
{' Anti-Virus',
 ' Desk Top',
 ' Digital Camera',
 ' Digital Camera ',
 ' External Hard-Drive',
 ' Flash Drive',
 ' Lab Top',
 ' Lab Top Case',
 ' Microsoft Office',
 ' Printer',
 ' Speakers'}
```

In [12]:

```
# To get a single item with their overall count in all transactions given.

def stage_1(items, minimum_s_count):
    c1 = {i:items.count(i) for i in items}
    l1 = {}
    for key, value in c1.items():
        if value >= minimum_s_count:
            l1[key] = value

    return c1, l1
```

In [13]:

```
# To get sets of 2 items with their overall count in all transactions given.
def stage_2(l1, records, minimum_s_count):
    l1 = sorted(list(l1.keys()))
    L1 = list(itertools.combinations(l1, 2))              # To make all possible combina
tions of 2 items
    c2 = {}
    l2 = {}
    for iter1 in L1:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c2[iter1] = count
    for key, value in c2.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l1, 1):
                l2[key] = value

    return c2, l2
```

In [14]:

```
# To get sets of 3 items with their overall count in all transactions given.
def stage_3(l2, records, minimum_s_count):
    l2 = list(l2.keys())
    L2 = sorted(list(set([item for t in l2 for item in t])))
    L2 = list(itertools.combinations(L2, 3))           # To make all possible combinatio
ns of 3 items
    c3 = {}
    l3 = {}
    for iter1 in L2:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c3[iter1] = count
    for key, value in c3.items():
```

```
        if value >= minimum_s_count:
            if check_subset_frequency(key, l2, 2):
                l3[key] = value

    return c3, l3
```

In [15]:

```
# To get sets of 4 items with their overall count in all transactions given.
def stage_4(l3, records, minimum_s_count):
    l3 = list(l3.keys())
    L3 = sorted(list(set([item for t in l3 for item in t])))
    L3 = list(itertools.combinations(L3, 4))              # To make all possible combinat
ions of 4 items
    c4 = {}
    l4 = {}
    for iter1 in L3:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c4[iter1] = count
    for key, value in c4.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l3, 3):
                l4[key] = value

    return c4, l4
```

In [16]:

```
def sublist(lst1, lst2):
    return set(lst1) <= set(lst2)

def check_subset_frequency(itemset, l, n):
    if n>1:
        subsets = list(itertools.combinations(itemset, n))
    else:
        subsets = itemset
    for iter1 in subsets:
        if not iter1 in l:
            return False
    return True
```

Below part prints all L1,L2,L3

All of them have pairs of 1 item, 2 items, 3 items respectively following minimum support decided by user above.

In [17]:

```
c1, l1 = stage_1(items, minimum_s_count)
c2, l2 = stage_2(l1, records, minimum_s_count)
c3, l3 = stage_3(l2, records, minimum_s_count)
c4, l4 = stage_4(l3, records, minimum_s_count)
```

In [18]:

```
for key, value in l1.items():
    print(key, ' : ', value)
```

```
Anti-Virus  :  13
Digital Camera  :  8
External Hard-Drive  :  8
Flash Drive  :  9
Lab Top  :  6
Lab Top Case  :  12
Microsoft Office  :  10
Speakers  :  10
```

In [19]:

```
for key, value in l2.items():
    print(key, ' : ', value)
```

```
(' Anti-Virus', ' External Hard-Drive')  :  8
(' Anti-Virus', ' Flash Drive')  :  6
(' Anti-Virus', ' Lab Top')  :  5
(' Anti-Virus', ' Lab Top Case')  :  11
(' Anti-Virus', ' Microsoft Office')  :  7
(' Anti-Virus', ' Speakers')  :  9
(' Digital Camera ', ' Lab Top Case')  :  5
(' Digital Camera ', ' Speakers')  :  5
(' External Hard-Drive', ' Lab Top Case')  :  8
(' External Hard-Drive', ' Microsoft Office')  :  5
(' External Hard-Drive', ' Speakers')  :  6
(' Flash Drive', ' Lab Top Case')  :  5
(' Flash Drive', ' Microsoft Office')  :  8
(' Lab Top Case', ' Microsoft Office')  :  6
(' Lab Top Case', ' Speakers')  :  8
(' Microsoft Office', ' Speakers')  :  6
```

In [20]:

```
for key, value in l3.items():
    print(key, ' : ', value)
```

```
(' Anti-Virus', ' External Hard-Drive', ' Lab Top Case')  :  8
(' Anti-Virus', ' External Hard-Drive', ' Microsoft Office')  :  5
(' Anti-Virus', ' External Hard-Drive', ' Speakers')  :  6
(' Anti-Virus', ' Flash Drive', ' Lab Top Case')  :  5
(' Anti-Virus', ' Flash Drive', ' Microsoft Office')  :  5
(' Anti-Virus', ' Lab Top Case', ' Microsoft Office')  :  6
(' Anti-Virus', ' Lab Top Case', ' Speakers')  :  8
(' Anti-Virus', ' Microsoft Office', ' Speakers')  :  6
(' External Hard-Drive', ' Lab Top Case', ' Microsoft Office')  :  5
(' External Hard-Drive', ' Lab Top Case', ' Speakers')  :  6
(' Lab Top Case', ' Microsoft Office', ' Speakers')  :  5
```

Print Association Rules

(with their Minimum support & minimum confidence)

In [21]:

```
itemlist = {**l1, **l2, **l3, **l4}

def support_count(itemset, itemlist):
    return itemlist[itemset]

def print_sets():
    print('Possible sets with just 1 item ',l1)

sets = []
for iter1 in list(l3.keys()):
    subsets = list(itertools.combinations(iter1, 2))
    sets.append(subsets)

list_l3 = list(l3.keys())
print('ASSOCIATION RULES : '+ '\n')
for i in range(0, len(list_l3)):
    for iter1 in sets[i]:
        a = iter1
        b = set(list_l3[i]) - set(iter1)
        confidence = (support_count(list_l3[i], itemlist)/support_count(iter1, itemlist)
)*100
        support_i = support_count(iter1,itemlist)
        if confidence > minimum_confi:
            print("Confidence {}->{} = ".format(a,b), round(confidence,3),'%')
```

```
            print('Support{}->{} = '.format(a,b),round(support_i,3))
            print('\n')
#         print("Confidence{}->{} = ".format(a,b), confidence)
```

```
ASSOCIATION RULES :

Confidence (' Anti-Virus', ' External Hard-Drive')->{' Lab Top Case'} =  100.0 %
Support(' Anti-Virus', ' External Hard-Drive')->{' Lab Top Case'} =  8


Confidence (' Anti-Virus', ' Lab Top Case')->{' External Hard-Drive'} =  72.727 %
Support(' Anti-Virus', ' Lab Top Case')->{' External Hard-Drive'} =  11


Confidence (' External Hard-Drive', ' Lab Top Case')->{' Anti-Virus'} =  100.0 %
Support(' External Hard-Drive', ' Lab Top Case')->{' Anti-Virus'} =  8


Confidence (' Anti-Virus', ' External Hard-Drive')->{' Microsoft Office'} =  62.5 %
Support(' Anti-Virus', ' External Hard-Drive')->{' Microsoft Office'} =  8


Confidence (' Anti-Virus', ' Microsoft Office')->{' External Hard-Drive'} =  71.429 %
Support(' Anti-Virus', ' Microsoft Office')->{' External Hard-Drive'} =  7


Confidence (' External Hard-Drive', ' Microsoft Office')->{' Anti-Virus'} =  100.0 %
Support(' External Hard-Drive', ' Microsoft Office')->{' Anti-Virus'} =  5


Confidence (' Anti-Virus', ' External Hard-Drive')->{' Speakers'} =  75.0 %
Support(' Anti-Virus', ' External Hard-Drive')->{' Speakers'} =  8


Confidence (' Anti-Virus', ' Speakers')->{' External Hard-Drive'} =  66.667 %
Support(' Anti-Virus', ' Speakers')->{' External Hard-Drive'} =  9


Confidence (' External Hard-Drive', ' Speakers')->{' Anti-Virus'} =  100.0 %
Support(' External Hard-Drive', ' Speakers')->{' Anti-Virus'} =  6


Confidence (' Anti-Virus', ' Flash Drive')->{' Lab Top Case'} =  83.333 %
Support(' Anti-Virus', ' Flash Drive')->{' Lab Top Case'} =  6


Confidence (' Flash Drive', ' Lab Top Case')->{' Anti-Virus'} =  100.0 %
Support(' Flash Drive', ' Lab Top Case')->{' Anti-Virus'} =  5


Confidence (' Anti-Virus', ' Flash Drive')->{' Microsoft Office'} =  83.333 %
Support(' Anti-Virus', ' Flash Drive')->{' Microsoft Office'} =  6


Confidence (' Anti-Virus', ' Microsoft Office')->{' Flash Drive'} =  71.429 %
Support(' Anti-Virus', ' Microsoft Office')->{' Flash Drive'} =  7


Confidence (' Flash Drive', ' Microsoft Office')->{' Anti-Virus'} =  62.5 %
Support(' Flash Drive', ' Microsoft Office')->{' Anti-Virus'} =  8


Confidence (' Anti-Virus', ' Microsoft Office')->{' Lab Top Case'} =  85.714 %
Support(' Anti-Virus', ' Microsoft Office')->{' Lab Top Case'} =  7


Confidence (' Lab Top Case', ' Microsoft Office')->{' Anti-Virus'} =  100.0 %
Support(' Lab Top Case', ' Microsoft Office')->{' Anti-Virus'} =  6


Confidence (' Anti-Virus', ' Lab Top Case')->{' Speakers'} =  72.727 %
Support(' Anti-Virus', ' Lab Top Case')->{' Speakers'} =  11
```

```
Confidence (' Anti-Virus', ' Speakers')->{' Lab Top Case'} =  88.889 %
Support(' Anti-Virus', ' Speakers')->{' Lab Top Case'} =  9


Confidence (' Lab Top Case', ' Speakers')->{' Anti-Virus'} =  100.0 %
Support(' Lab Top Case', ' Speakers')->{' Anti-Virus'} =  8


Confidence (' Anti-Virus', ' Microsoft Office')->{' Speakers'} =  85.714 %
Support(' Anti-Virus', ' Microsoft Office')->{' Speakers'} =  7


Confidence (' Anti-Virus', ' Speakers')->{' Microsoft Office'} =  66.667 %
Support(' Anti-Virus', ' Speakers')->{' Microsoft Office'} =  9


Confidence (' Microsoft Office', ' Speakers')->{' Anti-Virus'} =  100.0 %
Support(' Microsoft Office', ' Speakers')->{' Anti-Virus'} =  6


Confidence (' External Hard-Drive', ' Lab Top Case')->{' Microsoft Office'} =  62.5 %
Support(' External Hard-Drive', ' Lab Top Case')->{' Microsoft Office'} =  8


Confidence (' External Hard-Drive', ' Microsoft Office')->{' Lab Top Case'} =  100.0 %
Support(' External Hard-Drive', ' Microsoft Office')->{' Lab Top Case'} =  5


Confidence (' Lab Top Case', ' Microsoft Office')->{' External Hard-Drive'} =  83.333 %
Support(' Lab Top Case', ' Microsoft Office')->{' External Hard-Drive'} =  6


Confidence (' External Hard-Drive', ' Lab Top Case')->{' Speakers'} =  75.0 %
Support(' External Hard-Drive', ' Lab Top Case')->{' Speakers'} =  8


Confidence (' External Hard-Drive', ' Speakers')->{' Lab Top Case'} =  100.0 %
Support(' External Hard-Drive', ' Speakers')->{' Lab Top Case'} =  6


Confidence (' Lab Top Case', ' Speakers')->{' External Hard-Drive'} =  75.0 %
Support(' Lab Top Case', ' Speakers')->{' External Hard-Drive'} =  8


Confidence (' Lab Top Case', ' Microsoft Office')->{' Speakers'} =  83.333 %
Support(' Lab Top Case', ' Microsoft Office')->{' Speakers'} =  6


Confidence (' Lab Top Case', ' Speakers')->{' Microsoft Office'} =  62.5 %
Support(' Lab Top Case', ' Speakers')->{' Microsoft Office'} =  8


Confidence (' Microsoft Office', ' Speakers')->{' Lab Top Case'} =  83.333 %
Support(' Microsoft Office', ' Speakers')->{' Lab Top Case'} =  6
```

Dataset 2

Note- (i=2 for dataset 2)

In [22]:

```
i = int(input('Enter any desired value '))
if i == 1:
    data = pd.read_csv('GroceryStoreDataSet.csv')
if i ==2:
    data = pd.read_csv('nike.csv')
```

```
else:
    data = pd.read_csv('flipkart.csv')
```

```
Enter any desired value 2
```

In [23]:

```
data.head()
```

Out[23]:

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
0	Running Shoe	Socks	Sweatshirts	Modern Pants	NaN	NaN	NaN	NaN	NaN	NaN
1	Running Shoe	Socks	Sweatshirts	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	Running Shoe	Socks	Sweatshirts	Modern Pants	NaN	NaN	NaN	NaN	NaN	NaN
3	Running Shoe	Sweatshirts	Modern Pants	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Running Shoe	Socks	Sweatshirts	Modern Pants	Soccer Shoe	NaN	NaN	NaN	NaN	NaN

In [24]:

```
data.tail()
```

Out[24]:

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

In [25]:

```
data.shape
```

Out[25]:

```
(20, 10)
```

List of all unique items in each transaction

In [26]:

```
for i in range(data.shape[0]):
    print(f'List of Unique Items in Transaction : {i} ')
    print(data.iloc[i].dropna().unique(),' \n')
```

```
List of Unique Items in Transaction : 0
['Running Shoe' ' Socks' ' Sweatshirts' ' Modern Pants']

List of Unique Items in Transaction : 1
['Running Shoe' ' Socks' ' Sweatshirts']

List of Unique Items in Transaction : 2
['Running Shoe' ' Socks' ' Sweatshirts' ' Modern Pants']

List of Unique Items in Transaction : 3
['Running Shoe' ' Sweatshirts' ' Modern Pants']
```

```
List of Unique Items in Transaction : 4
['Running Shoe' ' Socks' ' Sweatshirts' ' Modern Pants' ' Soccer Shoe ']

List of Unique Items in Transaction : 5
['Running Shoe' ' Socks' ' Sweatshirts']

List of Unique Items in Transaction : 6
['Running Shoe' ' Socks' ' Sweatshirts' ' Modern Pants' ' Tech Pants'
 ' Rash Guard' ' Hoodies']

List of Unique Items in Transaction : 7
['Swimming Shirt' ' Socks' ' Sweatshirts']

List of Unique Items in Transaction : 8
['Swimming Shirt' ' Rash Guard' ' Dry Fit V-Nick' ' Hoodies' ' Tech Pants']

List of Unique Items in Transaction : 9
['Swimming Shirt' ' Rash Guard' ' Dry']

List of Unique Items in Transaction : 10
['Swimming Shirt' ' Rash Guard' ' Dry Fit V-Nick']

List of Unique Items in Transaction : 11
[' Running Shoe' ' Swimming Shirt' ' Socks' ' Sweatshirts' ' Modern Pants'
 ' Soccer Shoe' ' Rash Guard' ' Hoodies' ' Tech Pants' ' Dry Fit V-Nick']

List of Unique Items in Transaction : 12
['Running Shoe' ' Swimming Shirt' ' Socks' ' Sweatshirts' ' Modern Pants'
 ' Soccer Shoe' ' Rash Guard' ' Tech Pants' ' Dry Fit V-Nick' ' Hoodies']

List of Unique Items in Transaction : 13
['Running Shoe' ' Swimming Shirt' ' Rash Guard' ' Tech Pants' ' Hoodies'
 ' Dry Fit V-Nick']

List of Unique Items in Transaction : 14
['Running Shoe' ' Swimming Shirt' ' Socks' ' Sweatshirts' ' Modern Pants'
 ' Dry Fit V-Nick' ' Rash Guard' ' Tech Pants']

List of Unique Items in Transaction : 15
['Swimming Shirt' ' Soccer Shoe' ' Hoodies' ' Dry Fit V-Nick'
 ' Tech Pants' ' Rash Guard']

List of Unique Items in Transaction : 16
['Running Shoe' ' Socks']

List of Unique Items in Transaction : 17
['Socks' ' Sweatshirts' ' Modern Pants' ' Soccer Shoe' ' Hoodies'
 ' Rash Guard' ' Tech Pants' ' Dry Fit V-Nick']

List of Unique Items in Transaction : 18
['Running Shoe' ' Swimming Shirt' ' Rash Guard']

List of Unique Items in Transaction : 19
['Running Shoe' ' Swimming Shirt' ' Socks' ' Sweatshirts' ' Modern Pants'
 ' Soccer Shoe' ' Hoodies' ' Tech Pants' ' Rash Guard' ' Dry Fit V-Nick']
```

In [27]:

```
print(f'Total no. of transactions in this Nike set {data.shape[0]}')
```

```
Total no. of transactions in this Nike set 20
```

In [28]:

```
minimum_s_count = float(input('enter any value for min support count '))
```

```
enter any value for min support count 5
```

In [29]:

```
minimum_conf1 = float(input('Enter any desired value for confidence '))
```

```
Enter any desired value for confidence 54
```

In [30]:

```
records = []
for i in range(0, data.shape[0]):
    records.append([str(data.values[i,j]) for j in range(0, data.shape[1])])

items = sorted([item for sublist in records for item in sublist if item != 'nan'])
```

List of Unique itemsets in this dataset

In [31]:

```
set(items)
```

Out[31]:

```
{' Dry',
 ' Dry Fit V-Nick',
 ' Hoodies',
 ' Modern Pants',
 ' Rash Guard',
 ' Running Shoe',
 ' Soccer Shoe',
 ' Soccer Shoe ',
 ' Socks',
 ' Sweatshirts',
 ' Swimming Shirt',
 ' Tech Pants',
 'Running Shoe',
 'Socks',
 'Swimming Shirt'}
```

In [32]:

```
# To get a single item with their overall count in all transactions given.

def stage_1(items, minimum_s_count):
    c1 = {i:items.count(i) for i in items}
    l1 = {}
    for key, value in c1.items():
        if value >= minimum_s_count:
            l1[key] = value

    return c1, l1
```

In [33]:

```
# To get sets of 2 items with their overall count in all transactions given.
def stage_2(l1, records, minimum_s_count):
    l1 = sorted(list(l1.keys()))
    L1 = list(itertools.combinations(l1, 2))              # To make all possible combina
tions of 2 items
    c2 = {}
    l2 = {}
    for iter1 in L1:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c2[iter1] = count
    for key, value in c2.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l1, 1):
                l2[key] = value

    return c2, l2
```

In [34]:

```
# To get sets of 3 items with their overall count in all transactions given.
def stage_3(l2, records, minimum_s_count):
    l2 = list(l2.keys())
    L2 = sorted(list(set([item for t in l2 for item in t])))
    L2 = list(itertools.combinations(L2, 3))           # To make all possible combinatio
ns of 3 items
    c3 = {}
    l3 = {}
    for iter1 in L2:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c3[iter1] = count
    for key, value in c3.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l2, 2):
                l3[key] = value

    return c3, l3
```

In [35]:

```
# To get sets of 4 items with their overall count in all transactions given.
def stage_4(l3, records, minimum_s_count):
    l3 = list(l3.keys())
    L3 = sorted(list(set([item for t in l3 for item in t])))
    L3 = list(itertools.combinations(L3, 4))             # To make all possible combinat
ions of 4 items
    c4 = {}
    l4 = {}
    for iter1 in L3:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c4[iter1] = count
    for key, value in c4.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l3, 3):
                l4[key] = value

    return c4, l4
```

In [36]:

```
def sublist(lst1, lst2):
    return set(lst1) <= set(lst2)

def check_subset_frequency(itemset, l, n):
    if n>1:
        subsets = list(itertools.combinations(itemset, n))
    else:
        subsets = itemset
    for iter1 in subsets:
        if not iter1 in l:
            return False
    return True
```

Below part prints all L1,L2,L3

All of them have pairs of 1 item, 2 items, 3 items respectively following minimum support decided by user above

In [37]:

```
c1, l1 = stage_1(items, minimum_s_count)
```

```
c2, l2 = stage_2(l1, records, minimum_s_count)
c3, l3 = stage_3(l2, records, minimum_s_count)
c4, l4 = stage_4(l3, records, minimum_s_count)
```

In [38]:

```
for key, value in l1.items():
    print(key, ' : ', value)
```

```
 Dry Fit V-Nick  :  9
 Hoodies  :  8
 Modern Pants  :  10
 Rash Guard  :  12
 Soccer Shoe  :  5
 Socks  :  12
 Sweatshirts  :  13
 Swimming Shirt  :  6
 Tech Pants  :  9
Running Shoe  :  13
Swimming Shirt  :  5
```

In [39]:

```
for key, value in l2.items():
    print(key, ' : ', value)
```

```
(' Dry Fit V-Nick', ' Hoodies')  :  7
(' Dry Fit V-Nick', ' Modern Pants')  :  5
(' Dry Fit V-Nick', ' Rash Guard')  :  9
(' Dry Fit V-Nick', ' Soccer Shoe')  :  5
(' Dry Fit V-Nick', ' Sweatshirts')  :  5
(' Dry Fit V-Nick', ' Swimming Shirt')  :  5
(' Dry Fit V-Nick', ' Tech Pants')  :  8
(' Hoodies', ' Modern Pants')  :  5
(' Hoodies', ' Rash Guard')  :  8
(' Hoodies', ' Soccer Shoe')  :  5
(' Hoodies', ' Sweatshirts')  :  5
(' Hoodies', ' Tech Pants')  :  8
(' Modern Pants', ' Rash Guard')  :  6
(' Modern Pants', ' Socks')  :  8
(' Modern Pants', ' Sweatshirts')  :  10
(' Modern Pants', ' Tech Pants')  :  6
(' Modern Pants', 'Running Shoe')  :  8
(' Rash Guard', ' Soccer Shoe')  :  5
(' Rash Guard', ' Socks')  :  5
(' Rash Guard', ' Sweatshirts')  :  6
(' Rash Guard', ' Swimming Shirt')  :  6
(' Rash Guard', ' Tech Pants')  :  9
(' Rash Guard', 'Running Shoe')  :  6
(' Soccer Shoe', ' Tech Pants')  :  5
(' Socks', ' Sweatshirts')  :  11
(' Socks', ' Tech Pants')  :  5
(' Socks', 'Running Shoe')  :  10
(' Sweatshirts', ' Tech Pants')  :  6
(' Sweatshirts', 'Running Shoe')  :  10
(' Swimming Shirt', ' Tech Pants')  :  5
(' Swimming Shirt', 'Running Shoe')  :  5
(' Tech Pants', 'Running Shoe')  :  5
```

In [40]:

```
for key, value in l3.items():
    print(key, ' : ', value)
```

```
(' Dry Fit V-Nick', ' Hoodies', ' Rash Guard')  :  7
(' Dry Fit V-Nick', ' Hoodies', ' Soccer Shoe')  :  5
(' Dry Fit V-Nick', ' Hoodies', ' Tech Pants')  :  7
(' Dry Fit V-Nick', ' Modern Pants', ' Rash Guard')  :  5
(' Dry Fit V-Nick', ' Modern Pants', ' Sweatshirts')  :  5
(' Dry Fit V-Nick', ' Modern Pants', ' Tech Pants')  :  5
(' Dry Fit V-Nick', ' Rash Guard', ' Soccer Shoe')  :  5
(' Dry Fit V-Nick', ' Rash Guard', ' Sweatshirts')  :  5
```

```
(' Dry Fit V-Nick', ' Rash Guard', ' Swimming Shirt')  :  5
(' Dry Fit V-Nick', ' Rash Guard', ' Tech Pants')  :  8
(' Dry Fit V-Nick', ' Soccer Shoe', ' Tech Pants')  :  5
(' Dry Fit V-Nick', ' Sweatshirts', ' Tech Pants')  :  5
(' Dry Fit V-Nick', ' Swimming Shirt', ' Tech Pants')  :  5
(' Hoodies', ' Modern Pants', ' Rash Guard')  :  5
(' Hoodies', ' Modern Pants', ' Sweatshirts')  :  5
(' Hoodies', ' Modern Pants', ' Tech Pants')  :  5
(' Hoodies', ' Rash Guard', ' Soccer Shoe')  :  5
(' Hoodies', ' Rash Guard', ' Sweatshirts')  :  5
(' Hoodies', ' Rash Guard', ' Tech Pants')  :  8
(' Hoodies', ' Soccer Shoe', ' Tech Pants')  :  5
(' Hoodies', ' Sweatshirts', ' Tech Pants')  :  5
(' Modern Pants', ' Rash Guard', ' Socks')  :  5
(' Modern Pants', ' Rash Guard', ' Sweatshirts')  :  6
(' Modern Pants', ' Rash Guard', ' Tech Pants')  :  6
(' Modern Pants', ' Socks', ' Sweatshirts')  :  8
(' Modern Pants', ' Socks', ' Tech Pants')  :  5
(' Modern Pants', ' Socks', 'Running Shoe')  :  7
(' Modern Pants', ' Sweatshirts', ' Tech Pants')  :  6
(' Modern Pants', ' Sweatshirts', 'Running Shoe')  :  8
(' Rash Guard', ' Soccer Shoe', ' Tech Pants')  :  5
(' Rash Guard', ' Socks', ' Sweatshirts')  :  5
(' Rash Guard', ' Socks', ' Tech Pants')  :  5
(' Rash Guard', ' Sweatshirts', ' Tech Pants')  :  6
(' Rash Guard', ' Swimming Shirt', ' Tech Pants')  :  5
(' Rash Guard', ' Swimming Shirt', 'Running Shoe')  :  5
(' Rash Guard', ' Tech Pants', 'Running Shoe')  :  5
(' Socks', ' Sweatshirts', ' Tech Pants')  :  5
(' Socks', ' Sweatshirts', 'Running Shoe')  :  9
```

Print Association Rules

(with their Minimum support & minimum confidence)

In [41]:

```
itemlist = {**l1, **l2, **l3, **l4}

def support_count(itemset, itemlist):
    return itemlist[itemset]

def print_sets():
    print('Possible sets with just 1 item ',l1)

sets = []
for iter1 in list(l3.keys()):
    subsets = list(itertools.combinations(iter1, 2))
    sets.append(subsets)

list_l3 = list(l3.keys())
print('ASSOCIATION RULES : '+ '\n')
for i in range(0, len(list_l3)):
    for iter1 in sets[i]:
        a = iter1
        b = set(list_l3[i]) - set(iter1)
        confidence = (support_count(list_l3[i], itemlist)/support_count(iter1, itemlist)
)*100
        support_i = support_count(iter1,itemlist)
        if confidence > minimum_confi:
            print("Confidence {}->{} = ".format(a,b), round(confidence,3),'%')
            print('Support{}->{} = '.format(a,b),round(support_i,3))
            print('\n')
#         print("Confidence{}->{} = ".format(a,b), confidence)
```

```
ASSOCIATION RULES :

Confidence (' Dry Fit V-Nick', ' Hoodies')->{' Rash Guard'} =  100.0 %
Support(' Dry Fit V-Nick', ' Hoodies')->{' Rash Guard'} =  7
```

Confidence (' Dry Fit V-Nick', ' Rash Guard')->{' Hoodies'} = 77.778 %
Support(' Dry Fit V-Nick', ' Rash Guard')->{' Hoodies'} = 9

Confidence (' Hoodies', ' Rash Guard')->{' Dry Fit V-Nick'} = 87.5 %
Support(' Hoodies', ' Rash Guard')->{' Dry Fit V-Nick'} = 8

Confidence (' Dry Fit V-Nick', ' Hoodies')->{' Soccer Shoe'} = 71.429 %
Support(' Dry Fit V-Nick', ' Hoodies')->{' Soccer Shoe'} = 7

Confidence (' Dry Fit V-Nick', ' Soccer Shoe')->{' Hoodies'} = 100.0 %
Support(' Dry Fit V-Nick', ' Soccer Shoe')->{' Hoodies'} = 5

Confidence (' Hoodies', ' Soccer Shoe')->{' Dry Fit V-Nick'} = 100.0 %
Support(' Hoodies', ' Soccer Shoe')->{' Dry Fit V-Nick'} = 5

Confidence (' Dry Fit V-Nick', ' Hoodies')->{' Tech Pants'} = 100.0 %
Support(' Dry Fit V-Nick', ' Hoodies')->{' Tech Pants'} = 7

Confidence (' Dry Fit V-Nick', ' Tech Pants')->{' Hoodies'} = 87.5 %
Support(' Dry Fit V-Nick', ' Tech Pants')->{' Hoodies'} = 8

Confidence (' Hoodies', ' Tech Pants')->{' Dry Fit V-Nick'} = 87.5 %
Support(' Hoodies', ' Tech Pants')->{' Dry Fit V-Nick'} = 8

Confidence (' Dry Fit V-Nick', ' Modern Pants')->{' Rash Guard'} = 100.0 %
Support(' Dry Fit V-Nick', ' Modern Pants')->{' Rash Guard'} = 5

Confidence (' Dry Fit V-Nick', ' Rash Guard')->{' Modern Pants'} = 55.556 %
Support(' Dry Fit V-Nick', ' Rash Guard')->{' Modern Pants'} = 9

Confidence (' Modern Pants', ' Rash Guard')->{' Dry Fit V-Nick'} = 83.333 %
Support(' Modern Pants', ' Rash Guard')->{' Dry Fit V-Nick'} = 6

Confidence (' Dry Fit V-Nick', ' Modern Pants')->{' Sweatshirts'} = 100.0 %
Support(' Dry Fit V-Nick', ' Modern Pants')->{' Sweatshirts'} = 5

Confidence (' Dry Fit V-Nick', ' Sweatshirts')->{' Modern Pants'} = 100.0 %
Support(' Dry Fit V-Nick', ' Sweatshirts')->{' Modern Pants'} = 5

Confidence (' Dry Fit V-Nick', ' Modern Pants')->{' Tech Pants'} = 100.0 %
Support(' Dry Fit V-Nick', ' Modern Pants')->{' Tech Pants'} = 5

Confidence (' Dry Fit V-Nick', ' Tech Pants')->{' Modern Pants'} = 62.5 %
Support(' Dry Fit V-Nick', ' Tech Pants')->{' Modern Pants'} = 8

Confidence (' Modern Pants', ' Tech Pants')->{' Dry Fit V-Nick'} = 83.333 %
Support(' Modern Pants', ' Tech Pants')->{' Dry Fit V-Nick'} = 6

Confidence (' Dry Fit V-Nick', ' Rash Guard')->{' Soccer Shoe'} = 55.556 %
Support(' Dry Fit V-Nick', ' Rash Guard')->{' Soccer Shoe'} = 9

Confidence (' Dry Fit V-Nick', ' Soccer Shoe')->{' Rash Guard'} = 100.0 %
Support(' Dry Fit V-Nick', ' Soccer Shoe')->{' Rash Guard'} = 5

```
Confidence (' Rash Guard', ' Soccer Shoe')->{' Dry Fit V-Nick'} =  100.0 %
Support(' Rash Guard', ' Soccer Shoe')->{' Dry Fit V-Nick'} =  5


Confidence (' Dry Fit V-Nick', ' Rash Guard')->{' Sweatshirts'} =  55.556 %
Support(' Dry Fit V-Nick', ' Rash Guard')->{' Sweatshirts'} =  9


Confidence (' Dry Fit V-Nick', ' Sweatshirts')->{' Rash Guard'} =  100.0 %
Support(' Dry Fit V-Nick', ' Sweatshirts')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Sweatshirts')->{' Dry Fit V-Nick'} =  83.333 %
Support(' Rash Guard', ' Sweatshirts')->{' Dry Fit V-Nick'} =  6


Confidence (' Dry Fit V-Nick', ' Rash Guard')->{' Swimming Shirt'} =  55.556 %
Support(' Dry Fit V-Nick', ' Rash Guard')->{' Swimming Shirt'} =  9


Confidence (' Dry Fit V-Nick', ' Swimming Shirt')->{' Rash Guard'} =  100.0 %
Support(' Dry Fit V-Nick', ' Swimming Shirt')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Swimming Shirt')->{' Dry Fit V-Nick'} =  83.333 %
Support(' Rash Guard', ' Swimming Shirt')->{' Dry Fit V-Nick'} =  6


Confidence (' Dry Fit V-Nick', ' Rash Guard')->{' Tech Pants'} =  88.889 %
Support(' Dry Fit V-Nick', ' Rash Guard')->{' Tech Pants'} =  9


Confidence (' Dry Fit V-Nick', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Dry Fit V-Nick', ' Tech Pants')->{' Rash Guard'} =  8


Confidence (' Rash Guard', ' Tech Pants')->{' Dry Fit V-Nick'} =  88.889 %
Support(' Rash Guard', ' Tech Pants')->{' Dry Fit V-Nick'} =  9


Confidence (' Dry Fit V-Nick', ' Soccer Shoe')->{' Tech Pants'} =  100.0 %
Support(' Dry Fit V-Nick', ' Soccer Shoe')->{' Tech Pants'} =  5


Confidence (' Dry Fit V-Nick', ' Tech Pants')->{' Soccer Shoe'} =  62.5 %
Support(' Dry Fit V-Nick', ' Tech Pants')->{' Soccer Shoe'} =  8


Confidence (' Soccer Shoe', ' Tech Pants')->{' Dry Fit V-Nick'} =  100.0 %
Support(' Soccer Shoe', ' Tech Pants')->{' Dry Fit V-Nick'} =  5


Confidence (' Dry Fit V-Nick', ' Sweatshirts')->{' Tech Pants'} =  100.0 %
Support(' Dry Fit V-Nick', ' Sweatshirts')->{' Tech Pants'} =  5


Confidence (' Dry Fit V-Nick', ' Tech Pants')->{' Sweatshirts'} =  62.5 %
Support(' Dry Fit V-Nick', ' Tech Pants')->{' Sweatshirts'} =  8


Confidence (' Sweatshirts', ' Tech Pants')->{' Dry Fit V-Nick'} =  83.333 %
Support(' Sweatshirts', ' Tech Pants')->{' Dry Fit V-Nick'} =  6


Confidence (' Dry Fit V-Nick', ' Swimming Shirt')->{' Tech Pants'} =  100.0 %
Support(' Dry Fit V-Nick', ' Swimming Shirt')->{' Tech Pants'} =  5


Confidence (' Dry Fit V-Nick', ' Tech Pants')->{' Swimming Shirt'} =  62.5 %
Support(' Dry Fit V-Nick', ' Tech Pants')->{' Swimming Shirt'} =  8
```

```
Confidence (' Swimming Shirt', ' Tech Pants')->{' Dry Fit V-Nick'} =  100.0 %
Support(' Swimming Shirt', ' Tech Pants')->{' Dry Fit V-Nick'} =  5


Confidence (' Hoodies', ' Modern Pants')->{' Rash Guard'} =  100.0 %
Support(' Hoodies', ' Modern Pants')->{' Rash Guard'} =  5


Confidence (' Hoodies', ' Rash Guard')->{' Modern Pants'} =  62.5 %
Support(' Hoodies', ' Rash Guard')->{' Modern Pants'} =  8


Confidence (' Modern Pants', ' Rash Guard')->{' Hoodies'} =  83.333 %
Support(' Modern Pants', ' Rash Guard')->{' Hoodies'} =  6


Confidence (' Hoodies', ' Modern Pants')->{' Sweatshirts'} =  100.0 %
Support(' Hoodies', ' Modern Pants')->{' Sweatshirts'} =  5


Confidence (' Hoodies', ' Sweatshirts')->{' Modern Pants'} =  100.0 %
Support(' Hoodies', ' Sweatshirts')->{' Modern Pants'} =  5


Confidence (' Hoodies', ' Modern Pants')->{' Tech Pants'} =  100.0 %
Support(' Hoodies', ' Modern Pants')->{' Tech Pants'} =  5


Confidence (' Hoodies', ' Tech Pants')->{' Modern Pants'} =  62.5 %
Support(' Hoodies', ' Tech Pants')->{' Modern Pants'} =  8


Confidence (' Modern Pants', ' Tech Pants')->{' Hoodies'} =  83.333 %
Support(' Modern Pants', ' Tech Pants')->{' Hoodies'} =  6


Confidence (' Hoodies', ' Rash Guard')->{' Soccer Shoe'} =  62.5 %
Support(' Hoodies', ' Rash Guard')->{' Soccer Shoe'} =  8


Confidence (' Hoodies', ' Soccer Shoe')->{' Rash Guard'} =  100.0 %
Support(' Hoodies', ' Soccer Shoe')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Soccer Shoe')->{' Hoodies'} =  100.0 %
Support(' Rash Guard', ' Soccer Shoe')->{' Hoodies'} =  5


Confidence (' Hoodies', ' Rash Guard')->{' Sweatshirts'} =  62.5 %
Support(' Hoodies', ' Rash Guard')->{' Sweatshirts'} =  8


Confidence (' Hoodies', ' Sweatshirts')->{' Rash Guard'} =  100.0 %
Support(' Hoodies', ' Sweatshirts')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Sweatshirts')->{' Hoodies'} =  83.333 %
Support(' Rash Guard', ' Sweatshirts')->{' Hoodies'} =  6


Confidence (' Hoodies', ' Rash Guard')->{' Tech Pants'} =  100.0 %
Support(' Hoodies', ' Rash Guard')->{' Tech Pants'} =  8


Confidence (' Hoodies', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Hoodies', ' Tech Pants')->{' Rash Guard'} =  8


Confidence (' Rash Guard', ' Tech Pants')->{' Hoodies'} =  88.889 %
Support(' Rash Guard', ' Tech Pants')->{' Hoodies'} =  9
```

```
Confidence (' Hoodies', ' Soccer Shoe')->{' Tech Pants'} =  100.0 %
Support(' Hoodies', ' Soccer Shoe')->{' Tech Pants'} =  5


Confidence (' Hoodies', ' Tech Pants')->{' Soccer Shoe'} =  62.5 %
Support(' Hoodies', ' Tech Pants')->{' Soccer Shoe'} =  8


Confidence (' Soccer Shoe', ' Tech Pants')->{' Hoodies'} =  100.0 %
Support(' Soccer Shoe', ' Tech Pants')->{' Hoodies'} =  5


Confidence (' Hoodies', ' Sweatshirts')->{' Tech Pants'} =  100.0 %
Support(' Hoodies', ' Sweatshirts')->{' Tech Pants'} =  5


Confidence (' Hoodies', ' Tech Pants')->{' Sweatshirts'} =  62.5 %
Support(' Hoodies', ' Tech Pants')->{' Sweatshirts'} =  8


Confidence (' Sweatshirts', ' Tech Pants')->{' Hoodies'} =  83.333 %
Support(' Sweatshirts', ' Tech Pants')->{' Hoodies'} =  6


Confidence (' Modern Pants', ' Rash Guard')->{' Socks'} =  83.333 %
Support(' Modern Pants', ' Rash Guard')->{' Socks'} =  6


Confidence (' Modern Pants', ' Socks')->{' Rash Guard'} =  62.5 %
Support(' Modern Pants', ' Socks')->{' Rash Guard'} =  8


Confidence (' Rash Guard', ' Socks')->{' Modern Pants'} =  100.0 %
Support(' Rash Guard', ' Socks')->{' Modern Pants'} =  5


Confidence (' Modern Pants', ' Rash Guard')->{' Sweatshirts'} =  100.0 %
Support(' Modern Pants', ' Rash Guard')->{' Sweatshirts'} =  6


Confidence (' Modern Pants', ' Sweatshirts')->{' Rash Guard'} =  60.0 %
Support(' Modern Pants', ' Sweatshirts')->{' Rash Guard'} =  10


Confidence (' Rash Guard', ' Sweatshirts')->{' Modern Pants'} =  100.0 %
Support(' Rash Guard', ' Sweatshirts')->{' Modern Pants'} =  6


Confidence (' Modern Pants', ' Rash Guard')->{' Tech Pants'} =  100.0 %
Support(' Modern Pants', ' Rash Guard')->{' Tech Pants'} =  6


Confidence (' Modern Pants', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Modern Pants', ' Tech Pants')->{' Rash Guard'} =  6


Confidence (' Rash Guard', ' Tech Pants')->{' Modern Pants'} =  66.667 %
Support(' Rash Guard', ' Tech Pants')->{' Modern Pants'} =  9


Confidence (' Modern Pants', ' Socks')->{' Sweatshirts'} =  100.0 %
Support(' Modern Pants', ' Socks')->{' Sweatshirts'} =  8


Confidence (' Modern Pants', ' Sweatshirts')->{' Socks'} =  80.0 %
Support(' Modern Pants', ' Sweatshirts')->{' Socks'} =  10


Confidence (' Socks', ' Sweatshirts')->{' Modern Pants'} =  72.727 %
Support(' Socks', ' Sweatshirts')->{' Modern Pants'} =  11
```

```
Confidence (' Modern Pants', ' Socks')->{' Tech Pants'} =  62.5 %
Support(' Modern Pants', ' Socks')->{' Tech Pants'} =  8


Confidence (' Modern Pants', ' Tech Pants')->{' Socks'} =  83.333 %
Support(' Modern Pants', ' Tech Pants')->{' Socks'} =  6


Confidence (' Socks', ' Tech Pants')->{' Modern Pants'} =  100.0 %
Support(' Socks', ' Tech Pants')->{' Modern Pants'} =  5


Confidence (' Modern Pants', ' Socks')->{'Running Shoe'} =  87.5 %
Support(' Modern Pants', ' Socks')->{'Running Shoe'} =  8


Confidence (' Modern Pants', 'Running Shoe')->{' Socks'} =  87.5 %
Support(' Modern Pants', 'Running Shoe')->{' Socks'} =  8


Confidence (' Socks', 'Running Shoe')->{' Modern Pants'} =  70.0 %
Support(' Socks', 'Running Shoe')->{' Modern Pants'} =  10


Confidence (' Modern Pants', ' Sweatshirts')->{' Tech Pants'} =  60.0 %
Support(' Modern Pants', ' Sweatshirts')->{' Tech Pants'} =  10


Confidence (' Modern Pants', ' Tech Pants')->{' Sweatshirts'} =  100.0 %
Support(' Modern Pants', ' Tech Pants')->{' Sweatshirts'} =  6


Confidence (' Sweatshirts', ' Tech Pants')->{' Modern Pants'} =  100.0 %
Support(' Sweatshirts', ' Tech Pants')->{' Modern Pants'} =  6


Confidence (' Modern Pants', ' Sweatshirts')->{'Running Shoe'} =  80.0 %
Support(' Modern Pants', ' Sweatshirts')->{'Running Shoe'} =  10


Confidence (' Modern Pants', 'Running Shoe')->{' Sweatshirts'} =  100.0 %
Support(' Modern Pants', 'Running Shoe')->{' Sweatshirts'} =  8


Confidence (' Sweatshirts', 'Running Shoe')->{' Modern Pants'} =  80.0 %
Support(' Sweatshirts', 'Running Shoe')->{' Modern Pants'} =  10


Confidence (' Rash Guard', ' Soccer Shoe')->{' Tech Pants'} =  100.0 %
Support(' Rash Guard', ' Soccer Shoe')->{' Tech Pants'} =  5


Confidence (' Rash Guard', ' Tech Pants')->{' Soccer Shoe'} =  55.556 %
Support(' Rash Guard', ' Tech Pants')->{' Soccer Shoe'} =  9


Confidence (' Soccer Shoe', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Soccer Shoe', ' Tech Pants')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Socks')->{' Sweatshirts'} =  100.0 %
Support(' Rash Guard', ' Socks')->{' Sweatshirts'} =  5


Confidence (' Rash Guard', ' Sweatshirts')->{' Socks'} =  83.333 %
Support(' Rash Guard', ' Sweatshirts')->{' Socks'} =  6


Confidence (' Rash Guard', ' Socks')->{' Tech Pants'} =  100.0 %
Support(' Rash Guard', ' Socks')->{' Tech Pants'} =  5
```

```
Confidence (' Rash Guard', ' Tech Pants')->{' Socks'} =  55.556 %
Support(' Rash Guard', ' Tech Pants')->{' Socks'} =  9


Confidence (' Socks', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Socks', ' Tech Pants')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Sweatshirts')->{' Tech Pants'} =  100.0 %
Support(' Rash Guard', ' Sweatshirts')->{' Tech Pants'} =  6


Confidence (' Rash Guard', ' Tech Pants')->{' Sweatshirts'} =  66.667 %
Support(' Rash Guard', ' Tech Pants')->{' Sweatshirts'} =  9


Confidence (' Sweatshirts', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Sweatshirts', ' Tech Pants')->{' Rash Guard'} =  6


Confidence (' Rash Guard', ' Swimming Shirt')->{' Tech Pants'} =  83.333 %
Support(' Rash Guard', ' Swimming Shirt')->{' Tech Pants'} =  6


Confidence (' Rash Guard', ' Tech Pants')->{' Swimming Shirt'} =  55.556 %
Support(' Rash Guard', ' Tech Pants')->{' Swimming Shirt'} =  9


Confidence (' Swimming Shirt', ' Tech Pants')->{' Rash Guard'} =  100.0 %
Support(' Swimming Shirt', ' Tech Pants')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Swimming Shirt')->{'Running Shoe'} =  83.333 %
Support(' Rash Guard', ' Swimming Shirt')->{'Running Shoe'} =  6


Confidence (' Rash Guard', 'Running Shoe')->{' Swimming Shirt'} =  83.333 %
Support(' Rash Guard', 'Running Shoe')->{' Swimming Shirt'} =  6


Confidence (' Swimming Shirt', 'Running Shoe')->{' Rash Guard'} =  100.0 %
Support(' Swimming Shirt', 'Running Shoe')->{' Rash Guard'} =  5


Confidence (' Rash Guard', ' Tech Pants')->{'Running Shoe'} =  55.556 %
Support(' Rash Guard', ' Tech Pants')->{'Running Shoe'} =  9


Confidence (' Rash Guard', 'Running Shoe')->{' Tech Pants'} =  83.333 %
Support(' Rash Guard', 'Running Shoe')->{' Tech Pants'} =  6


Confidence (' Tech Pants', 'Running Shoe')->{' Rash Guard'} =  100.0 %
Support(' Tech Pants', 'Running Shoe')->{' Rash Guard'} =  5


Confidence (' Socks', ' Tech Pants')->{' Sweatshirts'} =  100.0 %
Support(' Socks', ' Tech Pants')->{' Sweatshirts'} =  5


Confidence (' Sweatshirts', ' Tech Pants')->{' Socks'} =  83.333 %
Support(' Sweatshirts', ' Tech Pants')->{' Socks'} =  6


Confidence (' Socks', ' Sweatshirts')->{'Running Shoe'} =  81.818 %
Support(' Socks', ' Sweatshirts')->{'Running Shoe'} =  11


Confidence (' Socks', 'Running Shoe')->{' Sweatshirts'} =  90.0 %
Support(' Socks', 'Running Shoe')->{' Sweatshirts'} =  10
```

```
Confidence (' Sweatshirts', 'Running Shoe')->{' Socks'} =  90.0 %
Support(' Sweatshirts', 'Running Shoe')->{' Socks'} =  10
```

Dataset 3

Note- (Select any value for 'i' other than 1 & 2)

In [42]:

```
i = int(input('Enter any desired value '))
if i == 1:
    data = pd.read_csv('GroceryStoreDataSet.csv')
if i ==2:
    data = pd.read_csv('nike.csv')
else:
    data = pd.read_csv('flipkart.csv')
```

```
Enter any desired value 4
```

In [43]:

```
data.head()
```

Out[43]:

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9
0	Desk Top	Flash Drive	Microsoft Office	Speakers	Anti-Virus	NaN	NaN	NaN	NaN
1	Lab Top	Flash Drive	Microsoft Office	Anti-Virus	Lab Top Case	External Hard-Drive	NaN	NaN	NaN
2	Lab Top	Flash Drive	Anti-Virus	External Hard-Drive	Lab Top Case	NaN	NaN	NaN	NaN
3	Lab Top	Lab Top Case	Anti-Virus	NaN	NaN	NaN	NaN	NaN	NaN
4	Lab Top	Flash Drive	Microsoft Office	NaN	NaN	NaN	NaN	NaN	NaN

In [44]:

```
data.tail()
```

Out[44]:

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9
14	Digital Camera	Microsoft Office	Anti-Virus	Lab Top Case	External Hard-Drive	Speakers	NaN	NaN	NaN
15	Digital Camera	Lab Top Case	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	Digital Camera	Speakers	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	Digital Camera	Printer	Flash Drive	Microsoft Office	Speakers	Lab Top Case	Anti-Virus	NaN	NaN
18	Digital Camera	Speakers	Anti-Virus	Lab Top Case	NaN	NaN	NaN	NaN	NaN

In [45]:

```
data.shape
```

Out[45]:

```
(19, 9)
```

List of all unique items in each transaction

List of all unique items in each transaction

In [46]:

```
for i in range(data.shape[0]):
    print(f'List of Unique Items in Transaction : {i} ')
    print(data.iloc[i].dropna().unique(),' \n')
```

```
List of Unique Items in Transaction : 0
[' Desk Top' ' Flash Drive' ' Microsoft Office' ' Speakers' ' Anti-Virus'] 

List of Unique Items in Transaction : 1
[' Lab Top' ' Flash Drive' ' Microsoft Office' ' Anti-Virus'
 ' Lab Top Case' ' External Hard-Drive'] 

List of Unique Items in Transaction : 2
[' Lab Top' ' Flash Drive' ' Anti-Virus' ' External Hard-Drive'
 ' Lab Top Case'] 

List of Unique Items in Transaction : 3
[' Lab Top' ' Lab Top Case' ' Anti-Virus'] 

List of Unique Items in Transaction : 4
[' Lab Top' ' Flash Drive' ' Microsoft Office'] 

List of Unique Items in Transaction : 5
[' Desk Top' ' Flash Drive' ' Microsoft Office'] 

List of Unique Items in Transaction : 6
[' Lab Top' ' Anti-Virus'] 

List of Unique Items in Transaction : 7
[' Desk Top' ' Flash Drive' ' Microsoft Office' ' Lab Top Case'
 ' Anti-Virus' ' Speakers' ' External Hard-Drive'] 

List of Unique Items in Transaction : 8
[' Digital Camera ' ' Desk Top' ' Printer' ' Flash Drive'
 ' Microsoft Office' ' Lab Top Case' ' Anti-Virus' ' External Hard-Drive'
 ' Speakers'] 

List of Unique Items in Transaction : 9
[' Lab Top' ' Lab Top Case' ' External Hard-Drive' ' Speakers'
 ' Anti-Virus'] 

List of Unique Items in Transaction : 10
[' Digital Camera ' ' Lab Top Case' ' External Hard-Drive' ' Anti-Virus'
 ' Speakers'] 

List of Unique Items in Transaction : 11
[' Digital Camera '] 

List of Unique Items in Transaction : 12
[' Digital Camera ' ' Printer' ' Flash Drive' ' Microsoft Office'] 

List of Unique Items in Transaction : 13
[' Printer' ' Microsoft Office' ' Anti-Virus' ' Lab Top Case' ' Speakers'
 ' External Hard-Drive'] 

List of Unique Items in Transaction : 14
[' Digital Camera' ' Microsoft Office' ' Anti-Virus' ' Lab Top Case'
 ' External Hard-Drive' ' Speakers'] 

List of Unique Items in Transaction : 15
[' Digital Camera ' ' Lab Top Case'] 

List of Unique Items in Transaction : 16
[' Digital Camera ' ' Speakers'] 

List of Unique Items in Transaction : 17
[' Digital Camera ' ' Printer' ' Flash Drive' ' Microsoft Office'
 ' Speakers' ' Lab Top Case' ' Anti-Virus'] 

List of Unique Items in Transaction : 18
```

List of unique items in transaction : 16
[' Digital Camera ' ' Speakers' ' Anti-Virus' ' Lab Top Case']

In [47]:

```
print(f'Total no. of transactions in this dataset {data.shape[0]}')
```

Total no. of transactions in this dataset 19

In [48]:

```
minimum_s_count = float(input('enter any value for min support count '))
```

enter any value for min support count 7

In [49]:

```
minimum_confi = float(input('Enter any desired value for confidence '))
```

Enter any desired value for confidence 45.9

In [50]:

```
records = []
for i in range(0, data.shape[0]):
    records.append([str(data.values[i,j]) for j in range(0, data.shape[1])])

items = sorted([item for sublist in records for item in sublist if item != 'nan'])
```

List of all unique items in this dataset

In [51]:

```
set(items)
```

Out[51]:

```
{' Anti-Virus',
 ' Desk Top',
 ' Digital Camera',
 ' Digital Camera ',
 ' External Hard-Drive',
 ' Flash Drive',
 ' Lab Top',
 ' Lab Top Case',
 ' Microsoft Office',
 ' Printer',
 ' Speakers'}
```

In [52]:

```
# To get a single item with their overall count in all transactions given.

def stage_1(items, minimum_s_count):
    c1 = {i:items.count(i) for i in items}
    l1 = {}
    for key, value in c1.items():
        if value >= minimum_s_count:
            l1[key] = value

    return c1, l1
```

In [53]:

```
# To get sets of 2 items with their overall count in all transactions given.
def stage_2(l1, records, minimum_s_count):
    l1 = sorted(list(l1.keys()))
    L1 = list(itertools.combinations(l1, 2))              # To make all possible combina
tions of 2 items
```

```
    c2 = {}
    l2 = {}
    for iter1 in L1:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c2[iter1] = count
    for key, value in c2.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l1, 1):
                l2[key] = value

    return c2, l2
```

In [54]:

```
# To get sets of 3 items with their overall count in all transactions given.
def stage_3(l2, records, minimum_s_count):
    l2 = list(l2.keys())
    L2 = sorted(list(set([item for t in l2 for item in t])))
    L2 = list(itertools.combinations(L2, 3))           # To make all possible combinatio
ns of 3 items
    c3 = {}
    l3 = {}
    for iter1 in L2:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c3[iter1] = count
    for key, value in c3.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l2, 2):
                l3[key] = value

    return c3, l3
```

In [55]:

```
# To get sets of 4 items with their overall count in all transactions given.
def stage_4(l3, records, minimum_s_count):
    l3 = list(l3.keys())
    L3 = sorted(list(set([item for t in l3 for item in t])))
    L3 = list(itertools.combinations(L3, 4))             # To make all possible combinat
ions of 4 items
    c4 = {}
    l4 = {}
    for iter1 in L3:
        count = 0
        for iter2 in records:
            if sublist(iter1, iter2):
                count+=1
        c4[iter1] = count
    for key, value in c4.items():
        if value >= minimum_s_count:
            if check_subset_frequency(key, l3, 3):
                l4[key] = value

    return c4, l4
```

In [56]:

```
def sublist(lst1, lst2):
    return set(lst1) <= set(lst2)

def check_subset_frequency(itemset, l, n):
    if n>1:
        subsets = list(itertools.combinations(itemset, n))
    else:
        subsets = itemset
```

```
    for iter1 in subsets:
        if not iter1 in l:
            return False
    return True
```

Below part prints all L1,L2,L3

All of them have pairs of 1 item, 2 items, 3 items respectively following minimum support decided by user above

In [57]:

```
c1, l1 = stage_1(items, minimum_s_count)
c2, l2 = stage_2(l1, records, minimum_s_count)
c3, l3 = stage_3(l2, records, minimum_s_count)
c4, l4 = stage_4(l3, records, minimum_s_count)
```

In [58]:

```
for key, value in l1.items():
    print(key, ' : ', value)
```

```
 Anti-Virus  :  13
 Digital Camera  :  8
 External Hard-Drive  :  8
 Flash Drive  :  9
 Lab Top Case  :  12
 Microsoft Office  :  10
 Speakers  :  10
```

In [59]:

```
for key, value in l2.items():
    print(key, ' : ', value)
```

```
(' Anti-Virus', ' External Hard-Drive')  :  8
(' Anti-Virus', ' Lab Top Case')  :  11
(' Anti-Virus', ' Microsoft Office')  :  7
(' Anti-Virus', ' Speakers')  :  9
(' External Hard-Drive', ' Lab Top Case')  :  8
(' Flash Drive', ' Microsoft Office')  :  8
(' Lab Top Case', ' Speakers')  :  8
```

In [60]:

```
for key, value in l3.items():
    print(key, ' : ', value)
```

```
(' Anti-Virus', ' External Hard-Drive', ' Lab Top Case')  :  8
(' Anti-Virus', ' Lab Top Case', ' Speakers')  :  8
```

Print Association Rules

(with their Minimum support & minimum confidence)

In [61]:

```
itemlist = {**l1, **l2, **l3, **l4}

def support_count(itemset, itemlist):
    return itemlist[itemset]

def print_sets():
    print('Possible sets with just 1 item ',l1)

sets = []
for iter1 in list(l3.keys()):
```

```
    subsets = list(itertools.combinations(iter1, 2))
    sets.append(subsets)

list_l3 = list(l3.keys())
print('ASSOCIATION RULES : '+ '\n')
for i in range(0, len(list_l3)):
    for iter1 in sets[i]:
        a = iter1
        b = set(list_l3[i]) - set(iter1)
        confidence = (support_count(list_l3[i], itemlist)/support_count(iter1, itemlist)
)*100
        support_i = support_count(iter1,itemlist)
        if confidence > minimum_confi:
            print("Confidence {}->{} = ".format(a,b), round(confidence,3),'%')
            print('Support{}->{} = '.format(a,b),round(support_i,3))
            print('\n')
#         print("Confidence{}->{} = ".format(a,b), confidence)
```

```
ASSOCIATION RULES :

Confidence (' Anti-Virus', ' External Hard-Drive')->{' Lab Top Case'} =  100.0 %
Support(' Anti-Virus', ' External Hard-Drive')->{' Lab Top Case'} =  8


Confidence (' Anti-Virus', ' Lab Top Case')->{' External Hard-Drive'} =  72.727 %
Support(' Anti-Virus', ' Lab Top Case')->{' External Hard-Drive'} =  11


Confidence (' External Hard-Drive', ' Lab Top Case')->{' Anti-Virus'} =  100.0 %
Support(' External Hard-Drive', ' Lab Top Case')->{' Anti-Virus'} =  8


Confidence (' Anti-Virus', ' Lab Top Case')->{' Speakers'} =  72.727 %
Support(' Anti-Virus', ' Lab Top Case')->{' Speakers'} =  11


Confidence (' Anti-Virus', ' Speakers')->{' Lab Top Case'} =  88.889 %
Support(' Anti-Virus', ' Speakers')->{' Lab Top Case'} =  9


Confidence (' Lab Top Case', ' Speakers')->{' Anti-Virus'} =  100.0 %
Support(' Lab Top Case', ' Speakers')->{' Anti-Virus'} =  8
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: