Name: Prekshita Vasudeo Patil

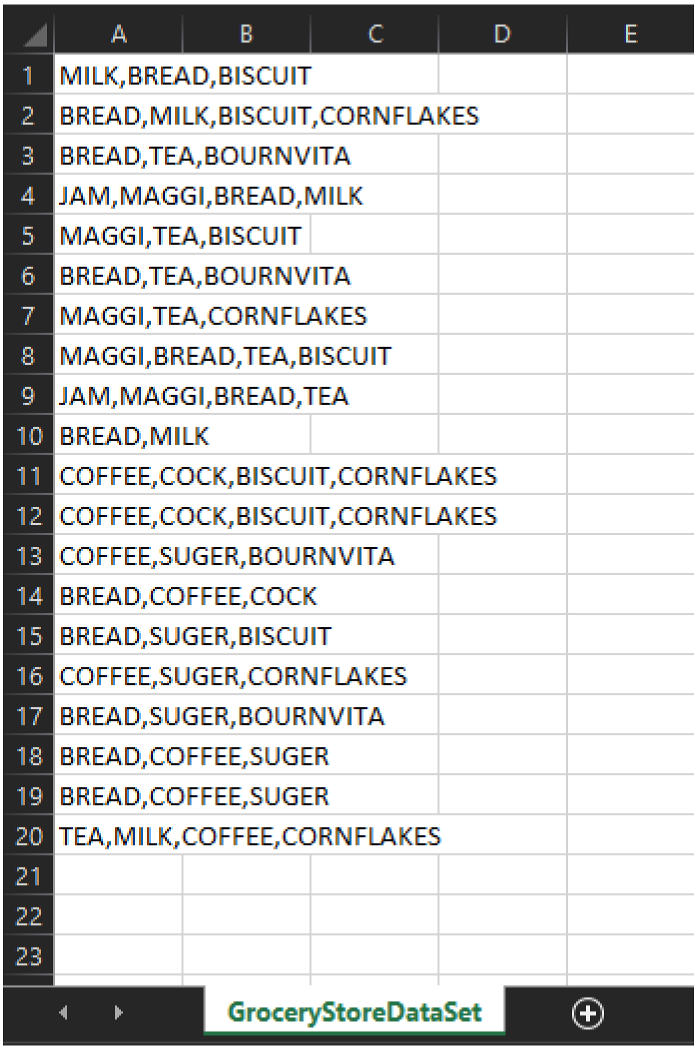
Registration No.: 20MAI0073

Subject: DWM

Description About dataset

The given dataset contains 20 rows and there are number of items present in each row.

- Each row is considered as transaction and the number of product present in it are the products purchased during that transaction.



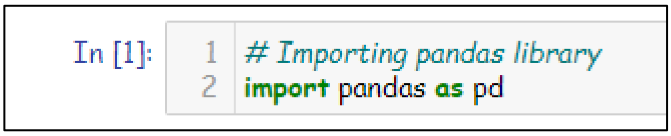
	A	B	C	D	E
1	MILK,BREAD,BISCUIT				
2	BREAD,MILK,BISCUIT,CORNFLAKES				
3	BREAD,TEA,BOURNVITA				
4	JAM,MAGGI,BREAD,MILK				
5	MAGGI,TEA,BISCUIT				
6	BREAD,TEA,BOURNVITA				
7	MAGGI,TEA,CORNFLAKES				
8	MAGGI,BREAD,TEA,BISCUIT				
9	JAM,MAGGI,BREAD,TEA				
10	BREAD,MILK				
11	COFFEE,COCK,BISCUIT,CORNFLAKES				
12	COFFEE,COCK,BISCUIT,CORNFLAKES				
13	COFFEE,SUGER,BOURNVITA				
14	BREAD,COFFEE,COCK				
15	BREAD,SUGER,BISCUIT				
16	COFFEE,SUGER,CORNFLAKES				
17	BREAD,SUGER,BOURNVITA				
18	BREAD,COFFEE,SUGER				
19	BREAD,COFFEE,SUGER				
20	TEA,MILK,COFFEE,CORNFLAKES				
21					
22					
23					

- There are 11 different items present in the dataset that are-
    - BISCUIT
    - BOURNVITA
    - BREAD
    - COCK
    - COFFEE
    - CORNFLAKES
    - JAM
    - MAGGI
    - MILK
    - SUGER
    - TEA
- The maximum number of transactions that can be seen at a glance are 4 whereas the minimum number of elements that can be seen are 2

Python Program Implementation

- Importing pandas library

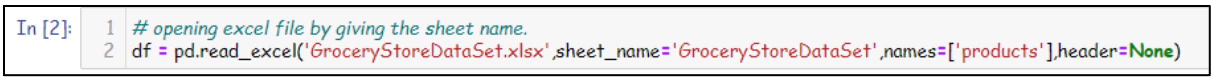
  import pandas as pd



```
In [1]:
# Importing pandas library
import pandas as pd
```

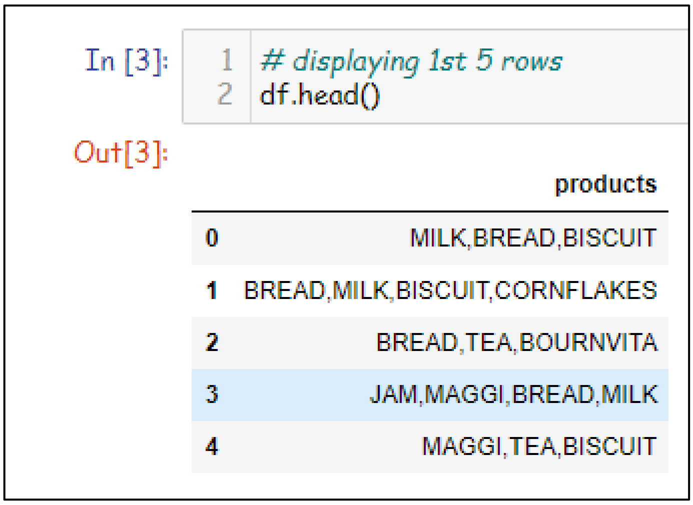
- opening excel file by giving the sheet name.

  df = pd.read_excel('GroceryStoreDataSet.xlsx',sheet_name='GroceryStoreDataSet',names=['products'],header=None)



```
In [2]:
# opening excel file by giving the sheet name.
df = pd.read_excel('GroceryStoreDataSet.xlsx',sheet_name='GroceryStoreDataSet',names=['products'],header=None)
```

- displaying 1st 5 rows
  df.head()



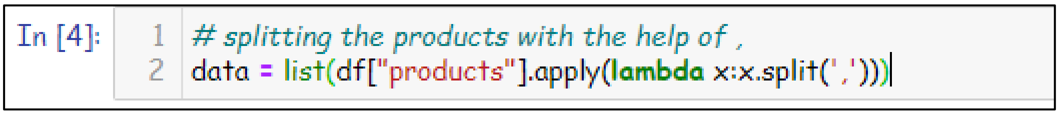
In [3]:

```
# displaying 1st 5 rows
df.head()
```

Out[3]:

	products
0	MILK,BREAD,BISCUIT
1	BREAD,MILK,BISCUIT,CORNFLAKES
2	BREAD,TEA,BOURNVITA
3	JAM,MAGGI,BREAD,MILK
4	MAGGI,TEA,BISCUIT

- splitting the products with the help of comma(,)
  data = list(df["products"].apply(lambda x:x.split(',')))

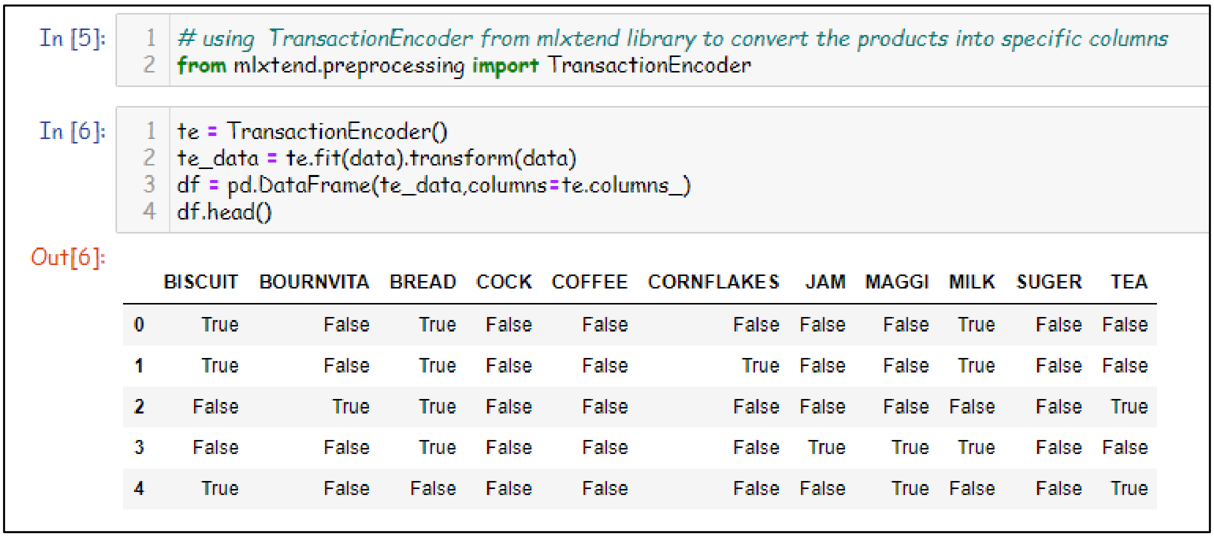


In [4]:

```
# splitting the products with the help of ,
data = list(df["products"].apply(lambda x:x.split(',')))
```

- using TransactionEncoder from mlxtend library to convert the products into specific columns

  from mlxtend.preprocessing import TransactionEncoder
  te = TransactionEncoder()
  te_data = te.fit(data).transform(data)
  df = pd.DataFrame(te_data,columns=te.columns_)
  df.head()



In [5]:
```
# using  TransactionEncoder from mlxtend library to convert the products into specific columns
from mlxtend.preprocessing import TransactionEncoder
```

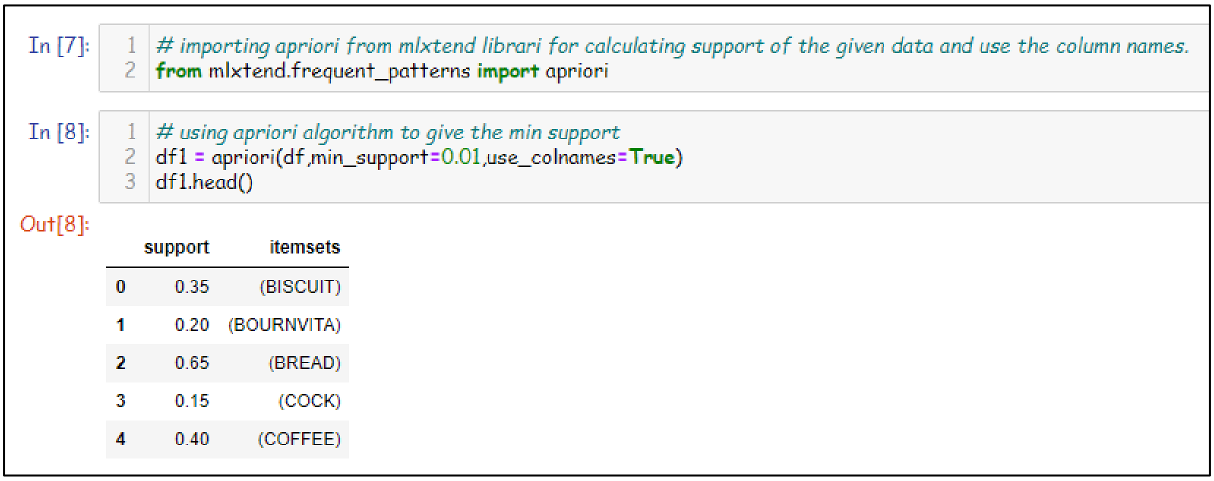
In [6]:
```
te = TransactionEncoder()
te_data = te.fit(data).transform(data)
df = pd.DataFrame(te_data,columns=te.columns_)
df.head()
```

Out[6]:

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	SUGER	TEA
0	True	False	True	False	False	False	False	False	True	False	False
1	True	False	True	False	False	True	False	False	True	False	False
2	False	True	True	False	False	False	False	False	False	False	True
3	False	False	True	False	False	False	True	True	True	False	False
4	True	False	False	False	False	False	False	True	False	False	True

- # importing apriori from mlxtend librari for calculating support of the given data and use the column names.

  from mlxtend.frequent_patterns import apriori
  # using apriori algorithm to give the min support
  df1 = apriori(df,min_support=0.01,use_colnames=True)
  df1.head()



In [7]:
```
# importing apriori from mlxtend librari for calculating support of the given data and use the column names.
from mlxtend.frequent_patterns import apriori
```

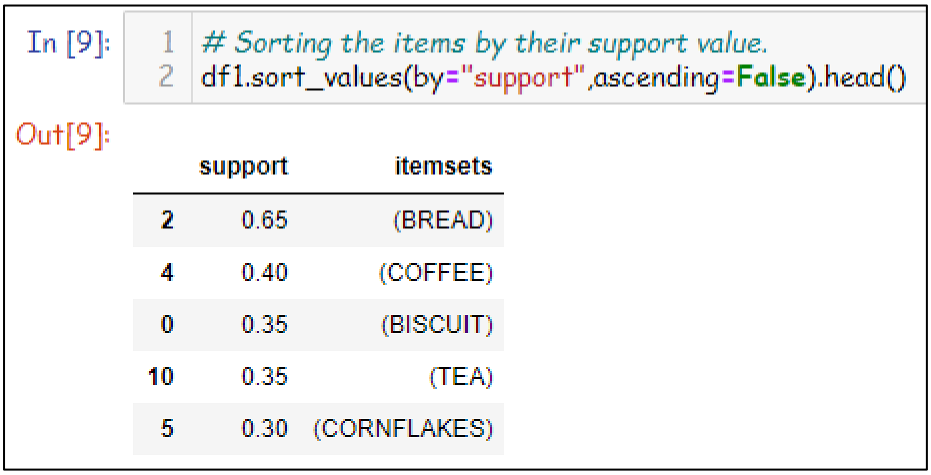
In [8]:
```
# using apriori algorithm to give the min support
df1 = apriori(df,min_support=0.01,use_colnames=True)
df1.head()
```

Out[8]:

	support	itemsets
0	0.35	(BISCUIT)
1	0.20	(BOURNVITA)
2	0.65	(BREAD)
3	0.15	(COCK)
4	0.40	(COFFEE)

- Sorting the items by their support value.

  df1.sort_values(by="support",ascending=False).head()

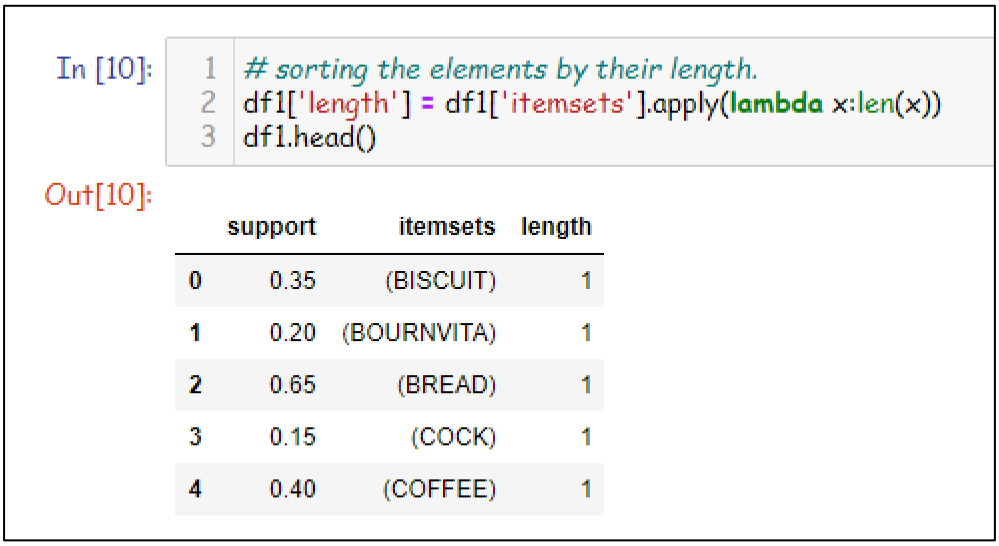


```
In [9]:  # Sorting the items by their support value.
          df1.sort_values(by="support",ascending=False).head()
```

```
Out[9]:
        support      itemsets
    2      0.65       (BREAD)
    4      0.40      (COFFEE)
    0      0.35     (BISCUIT)
   10      0.35         (TEA)
    5      0.30  (CORNFLAKES)
```

- sorting the elements by their length.

  df1['length'] = df1['itemsets'].apply(lambda x:len(x))
  df1.head()

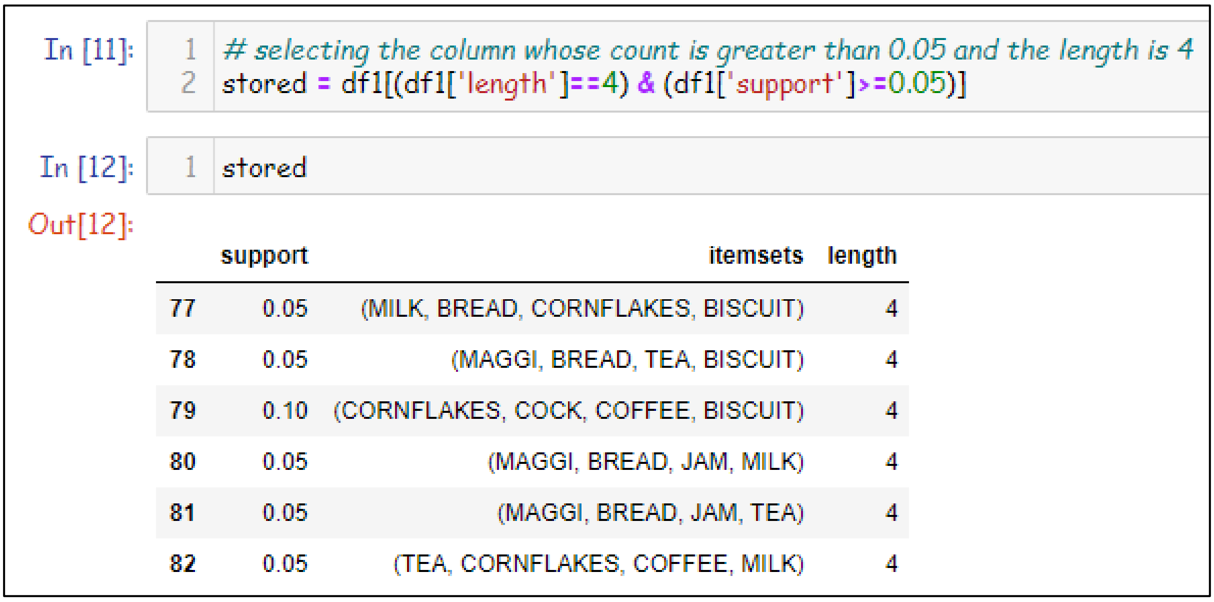


```
In [10]:  # sorting the elements by their length.
           df1['length'] = df1['itemsets'].apply(lambda x:len(x))
           df1.head()
```

```
Out[10]:
       support      itemsets  length
    0     0.35     (BISCUIT)       1
    1     0.20  (BOURNVITA)       1
    2     0.65       (BREAD)       1
    3     0.15        (COCK)       1
    4     0.40      (COFFEE)       1
```

- selecting the column whose count is greater than 0.05 and the length is 4

  stored = df1[(df1['length']==4) & (df1['support']>=0.05)]



In [11]:
```python
# selecting the column whose count is greater than 0.05 and the length is 4
stored = df1[(df1['length']==4) & (df1['support']>=0.05)]
```

In [12]:
```python
stored
```

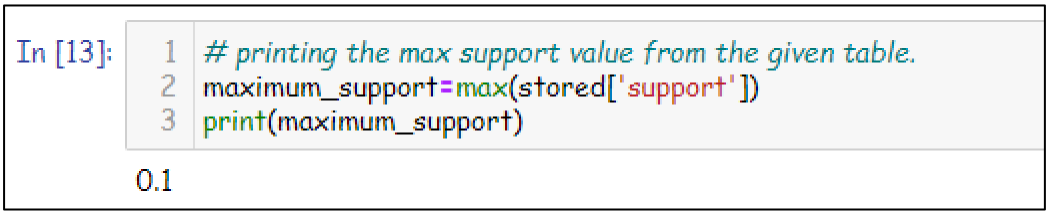
Out[12]:

	support	itemsets	length
77	0.05	(MILK, BREAD, CORNFLAKES, BISCUIT)	4
78	0.05	(MAGGI, BREAD, TEA, BISCUIT)	4
79	0.10	(CORNFLAKES, COCK, COFFEE, BISCUIT)	4
80	0.05	(MAGGI, BREAD, JAM, MILK)	4
81	0.05	(MAGGI, BREAD, JAM, TEA)	4
82	0.05	(TEA, CORNFLAKES, COFFEE, MILK)	4

- printing the max support value from the given table.

maximum_support=max(stored['support'])

print(maximum_support)

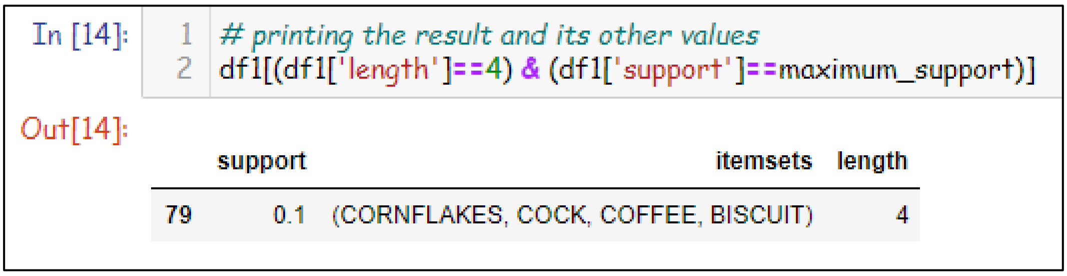


```
In [13]: # printing the max support value from the given table.
         maximum_support=max(stored['support'])
         print(maximum_support)

         0.1
```

- printing the result and its other values

df1[(df1['length']==4) & (df1['support']==maximum_support)]



```
In [14]: # printing the result and its other values
         df1[(df1['length']==4) & (df1['support']==maximum_support)]

Out[14]:
            support                             itemsets   length
        79      0.1   (CORNFLAKES, COCK, COFFEE, BISCUIT)       4
```