**3/5/2021**

**Name** Rishabh Sharma

**Registration Number** 20MAI0082

Association Rule Mining

ASSIGNMENT-2

**GitHub Link :-** https://github.com/rishabh5197/Data-Mining/tree/main/Assignment-2

**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.



* 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



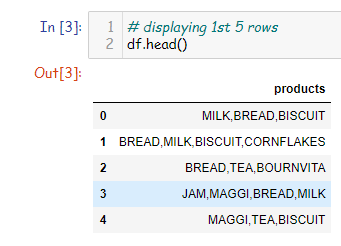
* opening excel file by giving the sheet name.

df = pd.read\_excel('GroceryStoreDataSet.xlsx',sheet\_name='GroceryStoreDataSet',names=['products'],header=None)



* displaying 1st 5 rows

df.head()



* splitting the products with the help of comma(,)

data = list(df["products"].apply(lambda x:x.split(',')))



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

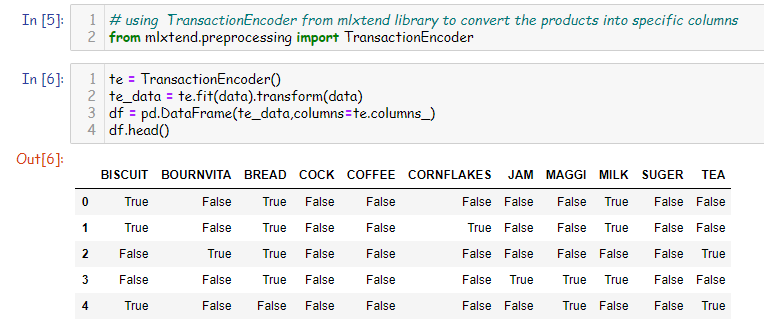
from mlxtend.preprocessing import TransactionEncoder

te = TransactionEncoder()

te\_data = te.fit(data).transform(data)

df = pd.DataFrame(te\_data,columns=te.columns\_)

df.head()



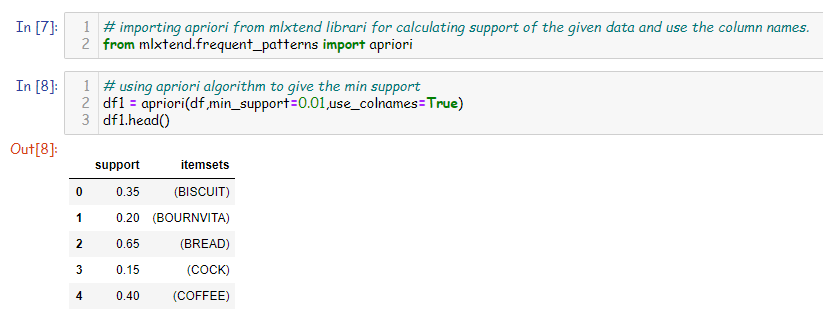
* # importing apriori from mlxtend librari for calculating support of the given data and use the column names.

from mlxtend.frequent\_patterns import apriori

# using apriori algorithm to give the min support

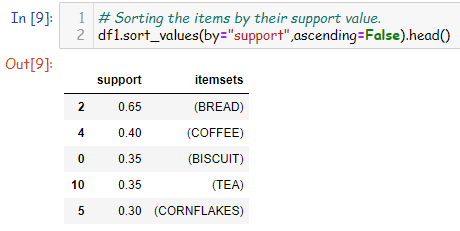
df1 = apriori(df,min\_support=0.01,use\_colnames=True)

df1.head()



* Sorting the items by their support value.

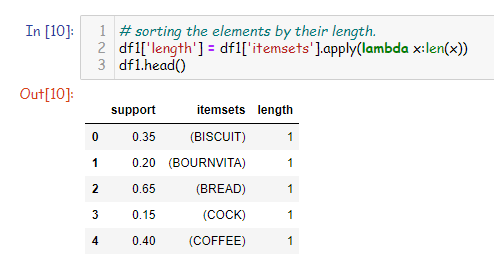
df1.sort\_values(by="support",ascending=False).head()



* sorting the elements by their length.

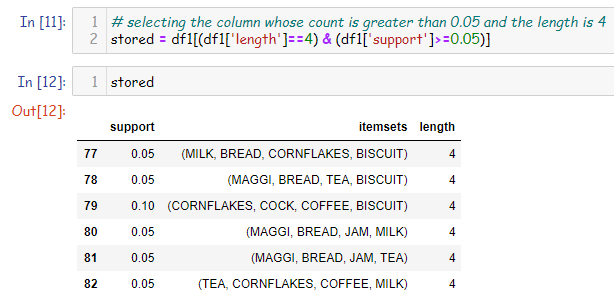
df1['length'] = df1['itemsets'].apply(lambda x:len(x))

df1.head()



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

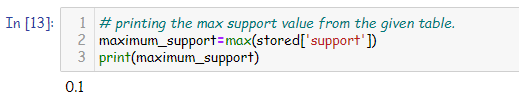
stored = df1[(df1['length']==4) & (df1['support']>=0.05)]



* printing the max support value from the given table.

maximum\_support=max(stored['support'])

print(maximum\_support)



* printing the result and its other values

df1[(df1['length']==4) & (df1['support']==maximum\_support)]

