Association Rules and Apriori Algorithm

Project Name:

Using Association Rules and Apriori Algorithm to increase the Sales of a Grocery Store ¶

In this project Association Rules and Apriori algorithm will be used to increase the Sales of a Grocery Store. This is a kind of Market Basket analysis. Association rules tell us that two or more items are related. Metrics allow us to quantify the usefulness of those relationships. In this analysis I will apply different metrics to evaluate association rules and I will interpret support, confidence, and lift metrics. I will use association rules and metrics to increase the sales of a grocery store. The data using in this project will be retrieved from "https://www.kaggle.com/shazadudwadia/supermarket (https://www.kaggle.com/shazadudwadia/supermarket)".

In [1]:

```
import pandas as pd
import numpy as np
!pip install mlxtend
from mlxtend.frequent_patterns import apriori, association_rules
```

```
Requirement already satisfied: mlxtend in c:\users\acer\anaconda3\lib\site
-packages (0.17.2)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\acer\anaco
nda3\lib\site-packages (from mlxtend) (0.22.1)
Requirement already satisfied: scipy>=1.2.1 in c:\users\acer\anaconda3\lib
\site-packages (from mlxtend) (1.4.1)
Requirement already satisfied: pandas>=0.24.2 in c:\users\acer\anaconda3\l
ib\site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\acer\anaconda
3\lib\site-packages (from mlxtend) (3.1.3)
Requirement already satisfied: setuptools in c:\users\acer\anaconda3\lib\s
ite-packages (from mlxtend) (45.2.0.post20200210)
Requirement already satisfied: joblib>=0.13.2 in c:\users\acer\anaconda3\l
ib\site-packages (from mlxtend) (0.14.1)
Requirement already satisfied: numpy>=1.16.2 in c:\users\acer\anaconda3\li
b\site-packages (from mlxtend) (1.18.1)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\acer\ana
conda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\users\acer\anaconda3\lib
\site-packages (from pandas>=0.24.2->mlxtend) (2019.3)
Requirement already satisfied: cycler>=0.10 in c:\users\acer\anaconda3\lib
\site-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
c:\users\acer\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxten
d) (2.4.6)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\acer\anaconda
3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.1.0)
Requirement already satisfied: six>=1.5 in c:\users\acer\anaconda3\lib\sit
e-packages (from python-dateutil>=2.6.1->pandas>=0.24.2->mlxtend) (1.14.0)
```

In [2]:

```
df = pd.read_csv('GroceryStoreDataSet.csv',
                  names = ['products'], header = None)
df
```

Out[2]:

	products
0	MILK,BREAD,BISCUIT
1	BREAD,MILK,BISCUIT,CORNFLAKES
2	BREAD,TEA,BOURNVITA
3	JAM,MAGGI,BREAD,MILK
4	MAGGI,TEA,BISCUIT
5	BREAD,TEA,BOURNVITA
6	MAGGI,TEA,CORNFLAKES
7	MAGGI,BREAD,TEA,BISCUIT
8	JAM,MAGGI,BREAD,TEA
9	BREAD,MILK
10	COFFEE,COCK,BISCUIT,CORNFLAKES
11	COFFEE,COCK,BISCUIT,CORNFLAKES
12	COFFEE,SUGER,BOURNVITA
13	BREAD,COFFEE,COCK
14	BREAD,SUGER,BISCUIT
15	COFFEE,SUGER,CORNFLAKES
16	BREAD,SUGER,BOURNVITA
17	BREAD,COFFEE,SUGER
18	BREAD,COFFEE,SUGER
19	TEA,MILK,COFFEE,CORNFLAKES

In [3]:

```
df.columns
```

Out[3]:

```
Index(['products'], dtype='object')
```

In [4]:

```
df.values
```

Out[4]:

```
array([['MILK,BREAD,BISCUIT'],
       ['BREAD,MILK,BISCUIT,CORNFLAKES'],
       ['BREAD,TEA,BOURNVITA'],
       ['JAM,MAGGI,BREAD,MILK'],
       ['MAGGI,TEA,BISCUIT'],
       ['BREAD,TEA,BOURNVITA'],
       ['MAGGI,TEA,CORNFLAKES'],
       ['MAGGI,BREAD,TEA,BISCUIT'],
       ['JAM,MAGGI,BREAD,TEA'],
       ['BREAD,MILK'],
       ['COFFEE,COCK,BISCUIT,CORNFLAKES'],
       ['COFFEE,COCK,BISCUIT,CORNFLAKES'],
       ['COFFEE,SUGER,BOURNVITA'],
       ['BREAD,COFFEE,COCK'],
       ['BREAD,SUGER,BISCUIT'],
       ['COFFEE,SUGER,CORNFLAKES'],
       ['BREAD,SUGER,BOURNVITA'],
       ['BREAD,COFFEE,SUGER'],
       ['BREAD,COFFEE,SUGER'],
       ['TEA,MILK,COFFEE,CORNFLAKES']], dtype=object)
```

In [5]:

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

Out[5]:

```
[['MILK', 'BREAD', 'BISCUIT'],
 ['BREAD', 'MILK', 'BISCUIT', 'CORNFLAKES'],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['JAM', 'MAGGI', 'BREAD', 'MILK'],
 ['MAGGI', 'TEA', 'BISCUIT'],
 ['BREAD', 'TEA', 'BOURNVITA'],
 ['MAGGI', 'TEA', 'CORNFLAKES'],
 ['MAGGI', 'BREAD', 'TEA', 'BISCUIT'],
 ['JAM', 'MAGGI', 'BREAD', 'TEA'],
 ['BREAD', 'MILK'],
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
 ['COFFEE', 'COCK', 'BISCUIT', 'CORNFLAKES'],
 ['COFFEE', 'SUGER', 'BOURNVITA'],
 ['BREAD', 'COFFEE', 'COCK'],
 ['BREAD', 'SUGER', 'BISCUIT'],
 ['COFFEE', 'SUGER', 'CORNFLAKES'],
 ['BREAD', 'SUGER', 'BOURNVITA'],
 ['BREAD', 'COFFEE', 'SUGER'],
 ['BREAD', 'COFFEE', 'SUGER'],
 ['TEA', 'MILK', 'COFFEE', 'CORNFLAKES']]
```

In [6]:

```
from mlxtend.preprocessing import TransactionEncoder
```

In [7]:

```
te = TransactionEncoder()
te_data = te.fit(data).transform(data)
df = pd.DataFrame(te_data, columns = te.columns_)
df
```

Out[7]:

	BISCUIT	BOURNVITA	BREAD	COCK	COFFEE	CORNFLAKES	JAM	MAGGI	MILK	S
0	True	False	True	False	False	False	False	False	True	
1	True	False	True	False	False	True	False	False	True	
2	False	True	True	False	False	False	False	False	False	
3	False	False	True	False	False	False	True	True	True	
4	True	False	False	False	False	False	False	True	False	
5	False	True	True	False	False	False	False	False	False	
6	False	False	False	False	False	True	False	True	False	
7	True	False	True	False	False	False	False	True	False	
8	False	False	True	False	False	False	True	True	False	
9	False	False	True	False	False	False	False	False	True	
10	True	False	False	True	True	True	False	False	False	
11	True	False	False	True	True	True	False	False	False	
12	False	True	False	False	True	False	False	False	False	
13	False	False	True	True	True	False	False	False	False	
14	True	False	True	False	False	False	False	False	False	
15	False	False	False	False	True	True	False	False	False	
16	False	True	True	False	False	False	False	False	False	
17	False	False	True	False	True	False	False	False	False	
18	False	False	True	False	True	False	False	False	False	
19	False	False	False	False	True	True	False	False	True	

In [8]:

```
from mlxtend.frequent_patterns import apriori
```

In [9]:

```
df1 = apriori(df, min_support = 0.2, use_colnames=True, verbose =1)
```

```
Processing 42 combinations | Sampling itemset size 3
```

In [10]:

```
df1
```

Out[10]:

	support	itemsets
0	0.35	(BISCUIT)
1	0.20	(BOURNVITA)
2	0.65	(BREAD)
3	0.40	(COFFEE)
4	0.30	(CORNFLAKES)
5	0.25	(MAGGI)
6	0.25	(MILK)
7	0.30	(SUGER)
8	0.35	(TEA)
9	0.20	(BISCUIT, BREAD)
10	0.20	(MILK, BREAD)
11	0.20	(SUGER, BREAD)
12	0.20	(TEA, BREAD)
13	0.20	(COFFEE, CORNFLAKES)
14	0.20	(COFFEE, SUGER)
15	0.20	(TEA, MAGGI)

In [11]:

```
association_rules(df1, metric = 'confidence', min_threshold = 0.6)
```

Out[11]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(MILK)	(BREAD)	0.25	0.65	0.2	0.800000	1.230769	(
1	(SUGER)	(BREAD)	0.30	0.65	0.2	0.666667	1.025641	(
2	(CORNFLAKES)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	(
3	(SUGER)	(COFFEE)	0.30	0.40	0.2	0.666667	1.666667	(
4	(MAGGI)	(TEA)	0.25	0.35	0.2	0.800000	2.285714	(

The interpretation of a row based on support, confidence, and lift metrics:

- Milk and bread are seen together in 20% of all purchases (support).
- 80% of customers who buy milk also buy bread (confidence).
- Sales of bread increased by 1.23 times in shopping with milk (lift).

According to these results, in order to attract customers' attention to less sold products and to increase the sales of these products, I recommend the grocery market management to place bread and milk on shelves far from each other. I also recommend management placing other targeted products on the shelves in the sections that customers are most likely to pass through to reach milk and bread.