Assignment No. 05

Name : Tejas Shyam Fulumbarkar

Subject : Machine Learning

Class : TE-IT (B)

[2]:
```
!pip install mlxtend
```

```
Requirement already satisfied: mlxtend in /opt/conda/lib/python3.7/site-packages (0.21.0)
Requirement already satisfied: pandas>=0.24.2 in /opt/conda/lib/python3.7/site-packages (from mlxtend) (1.3.5)
Requirement already satisfied: joblib>=0.13.2 in /opt/conda/lib/python3.7/site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from mlxtend) (59.8.0)
Requirement already satisfied: numpy>=1.16.2 in /opt/conda/lib/python3.7/site-packages (from mlxtend) (1.21.6)
Requirement already satisfied: scipy>=1.2.1 in /opt/conda/lib/python3.7/site-packages (from mlxtend) (1.7.3)
Requirement already satisfied: scikit-learn>=1.0.2 in /opt/conda/lib/python3.7/site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: matplotlib>=3.0.0 in /opt/conda/lib/python3.7/site-packages (from mlxtend) (3.5.3)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxtend) (2
1.3)
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxten
d) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxtend)
(1.4.3)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxtend)
(4.33.3)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxtend)
(3.0.9)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxtend) (0.1
1.0)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.7/site-packages (from matplotlib>=3.0.0->mlxtend) (9.
1.1)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas>=0.24.2->mlxtend) (2022.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.7/site-packages (from scikit-learn>=1.0.2->mlxt
end) (3.1.0)
Requirement already satisfied: typing-extensions in /opt/conda/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplotli
b>=3.0.0->mlxtend) (4.4.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.7/site-packages (from python-dateutil>=2.7->matplotlib>=3.
0.0->mlxtend) (1.15.0)
```

[3]:
```
import csv
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

[4]:
```
data = []
with open('../input/market-basket-optimization/Market_Basket_Optimisation.csv') as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        data +=[row]
```

[5]:
```
data[1:10] #list of list
```

[5]:
```
[['burgers', 'meatballs', 'eggs'],
 ['chutney'],
 ['turkey', 'avocado'],
 ['mineral water', 'milk', 'energy bar', 'whole wheat rice', 'green tea'],
 ['low fat yogurt'],
 ['whole wheat pasta', 'french fries'],
 ['soup', 'light cream', 'shallot'],
 ['frozen vegetables', 'spaghetti', 'green tea'],
 ['french fries']]
```

[6]:
```
len(data)
```

[6]:
```
7501
```

[7]:
```
te = TransactionEncoder()
x = te.fit_transform(data)
```

[8]:
```
x
```

[8]:
```
array([[False,  True,  True, ...,  True, False, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       ...,
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False,  True, False]])
```

[9]:
```
te.columns_
```

[9]:
```
[' asparagus',
 'almonds',
 'antioxydant juice',
 'asparagus',
 'avocado',
 'babies food',
 'bacon',
 'barbecue sauce',
 'black tea',
 'blueberries',
 'body spray',
 'bramble',
 'brownies',
 'bug spray',
 'burger sauce',
 'burgers',
 'butter',
 'cake',
 'candy bars',
 'carrots',
 'cauliflower',
 'cereals',
 'champagne',
 'chicken',
 'chili',
 'chocolate',
 'chocolate bread',
 'chutney',
 'cider',
 'clothes accessories',
 'cookies',

 'cream',
 'dessert wine',
 'eggplant',
 'eggs',
 'energy bar',
 'energy drink',
 'escalope',
 'extra dark chocolate',
 'flax seed',
 'french fries',
 'french wine',
 'fresh bread',
 'fresh tuna',
 'fromage blanc',
 'frozen smoothie',
 'frozen vegetables',
 'gluten free bar',
 'grated cheese',
 'green beans',
 'green grapes',
 'green tea',
 'ground beef',
 'gums',
 'ham',
 'hand protein bar',
 'herb & pepper',
 'honey',
 'hot dogs',
 'ketchup',
 'light cream',
 'light mayo',
 'low fat yogurt',
 'magazines',
```

[10]:
```
df = pd.DataFrame(x, columns=te.columns_)
```

[11]:
```
df
```

t[11]:

	asparagus	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	...	turkey	vegetables mix	wate spra
0	False	True	True	False	True	False	False	False	False	False	...	False	True	Fals
1	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals
3	False	False	False	False	True	False	False	False	False	False	...	True	False	Fals
4	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7496	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals
7497	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals
7498	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals
7499	False	False	False	False	False	False	False	False	False	False	...	False	False	Fals

[14]:
```
#Find the rules
rules = association_rules(freq_itemset, metric='confidence', min_threshold=0.10)
```

[15]:
```
rules = rules[['antecedents','consequents','support','confidence']]
rules
```

:[15]:

	antecedents	consequents	support	confidence
0	(avocado)	(mineral water)	0.011598	0.348000
1	(burgers)	(cake)	0.011465	0.131498
2	(cake)	(burgers)	0.011465	0.141447
3	(burgers)	(chocolate)	0.017064	0.195719
4	(chocolate)	(burgers)	0.017064	0.104150
...	...	...	...	...
315	(olive oil)	(mineral water, spaghetti)	0.010265	0.155870
316	(mineral water, pancakes)	(spaghetti)	0.011465	0.339921
317	(mineral water, spaghetti)	(pancakes)	0.011465	0.191964
318	(pancakes, spaghetti)	(mineral water)	0.011465	0.455026
319	(pancakes)	(mineral water, spaghetti)	0.011465	0.120617

320 rows × 4 columns

[16]:
```
rules[rules['antecedents'] == {'cake'}]['consequents']
```

:[16]:
```
2              (burgers)
25           (chocolate)
27               (eggs)
29        (french fries)
30    (frozen vegetables)
33           (green tea)
35               (milk)
37       (mineral water)
39            (pancakes)
41           (spaghetti)
Name: consequents, dtype: object
```