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

```
Requirement already satisfied: mlxtend in c:\users\omnic\anaconda3\lib\site-packages
(0.23.0)
Requirement already satisfied: scipy>=1.2.1 in c:\users\omnic\anaconda3\lib\site-pack
ages (from mlxtend) (1.10.1)
Requirement already satisfied: numpy>=1.16.2 in c:\users\omnic\anaconda3\lib\site-pac
kages (from mlxtend) (1.24.3)
Requirement already satisfied: pandas>=0.24.2 in c:\users\omnic\anaconda3\lib\site-pa
ckages (from mlxtend) (1.5.3)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\omnic\anaconda3\lib\si
te-packages (from mlxtend) (1.2.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\omnic\anaconda3\lib\site
-packages (from mlxtend) (3.7.1)
Requirement already satisfied: joblib>=0.13.2 in c:\users\omnic\anaconda3\lib\site-pa
ckages (from mlxtend) (1.2.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\omnic\anaconda3\lib\site-
packages (from matplotlib>=3.0.0->mlxtend) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\omnic\anaconda3\lib\site-pack
ages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\omnic\anaconda3\lib\site
-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\omnic\anaconda3\lib\site
-packages (from matplotlib>=3.0.0->mlxtend) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\omnic\anaconda3\lib\site-p
ackages (from matplotlib>=3.0.0->mlxtend) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\omnic\anaconda3\lib\site-pac
kages (from matplotlib>=3.0.0->mlxtend) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\omnic\anaconda3\lib\site-
packages (from matplotlib>=3.0.0->mlxtend) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\omnic\anaconda3\lib\s
ite-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\omnic\anaconda3\lib\site-pack
ages (from pandas>=0.24.2->mlxtend) (2022.7)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\omnic\anaconda3\lib\s
ite-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\omnic\anaconda3\lib\site-packages
(from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
```

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

In [5]:
```
dataset = []
with open('Market_Basket_Optimisation.csv')as file:
    reader = csv.reader(file, delimiter=',')
    for row in reader:
        dataset += [row]
```

In [6]:
```
len(dataset)
```

Out[6]:
```
7501
```

In [8]:
```
te = TransactionEncoder()
x = te.fit_transform(dataset)
x
```

Out[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]])
```

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

In [15]:
```
freq_itemset = apriori(df, min_support=0.01, use_colnames=True)
```

In [16]:
```
freq_itemset
```

Out[16]:

| | support | itemsets |
|---|---|---|
| **0** | 0.020397 | (almonds) |
| **1** | 0.033329 | (avocado) |
| **2** | 0.010799 | (barbecue sauce) |
| **3** | 0.014265 | (black tea) |
| **4** | 0.011465 | (body spray) |
| **...** | ... | ... |
| **252** | 0.011065 | (ground beef, mineral water, milk) |
| **253** | 0.017064 | (ground beef, spaghetti, mineral water) |
| **254** | 0.015731 | (spaghetti, mineral water, milk) |
| **255** | 0.010265 | (spaghetti, olive oil, mineral water) |
| **256** | 0.011465 | (spaghetti, mineral water, pancakes) |

257 rows × 2 columns

In [17]:
```
rules = association_rules(freq_itemset, metric= 'confidence', min_threshold=0.25)
```

In [18]:
```
rules
```

Out[18]:

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | con |
|---|---|---|---|---|---|---|---|---|---|
| **0** | (avocado) | (mineral water) | 0.033329 | 0.238368 | 0.011598 | 0.348000 | 1.459926 | 0.003654 | 1. |
| **1** | (burgers) | (eggs) | 0.087188 | 0.179709 | 0.028796 | 0.330275 | 1.837830 | 0.013128 | 1. |
| **2** | (burgers) | (french fries) | 0.087188 | 0.170911 | 0.021997 | 0.252294 | 1.476173 | 0.007096 | 1. |
| **3** | (burgers) | (mineral water) | 0.087188 | 0.238368 | 0.024397 | 0.279817 | 1.173883 | 0.003614 | 1. |
| **4** | (cake) | (mineral water) | 0.081056 | 0.238368 | 0.027463 | 0.338816 | 1.421397 | 0.008142 | 1. |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **90** | (mineral water, milk) | (spaghetti) | 0.047994 | 0.174110 | 0.015731 | 0.327778 | 1.882589 | 0.007375 | 1. |
| **91** | (spaghetti, olive oil) | (mineral water) | 0.022930 | 0.238368 | 0.010265 | 0.447674 | 1.878079 | 0.004799 | 1. |
| **92** | (olive oil, mineral water) | (spaghetti) | 0.027596 | 0.174110 | 0.010265 | 0.371981 | 2.136468 | 0.005460 | 1. |
| **93** | (spaghetti, pancakes) | (mineral water) | 0.025197 | 0.238368 | 0.011465 | 0.455026 | 1.908923 | 0.005459 | 1. |
| **94** | (mineral water, pancakes) | (spaghetti) | 0.033729 | 0.174110 | 0.011465 | 0.339921 | 1.952333 | 0.005593 | 1. |

95 rows × 10 columns

In [19]:
```
rules = rules [['antecedents','consequents','support','confidence']]
```

In [20]:
```
rules.head()
```

Out[20]:

| | antecedents | consequents | support | confidence |
|---|---|---|---|---|
| **0** | (avocado) | (mineral water) | 0.011598 | 0.348000 |
| **1** | (burgers) | (eggs) | 0.028796 | 0.330275 |
| **2** | (burgers) | (french fries) | 0.021997 | 0.252294 |
| **3** | (burgers) | (mineral water) | 0.024397 | 0.279817 |
| **4** | (cake) | (mineral water) | 0.027463 | 0.338816 |

In [22]:
```
rules[rules['antecedents']=={'cake'}]['consequents']
```

Out[22]:
```
4    (mineral water)
Name: consequents, dtype: object
```

```
In [ ]:
```