```
!pip install mlxtend
```

```
Requirement already satisfied: mlxtend in /usr/local/lib/python3.10/dist-packages (0.22.0)
Requirement already satisfied: scipy>=1.2.1 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.11.3)
Requirement already satisfied: numpy>=1.16.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.23.5)
Requirement already satisfied: pandas>=0.24.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.5.3)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.2.2)
Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (3.7.1)
Requirement already satisfied: joblib>=0.13.2 in /usr/local/lib/python3.10/dist-packages (from mlxtend) (1.3.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from mlxtend) (67.7.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlx
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxtend
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->ml
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->ml
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxt
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlxten
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->mlx
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0-
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.24.2->mlxtend) (
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotl
```

```
import csv
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules


dataset = []
with open("Market_Basket_Optimisation.csv") as file:
  reader = csv.reader(file, delimiter=",")
  for row in reader:
    dataset.append(row)


dataset
```

```
 'cider',
 'eggs',
 'honey',
 'cake',
 'green tea',
 'french fries',
 'brownies',
 'tomato juice'],
...]
```

```
te = TransactionEncoder()
```

```
df = te.fit_transform(dataset)
```

```
df = pd.DataFrame(df, columns=te.columns_)
df
```

| | asparagus | almonds | antioxydant juice | asparagus | avocado | babies food | bacon | barbecue sauce | black tea | blueberries | body spray | bramble | bro |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | False | True | True | False | True | False | False | False | False | False | False | False | |
| **1** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **2** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **3** | False | False | False | False | True | False | False | False | False | False | False | False | |
| **4** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **7496** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **7497** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **7498** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **7499** | False | False | False | False | False | False | False | False | False | False | False | False | |
| **7500** | False | False | False | False | False | False | False | False | False | False | False | False | |

7501 rows × 120 columns

```
item_set = apriori(df, min_support=0.01, use_colnames=True)
item_set
```

| | 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 | (spaghetti, ground beef, mineral water) |
| **254** | 0.015731 | (spaghetti, mineral water, milk) |
| **255** | 0.010265 | (spaghetti, mineral water, olive oil) |
| **256** | 0.011465 | (spaghetti, pancakes, mineral water) |

257 rows × 2 columns

```
rules = association_rules(item_set, metric="confidence", min_threshold=0.25)
print(f" Confidence Level : 25%, No of rules : {len(rules)}")
rules
```

```
Confidence Level : 25%, No of rules : 95
```

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction | zhangs_metric |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | (avocado) | (mineral water) | 0.033329 | 0.238368 | 0.011598 | 0.348000 | 1.459926 | 0.003654 | 1.168147 | 0.325896 |
| **1** | (burgers) | (eggs) | 0.087188 | 0.179709 | 0.028796 | 0.330275 | 1.837830 | 0.013128 | 1.224818 | 0.499424 |
| **2** | (burgers) | (french fries) | 0.087188 | 0.170911 | 0.021997 | 0.252294 | 1.476173 | 0.007096 | 1.108844 | 0.353384 |
| **3** | (burgers) | (mineral water) | 0.087188 | 0.238368 | 0.024397 | 0.279817 | 1.173883 | 0.003614 | 1.057552 | 0.162275 |
| **4** | (cake) | (mineral water) | 0.081056 | 0.238368 | 0.027463 | 0.338816 | 1.421397 | 0.008142 | 1.151921 | 0.322617 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **90** | (mineral water, milk) | (spaghetti) | 0.047994 | 0.174110 | 0.015731 | 0.327778 | 1.882589 | 0.007375 | 1.228597 | 0.492451 |
| **91** | (spaghetti, olive oil) | (mineral water) | 0.022930 | 0.238368 | 0.010265 | 0.447674 | 1.878079 | 0.004799 | 1.378954 | 0.478514 |
| **92** | (mineral water, olive oil) | (spaghetti) | 0.027596 | 0.174110 | 0.010265 | 0.371981 | 2.136468 | 0.005460 | 1.315071 | 0.547034 |

```
rules = association_rules(item_set, metric="confidence", min_threshold=0.20)
print(f" Confidence Level : 20%, No of rules : {len(rules)}")
rules
```

```
Confidence Level : 20%, No of rules : 162
```

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction | zhangs_metric |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | (avocado) | (mineral water) | 0.033329 | 0.238368 | 0.011598 | 0.348000 | 1.459926 | 0.003654 | 1.168147 | 0.325896 |
| **1** | (burgers) | (eggs) | 0.087188 | 0.179709 | 0.028796 | 0.330275 | 1.837830 | 0.013128 | 1.224818 | 0.499424 |
| **2** | (burgers) | (french fries) | 0.087188 | 0.170911 | 0.021997 | 0.252294 | 1.476173 | 0.007096 | 1.108844 | 0.353384 |
| **3** | (burgers) | (green tea) | 0.087188 | 0.132116 | 0.017464 | 0.200306 | 1.516139 | 0.005945 | 1.085270 | 0.372947 |
| **4** | (burgers) | (milk) | 0.087188 | 0.129583 | 0.017864 | 0.204893 | 1.581175 | 0.006566 | 1.094717 | 0.402667 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **157** | (mineral water, milk) | (spaghetti) | 0.047994 | 0.174110 | 0.015731 | 0.327778 | 1.882589 | 0.007375 | 1.228597 | 0.492451 |
| **158** | (spaghetti, olive oil) | (mineral water) | 0.022930 | 0.238368 | 0.010265 | 0.447674 | 1.878079 | 0.004799 | 1.378954 | 0.478514 |
| **159** | (mineral water, olive oil) | (spaghetti) | 0.027596 | 0.174110 | 0.010265 | 0.371981 | 2.136468 | 0.005460 | 1.315071 | 0.547034 |
| **160** | (spaghetti, pancakes) | (mineral water) | 0.025197 | 0.238368 | 0.011465 | 0.455026 | 1.908923 | 0.005459 | 1.397557 | 0.488452 |
| **161** | (mineral water, pancakes) | (spaghetti) | 0.033729 | 0.174110 | 0.011465 | 0.339921 | 1.952333 | 0.005593 | 1.251198 | 0.504819 |

162 rows × 10 columns

```
rules = rules[["antecedents", "consequents", "support", "confidence"]]
rules
```

| | antecedents | consequents | support | confidence |
|---|---|---|---|---|
| 159 | (mineral water, olive oil) | (spaghetti) | 0.010265 | 0.371981 |
| 160 | (spaghetti, pancakes) | (mineral water) | 0.011465 | 0.455026 |
| 161 | (mineral water, pancakes) | (spaghetti) | 0.011465 | 0.339921 |

162 rows × 4 columns

```
# Recommendation
ip = "milk"
# ip = input("Enter the item to purchase : ")
rules[rules["antecedents"] == {ip}]["consequents"]
```

```
27        (chocolate)
49             (eggs)
87    (mineral water)
92        (spaghetti)
Name: consequents, dtype: object
```