

```
In [1]: import pandas as pd
```

```
In [2]: df = pd.read_csv(r"D:\Abhishek\ML Practicals\Market_Basket_Optimisation.csv")
```

```
In [3]: df
```

```
Out[3]:
```

	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
7495	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN
7496	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN
7497	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN

7500 rows x 20 columns

```
In [7]: import csv
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

```
In [8]: data = []
with open(r"D:\Abhishek\ML Practicals\Market_Basket_Optimisation.csv") as f:
    reader = csv.reader(f, delimiter=',')
    for row in reader:
        data += [row]
```

```
In [9]: data[1:10]
```

```
Out[9]: [['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']]
```

```
In [10]: len(data)
```

Out[10]: 7501

```
In [15]: t = TransactionEncoder()  
x = t.fit_transform(data)  
x
```

```
Out[15]: 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 [16]: t.columns_
```

```
Out[16]: [' 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',
          'cooking oil',
          'corn',
          'cottage cheese',
          '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',  
'mashed potato',  
'mayonnaise',  
'meatballs',  
'melons',  
'milk',  
'mineral water',  
'mint',  
'mint green tea',  
'muffins',  
'mushroom cream sauce',  
'napkins',  
'nonfat milk',  
'oatmeal',  
'oil',  
'olive oil',  
'pancakes',  
'parmesan cheese',  
'pasta',  
'pepper',  
'pet food',  
'pickles',  
'protein bar',  
'red wine',  
'rice',  
'salad',  
'salmon',  
'salt',  
'sandwich',  
'shallot',  
'shampoo',  
'shrimp',  
'soda',  
'soup',  
'spaghetti',  
'sparkling water',  
'spinach',  
'strawberries',  
'strong cheese',  
'tea',  
'tomato juice',  
'tomato sauce',  
'tomatoes',  
'toothpaste',  
'turkey',  
'vegetables mix',  
'water spray',  
'white wine',  
'whole weat flour',  
'whole wheat pasta',  
'whole wheat rice',  
'yams',  
'yogurt cake',  
'zucchini']
```

```
In [18]: df = pd.DataFrame(x, columns = t.columns_)
```

```
In [19]: df
```

Out [19]:

	asparagus	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	bla t
0	False	True	True	False	True	False	False	False	Fal
1	False	False	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	Fal
3	False	False	False	False	True	False	False	False	Fal
4	False	False	False	False	False	False	False	False	Fal
...	...	...	...	...	...	...	...	...	...
7496	False	False	False	False	False	False	False	False	Fal
7497	False	False	False	False	False	False	False	False	Fal
7498	False	False	False	False	False	False	False	False	Fal
7499	False	False	False	False	False	False	False	False	Fal
7500	False	False	False	False	False	False	False	False	Fal

7501 rows × 120 columns

In [20]: `freq_itemset = apriori(df, min_support = 0.003, use_colnames = True)`In [21]: `freq_itemset`

Out [21]:

	support	itemsets
0	0.020397	(almonds)
1	0.008932	(antioxydant juice)
2	0.004666	(asparagus)
3	0.033329	(avocado)
4	0.004533	(babies food)
...	...	...
1438	0.003066	(ground beef, spaghetti, pancakes, mineral water)
1439	0.003066	(ground beef, tomatoes, spaghetti, mineral water)
1440	0.003333	(olive oil, spaghetti, milk, mineral water)
1441	0.003066	(shrimp, milk, spaghetti, mineral water)
1442	0.003333	(tomatoes, spaghetti, milk, mineral water)

1443 rows × 2 columns

In [27]: `r = association_rules(freq_itemset, metric = 'confidence', min_threshold = 0.5)`In [28]: `r = r[['antecedents', 'consequents', 'support', 'confidence']]`In [29]: `r`

Out [29]:

	antecedents	consequents	support	confidence
0	(almonds)	(burgers)	0.005199	0.254902
1	(almonds)	(chocolate)	0.005999	0.294118
2	(almonds)	(eggs)	0.006532	0.320261
3	(almonds)	(french fries)	0.004399	0.215686
4	(almonds)	(green tea)	0.005066	0.248366
...	...	...	...	...
1357	(tomatoes, spaghetti, milk)	(mineral water)	0.003333	0.568182
1358	(tomatoes, spaghetti, mineral water)	(milk)	0.003333	0.357143
1359	(tomatoes, mineral water, milk)	(spaghetti)	0.003333	0.510204
1360	(milk, spaghetti, mineral water)	(tomatoes)	0.003333	0.211864
1361	(tomatoes, milk)	(spaghetti, mineral water)	0.003333	0.238095

1362 rows × 4 columns

In [30]: `r[r['antecedents'] == {'cake'}]['consequents']`

Out[30]:

```

42      (eggs)
43  (french fries)
44  (mineral water)
45      (spaghetti)
Name: consequents, dtype: object

```

In [ ]: