In [ ]:
```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules
```

In [ ]:
```
df = pd.read_csv("./store_data.csv", header=None)
df.head()

def clean_func(x):
    if isinstance(x, str):
        return x.strip()
    return x
    # print(x)
df.map(clean_func)
```

Out[ ]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie	spinach	olive oil
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7496	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7501 rows × 20 columns

In [ ]:
```
rows = len(df)
cols = len(df.iloc[1,:])
print(rows,cols)

items = set()
for i in range(rows):
    for j in range(cols):
        val = df.iloc[i,j]
        items.add(val)
items.remove(np.nan)
items
```

```
7501 20
```

Out[ ]:
```
{' 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 [ ]:
```
from collections import defaultdict
data = defaultdict(list)
for i in range(rows):
    row_items = set(df.iloc[i , :])
    for product in items:
        if product in row_items:
            data[product].append(1)
        else:
            data[product].append(0)
```

In [ ]:
```
good_data = pd.DataFrame(data).sort_index(axis=1)
good_data.head()
```

Out[ ]:

	asparagus	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	...	turkey	vegetables mix	water spray	white wine	whole weat flour	whole wheat pasta	whole wheat rice	yams	yogurt cake	zucchini
0	0	1	1	0	1	0	0	0	0	0	...	0	1	0	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0

5 rows × 120 columns

In [ ]:
```
frq_items = apriori(good_data, min_support = 0.05, use_colnames = True)
rules = association_rules(frq_items, min_threshold = .07)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
rules
```

```
c:\Users\user\AppData\Local\Programs\Python\Python311\Lib\site-packages\mlxtend\frequent_patterns\fpcommon.py:109: DeprecationWarning: DataFrames with non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with bool type
  warnings.warn(
```

Out[ ]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
4	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314	0.369437
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357	0.308965
2	(eggs)	(mineral water)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815	0.193648
5	(mineral water)	(spaghetti)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008	0.400606
1	(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256	0.339197
3	(mineral water)	(eggs)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158	0.208562