```
import pandas as pd
import numpy as np
import csv
df= pd.read_csv("/content/Market_Basket_Optimisation.csv")
#how the data looks
df.head()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

	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
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
df.shape
```

```
(7500, 20)
```

```
print("DATA INFORMATION AND DATA TYPES")
df.info()
```

```
DATA INFORMATION AND DATA TYPES
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7500 entries, 0 to 7499
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   shrimp             7500 non-null   object
 1   almonds            5746 non-null   object
 2   avocado            4388 non-null   object
 3   vegetables mix     3344 non-null   object
 4   green grapes       2528 non-null   object
 5   whole weat flour   1863 non-null   object
 6   yams               1368 non-null   object
 7   cottage cheese     980 non-null    object
 8   energy drink       653 non-null    object
 9   tomato juice       394 non-null    object
 10  low fat yogurt     255 non-null    object
 11  green tea          153 non-null    object
 12  honey              86 non-null     object
 13  salad              46 non-null     object
 14  mineral water      24 non-null     object
 15  salmon             7 non-null      object
 16  antioxydant juice  3 non-null      object
 17  frozen smoothie    3 non-null      object
 18  spinach            2 non-null      object
 19  olive oil          0 non-null      float64
dtypes: float64(1), object(19)
memory usage: 1.1+ MB
```

```
df.describe()
```

	olive oil
count	0.0
mean	NaN
std	NaN
min	NaN
25%	NaN
50%	NaN
75%	NaN
max	NaN

```
df[0:10]
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

	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 wate
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	whole wheat pasta	french fries	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	soup	light cream	shallot	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	frozen vegetables	spaghetti	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	french fries	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	eggs	pet food	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
len(df)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
7500
```

```
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules


# convert the dataframe into a list of list where each inner list represents a transaction.

df_list = df.to_numpy().tolist()
df_list
dataset = list()
for i in range(len(df_list)) :
    item = list()
    for j in df_list[i] :
        if pd.notna(j):
            item.append(j)
    dataset.append(item)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

```
# Create an instace of our TransactionEncoder cabslass
te = TransactionEncoder()
# Fit and transform our dataset which is a list of lists into an array of True and False.
te_array = te.fit(dataset).transform(dataset)
te_array
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
array([[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],
       [False, False, False, ..., False,  True, False]])
```

```
final_df = pd.DataFrame(te_array,columns=te.columns_)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

```
# Use the apriori algorithm and the min_support for finding out items or group of items which have a support greater than the
```

```
frequent_itemsets_ap = apriori(final_df, min_support=0.01, use_colnames=True)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

```
frequent_itemsets_ap
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

	support	itemsets
0	0.020267	(almonds)
1	0.033200	(avocado)
2	0.010800	(barbecue sauce)
3	0.014267	(black tea)
4	0.011467	(body spray)
...	...	...
254	0.011067	(ground beef, mineral water, milk)
255	0.017067	(spaghetti, ground beef, mineral water)
256	0.015733	(spaghetti, mineral water, milk)
257	0.010267	(olive oil, spaghetti, mineral water)
258	0.011467	(spaghetti, pancakes, mineral water)

259 rows × 2 columns

```
# import association rules class to find association rules amonng the items/group of items which have a support greater than
from mlxtend.frequent_patterns import association_rules

# we have used the metric as confidence and min_threshold to filter out the rules based on these parameters.
rules_ap = association_rules(frequent_itemsets_ap, metric="confidence", min_threshold=0.2)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

```
# Convert the rules obtained into a dataframe for better visualisation
result = pd.DataFrame(rules_ap)
result.sort_values(by='lift',inplace=True,ascending=False)
result
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
75	(herb & pepper)	(ground beef)	0.049467	0.098267	0.016000	0.323450	3.291555	0.011139	1.332841	0.732423
153	(mineral water, spaghetti)	(ground beef)	0.059733	0.098267	0.017067	0.285714	2.907540	0.011197	1.262427	0.697745
69	(tomatoes)	(frozen vegetables)	0.068400	0.095333	0.016133	0.235867	2.474134	0.009613	1.183913	0.639564
67	(shrimp)	(frozen vegetables)	0.071333	0.095333	0.016667	0.233645	2.450820	0.009866	1.180480	0.637444
145	(mineral water, milk)	(frozen vegetables)	0.048000	0.095333	0.011067	0.230556	2.418415	0.006491	1.175740	0.616078
...	...	...	...	...	...	...	...	...	...	...
74	(green tea)	(spaghetti)	0.132000	0.174133	0.026533	0.201010	1.154346	0.003548	1.033638	0.154042
46	(ground beef)	(eggs)	0.098267	0.179733	0.020000	0.203528	1.132388	0.002338	1.029875	0.129651
19	(chocolate)	(eggs)	0.163867	0.179733	0.033200	0.202604	1.127246	0.003748	1.028681	0.135005

```
result=result[['antecedents','consequents','support','confidence']]
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call
  and should_run_async(code)
```

```
result
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `
  and should_run_async(code)
```

	antecedents	consequents	support	confidence
75	(herb & pepper)	(ground beef)	0.016000	0.323450
153	(mineral water, spaghetti)	(ground beef)	0.017067	0.285714
69	(tomatoes)	(frozen vegetables)	0.016133	0.235867
67	(shrimp)	(frozen vegetables)	0.016667	0.233645
145	(mineral water, milk)	(frozen vegetables)	0.011067	0.230556
...	...	...	...	...
74	(green tea)	(spaghetti)	0.026533	0.201010
46	(ground beef)	(eggs)	0.020000	0.203528
19	(chocolate)	(eggs)	0.033200	0.202604
73	(green tea)	(mineral water)	0.030933	0.234343
57	(escalope)	(mineral water)	0.017067	0.215126

162 rows × 4 columns