```
import pandas as pd
!pip install apyori
from apyori import apriori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5954 sha256=12a49e20ac7b4129dba0fe39d4886e1ebc17801f30e9b147beef
  Stored in directory: /root/.cache/pip/wheels/c4/1a/79/20f55c470a50bb3702a8cb7c94d8ada15573538c7f4baebe2d
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

```
data=pd.read_csv("Market_Basket_Optimisation.csv",header=None)
```

```
data.head(10)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1
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	antioxyda juic
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
5	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
6	whole wheat pasta	french fries	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
7	soup	light cream	shallot	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
8	frozen vegetables	spaghetti	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

```
[data.fillna(0,inplace=True) for _ in [data.head()]]
```

```
[None]
```

```
data
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
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	an
1	burgers	meatballs	eggs	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	chutney	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	turkey	avocado	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	mineral water	milk	energy bar	whole wheat rice	green tea	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7496	butter	light mayo	fresh bread	0	0	0	0	0	0	0	0	0	0	0	0	0	
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	0	0	0	0	0	0	0	0	0	0	
7498	chicken	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7499	escalope	green tea	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	0	0	0	0	0	0	0	0	0	0	0	0	

7501 rows × 20 columns

```
tranct=[]
for i in range(0,7501):
  tranct.append([str(data.values[i,j]) for j in range(0,20)])
```

```
tranct[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']
```

```
rl=apriori(tranct,min_support=0.003,min_confidence=0.2,min_lift=3,min_length=2,max_length=2)
rl
```

```
<generator object apriori at 0x7c8ee2127ae0>
```

```
result=list(rl)
```

```
for i in result:
  print(i)
```

```
RelationRecord(items=frozenset({'light cream', 'chicken'}), support=0.004532728969470737, ordered_statistics=[OrderedStatistic(items_bas
RelationRecord(items=frozenset({'mushroom cream sauce', 'escalope'}), support=0.005732568990801226, ordered_statistics=[OrderedStatistic
RelationRecord(items=frozenset({'escalope', 'pasta'}), support=0.005865884548726837, ordered_statistics=[OrderedStatistic(items_base=frc
RelationRecord(items=frozenset({'fromage blanc', 'honey'}), support=0.003332888948140248, ordered_statistics=[OrderedStatistic(items_bas
RelationRecord(items=frozenset({'herb & pepper', 'ground beef'}), support=0.015997866951073192, ordered_statistics=[OrderedStatistic(ite
RelationRecord(items=frozenset({'tomato sauce', 'ground beef'}), support=0.005332622317024397, ordered_statistics=[OrderedStatistic(item
RelationRecord(items=frozenset({'light cream', 'olive oil'}), support=0.003199573390214638, ordered_statistics=[OrderedStatistic(items_b
RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[OrderedStatistic(i
RelationRecord(items=frozenset({'pasta', 'shrimp'}), support=0.005065991201173177, ordered_statistics=[OrderedStatistic(items_base=froze
result=pd.DataFrame(result)
result.head()
```

	items	support	ordered_statistics
0	(light cream, chicken)	0.004533	[((light cream), (chicken), 0.2905982905982905...
1	(mushroom cream sauce, escalope)	0.005733	[((mushroom cream sauce), (escalope), 0.300699...
2	(escalope, pasta)	0.005866	[((pasta), (escalope), 0.3728813559322034, 4.7...
3	(fromage blanc, honey)	0.003333	[((fromage blanc), (honey), 0.2450980392156863...
4	(herb & pepper, ground beef)	0.015998	[((herb & pepper), (ground beef), 0.3234501347...

```
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori,association_rules
te=TransactionEncoder()
te_ary=te.fit(tranct).transform(tranct)
df=pd.DataFrame(te_ary,columns=te.columns_)
df
```

	asparagus	0	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	...	turkey	vegetables mix	water spray	white wine	w f
0	False	False	True	True	False	True	False	False	False	False	...	False	True	False	False	
1	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
2	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
3	False	True	False	False	False	True	False	False	False	False	...	True	False	False	False	F
4	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7496	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
7497	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
7498	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
7499	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F
7500	False	True	False	False	False	False	False	False	False	False	...	False	False	False	False	F

7501 rows × 121 columns

```
freqent_itemsets=apriori(df,min_support=0.003,use_colnames=True)
rules=association_rules(freqent_itemsets,metric="confidence",min_threshold=0.75)
antecedents=rules['antecedents'].apply(lambda x:list(x))
consequents=rules['consequents'].apply(lambda x:list(x))
df_rule=pd.DataFrame({'antecedents':antecedents,'consequents':consequents,'support':rules['support']})
df_rule
```

	antecedents	consequents	support
0	[almonds]	[0]	0.020264
1	[antioxydant juice]	[0]	0.008799
2	[asparagus]	[0]	0.004666
3	[avocado]	[0]	0.033196
4	[babies food]	[0]	0.004533
...	...	...	...
1440	[mineral water, spaghetti, pancakes, ground beef]	[0]	0.003066
1441	[mineral water, spaghetti, tomatoes, ground beef]	[0]	0.003066
1442	[mineral water, olive oil, spaghetti, milk]	[0]	0.003333
1443	[mineral water, spaghetti, milk, shrimp]	[0]	0.003066
1444	[mineral water, spaghetti, milk, tomatoes]	[0]	0.003333

1445 rows × 3 columns

```
rules[rules['antecedents'] == {'cake'}]['consequents']
```

```
     consequents
15           (0)

dtype: object
```