```
import pandas as pd
```

```
ds = pd.read_csv("Market_Basket.csv", header=None)
```

```
!pip install apyori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py): started
  Building wheel for apyori (setup.py): finished with status 'done'
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=597
  Stored in directory: c:\users\admin\appdata\local\pip\cache\wheels\77\3d\
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
from apyori import apriori
```

```
ds.head()
```

	0	1	2	3	4	5	6	7	8	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tor
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	

```
ds.fillna(0,inplace=True)
```

```
#no of transactions in dataset
len(ds)
```

```
7501
```

```
# Convert values to string
ds = ds.astype(str)
ds = ds.astype(str)

# list creation
transactions = ds.values.tolist()
print(transactions)
```

```
[['shrimp', 'almonds', 'avocado', 'vegetables mix', 'green grapes', 'whole
```

```
#set the rules using apriori
rules= apriori(transactions= transactions,min_support=0.002,min_confidance=0.3,
```

```
rules
```

```
<generator object apriori at 0x00000244C52E5FE0>
```

```
Results = list(rules)
```

```
Results
```

```
[RelationRecord(items=frozenset({'light cream', 'chicken'}),
support=0.004532728969470737,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'chicken'}),
items_add=frozenset({'light cream'}), confidence=0.07555555555555556,
lift=4.843950617283951), OrderedStatistic(items_base=frozenset({'light
cream'}), items_add=frozenset({'chicken'}),
confidence=0.29059829059829057, lift=4.84395061728395)]),
 RelationRecord(items=frozenset({'escalope', 'pasta'}),
support=0.005865884548726837,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'escalope'}),
items_add=frozenset({'pasta'}), confidence=0.07394957983193277,
lift=4.700811850163794), OrderedStatistic(items_base=frozenset({'pasta'}),
items_add=frozenset({'escalope'}), confidence=0.3728813559322034,
lift=4.700811850163794)]),
 RelationRecord(items=frozenset({'fresh tuna', 'french wine'}),
support=0.0025329956005865884,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'french
wine'}), items_add=frozenset({'fresh tuna'}),
confidence=0.11242603550295857, lift=5.049746660525103),
OrderedStatistic(items_base=frozenset({'fresh tuna'}),
items_add=frozenset({'french wine'}), confidence=0.11377245508982035,
lift=5.049746660525103)]),
 RelationRecord(items=frozenset({'fromage blanc', 'fresh tuna'}),
support=0.0023996800426609784,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'fresh tuna'}),
items_add=frozenset({'fromage blanc'}), confidence=0.10778443113772455,
lift=7.926382529059528), OrderedStatistic(items_base=frozenset({'fromage
blanc'}), items_add=frozenset({'fresh tuna'}),
confidence=0.1764705882352941, lift=7.926382529059527)]),
 RelationRecord(items=frozenset({'fromage blanc', 'honey'}),
support=0.003332888948140248,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'fromage
blanc'}), items_add=frozenset({'honey'}), confidence=0.2450980392156863,
lift=5.164270764485569), OrderedStatistic(items_base=frozenset({'honey'}),
items_add=frozenset({'fromage blanc'}), confidence=0.0702247191011236,
lift=5.16427076448557)]),
 RelationRecord(items=frozenset({'mushroom cream sauce', 'pasta'}),
 RelationRecord(items=frozenset({'mushroom cream sauce', 'pasta'}),
support=0.0026663111585121984,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'mushroom cream
sauce'}), items_add=frozenset({'pasta'}), confidence=0.13986013986013987,
lift=8.89060092449923), OrderedStatistic(items_base=frozenset({'pasta'}),
items_add=frozenset({'mushroom cream sauce'}),
confidence=0.1694915254237288, lift=8.89060092449923)]),
 RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}),
support=0.007998933475536596,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'olive oil'}),
items_add=frozenset({'whole wheat pasta'}),
confidence=0.12145748987854252, lift=4.1224100976422955),
OrderedStatistic(items_base=frozenset({'whole wheat pasta'}),
items_add=frozenset({'olive oil'}), confidence=0.2714932126696833,
lift=4.122410097642296)]),
 RelationRecord(items=frozenset({'shrimp', 'pasta'}),
support=0.005065991201173177,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}),
items_add=frozenset({'shrimp'}), confidence=0.3220338983050847,
lift=4.506672147735896),
OrderedStatistic(items_base=frozenset({'shrimp'}),
items_add=frozenset({'pasta'}), confidence=0.0708955223880597,
```

```
#dataframe of the Results
df_results = pd.DataFrame(Results)
```

```
df_results.head()
```

	items	support	ordered_statistics
0	(light cream, chicken)	0.004533	[((chicken), (light cream), 0.0755555555555555...
1	(escalope, pasta)	0.005866	[((escalope), (pasta), 0.07394957983193277, 4....
2	(fresh tuna, french wine)	0.002533	[((french wine), (fresh tuna), 0.1124260355029...
3	(fromage blanc, fresh tuna)	0.002400	[((fresh tuna), (fromage blanc), 0.10778443113...
4	(fromage blanc, honey)	0.003333	[((fromage blanc), (honey), 0.2450980392156863...

```
#final output for the assocation rules metrics
#support = frequency of the product
#confidence = reliability of the association rule
#lift = strength of the assocation rule ( >1 = strong, <1 = weak, = 0 => associ
for item in Results:
    pair = item[0]
    items = [x for x in pair]
    print("Rule: " + items[0] + " -> " + items[1])
    print("Support: " + str(item[1]))
    print("Confidence: " + str(item[2][0][2]))
    print("Lift: " + str(item[2][0][3]))
    print("=====================================")
```

```
Rule: light cream -> chicken
Support: 0.004532728969470737
Confidence: 0.07555555555555556
```

```
Lift: 4.843950617283951
=====================================
Rule: escalope -> pasta
Support: 0.005865884548726837
Confidence: 0.07394957983193277
Lift: 4.700811850163794
=====================================
Rule: fresh tuna -> french wine
Support: 0.0025329956005865884
Confidence: 0.11242603550295857
Lift: 5.049746660525103
=====================================
Rule: fromage blanc -> fresh tuna
Support: 0.0023996800426609784
Confidence: 0.10778443113772455
Lift: 7.926382529059528
=====================================
Rule: fromage blanc -> honey
Support: 0.003332888948140248
Confidence: 0.2450980392156863
Lift: 5.164270764485569
=====================================
Rule: mushroom cream sauce -> pasta
Support: 0.0026663111585121984
Confidence: 0.13986013986013987
Lift: 8.89060092449923
=====================================
Rule: whole wheat pasta -> olive oil
Support: 0.007998933475536596
Confidence: 0.12145748987854252
Lift: 4.1224100976422955
=====================================
Rule: shrimp -> pasta
Support: 0.005065991201173177
Confidence: 0.3220338983050847
Lift: 4.506672147735896
=====================================
Rule: rice -> red wine
Support: 0.0025329956005865884
Confidence: 0.09004739336492891
Lift: 4.790393600215118
=====================================
Rule: 0 -> light cream
Support: 0.004532728969470737
Confidence: 0.07555555555555556
Lift: 4.843950617283951
=====================================
Rule: 0 -> escalope
Support: 0.005865884548726837
Confidence: 0.07394957983193277
Lift: 4.700811850163794
=====================================
Rule: fresh tuna -> 0
Support: 0.0025329956005865884
Confidence: 0.11242603550295857
```

Start coding or generate with AI.