

In []:

```
import numpy as np
import pandas as pd
from apyori import apriori
```

In []:

```
basket_data = pd.read_csv('market-basket.csv')
basket = pd.DataFrame(basket_data)
```

In []:

basket

Out[]:

	Item(s)	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10	Item 11	Item 12	Item 13	Item 14	Item 15	Item 16	Item 17
0	4	citrus fruit	semi-finished bread	margarine	ready soups	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	3	tropical fruit	yogurt	coffee	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1	whole milk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	pip fruit	yogurt	cream cheese	meat spreads	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	4	other vegetables	whole milk	condensed milk	long life bakery product	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
9830	17	sausage	chicken	beef	hamburger meat	citrus fruit	grapes	root vegetables	whole milk	butter	whipped/sour cream	flour	coffee	red/blush wine	salty snack	chocolate	hygiene articles	napkin
9831	1	cooking chocolate	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9832	10	chicken	citrus fruit	other vegetables	butter	yogurt	frozen dessert	domestic eggs	rolls/buns	rum	cling film/bags	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9833	4	semi-finished bread	bottled water	soda	bottled beer	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9834	5	chicken	tropical fruit	other vegetables	vinegar	shopping bags	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

9835 rows x 33 columns



In []:

```
basketContents = []
for i in range(0,1000):
    basketContents.append([str(basket.values[i,j]) for j in range(1,33)])
```

In []:

```
association_rules = apriori(basketContents, min_support=0.015, min_confidence=0.2, min_lift=3, min_length=2)
association_results = list(association_rules)
```

In []:

```
print(len(association_results))
```

10

In [22]:

```
df_results = pd.DataFrame(association_results)
```

Out[22]:

```
[OrderedStatistic(items_base=frozenset({'berries'}), items_add=frozenset({'whipped/sour cream'}), confidence=0.3617021276595745, lift=4.88786658999425),
 OrderedStatistic(items_base=frozenset({'whipped/sour cream'}), items_add=frozenset({'berries'}), confidence=0.22972972972972977, lift=4.887866589994251)]
```

In [23]:

```
first_values = []
second_values = []
third_values = []
fourth_value = []

for i in range(df_results.shape[0]):
    single_list = df_results['ordered_statistics'][i][0]
    first_values.append(list(single_list[0]))
    second_values.append(list(single_list[1]))
    third_values.append(single_list[2])
    fourth_value.append(single_list[3])

support = pd.DataFrame(df_results['support'])
```

In [24]:

```
lhs = pd.DataFrame(first_values)
rhs = pd.DataFrame(second_values)
confidence = pd.DataFrame(third_values,columns=['confidence'])
lift = pd.DataFrame(fourth_value,columns=['lift'])
```

In [25]:

```
df_final = pd.concat([lhs,rhs,support,confidence,lift], axis=1)
```

In [26]:

```
df_final.fillna(value=' ', inplace=True)
df_final.columns = ['lhs',0,1,2,'rhs','suppport','confidence','lift']
```

In [27]:

```
df_final['lhs'] = df_final['lhs']+str(", ")+df_final[1]+str(", ")+df_final[2]
```

In [33]:

```
df_final.drop(columns=[0,1,2],axis=1,inplace=True)
```

In [34]:

```
df_final.columns = ['items_base','items_add','support','confidence','lift']
```

In [35]:

```
df_final
```

Out[35]:

	items_base	items_add	support	confidence	lift
0	berries, whipped/sour cream,		0.017	0.361702	4.887867
1	butter, root vegetables,		0.017	0.369565	3.359684
2	berries, nan, whipped/sour cream		0.017	0.361702	4.887867
3	butter, nan, root vegetables		0.017	0.369565	3.359684
4	curd, whole milk, yogurt		0.016	0.213333	3.809524
5	fruit/vegetable juice, whole milk, yogurt		0.015	0.223881	3.997868
6	whole milk, yogurt,		0.015	0.405405	3.192169
7	curd, whole milk, nan	yogurt	0.016	0.213333	3.809524
8	fruit/vegetable juice, whole milk, nan	yogurt	0.015	0.223881	3.997868
9	whole milk, yogurt, nan		0.015	0.405405	3.192169