Download

```
!pip3 install apyori
```

```
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=5955 sha256=91cd68a17de4d7734837423e8d1cbb94d9d9
  Stored in directory: /root/.cache/pip/wheels/c4/1a/79/20f55c470a50bb3702a8cb7c94d8ada15573538c7f4baebe2d
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

Libraries

+ Code + Text

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import sklearn
```

New Section

```
dataset = pd.read_csv('/content/Big Basket Cart Prediction.csv')
dataset

transactions = []

for i in range (0,7218):
  transactions.append([str(dataset.values[i,j]) for j in range (0,20)])


transactions
```

```
[['knor',
  'ginger garlic paste',
  'MTR Idli',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan'],
 ['turkey',
  'spirit fish',
  'tomatoes',
  'spaghetti',
  'mineral water',
  'patanjali tea',
  'salmon',
  'eggs',
  'chicken',
  'cadbury white chocolate',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan',
  'nan'],
 ['eggs',
  'nan',
```

```
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan',
        'nan'
```

Priori Algorithm on dataset

```
from apyori import apriori
```

```
basket_intelligence = apriori(transactions = transactions , min_support = 0.002 , min_confidence = 0.2 , min_lift = 3 , min
```

Raw Results

```
results = list(basket_intelligence)
```

```
results[:5]
```

```
[RelationRecord(items=frozenset({'pancakes', '5 star'}), support=0.0022166805209199226, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'5 star'}), items_add=frozenset({'pancakes'}), confidence=0.326530612244898,
lift=3.445757250268529)]),
 RelationRecord(items=frozenset({'parle g', 'MTR Idli'}), support=0.0020781379883624274, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'MTR Idli'}), items_add=frozenset({'parle g'}), confidence=0.26315789473684215,
lift=3.2469635627530367)]),
 RelationRecord(items=frozenset({'almonds', 'burgers'}), support=0.005403158769742311, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'almonds'}), items_add=frozenset({'burgers'}), confidence=0.26530612244897955,
lift=3.059072830410119)]),
 RelationRecord(items=frozenset({'barbecue sauce', 'turkey'}), support=0.0024937655860349127, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'barbecue sauce'}), items_add=frozenset({'turkey'}),
confidence=0.22499999999999998, lift=3.585099337748344)]),
 RelationRecord(items=frozenset({'buns', 'paneer'}), support=0.016070933776669436, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'buns'}), items_add=frozenset({'paneer'}), confidence=0.3240223463687151,
lift=3.2848220450693617)])]
```

Proper Display

```
def inspect(results):
  product1    =      [tuple(result[2][0][0])[0] for result in results]
  product2    =      [tuple(result[2][0][1])[0] for result in results]
  supports    =      [result[1] for result in results]
  confidences =      [result[2][0][2] for result in results]
  lifts       =      [result[2][0][3] for result in results]

  return list(zip(product1 , product2 , supports , confidences , lifts))
```

```
intelligence = inspect(results)
columns      = ['product1' , 'product2' , 'supports' , 'confidences' , 'lifts']
```

```
DataFrame_intelligence = pd.DataFrame(intelligence , columns = columns)
```

```
DataFrame_intelligence
```

	product1	product2	supports	confidences	lifts
0	5 star	pancakes	0.002217	0.326531	3.445757
1	MTR Idli	parle g	0.002078	0.263158	3.246964
2	almonds	burgers	0.005403	0.265306	3.059073
3	barbecue sauce	turkey	0.002494	0.225000	3.585099
4	buns	paneer	0.016071	0.324022	3.284822
...	...	...	...	...	...
487	mineral water	nan	0.003048	0.333333	3.492017
488	spaghetti	nan	0.002217	0.355556	3.724819
489	tomatoes	vegetable basket	0.002217	0.307692	10.886878
490	milk	mineral water	0.002078	0.326087	9.017991
491	milk	mineral water	0.002078	0.258621	4.261927

492 rows × 5 columns

```
DataFrame_intelligence.nlargest( n=10 , columns = 'lifts')
```

	product1	product2	supports	confidences	lifts
146	pasta	pepper spray	0.002494	0.461538	23.966796
374	pasta	nan	0.002494	0.461538	23.966796
391	tomatoes	vegetable basket	0.002217	0.307692	10.886878
489	tomatoes	vegetable basket	0.002217	0.307692	10.886878
420	milk	mineral water	0.002078	0.326087	9.017991
490	milk	mineral water	0.002078	0.326087	9.017991
329	maggi	vegetable basket	0.002494	0.219512	7.766858
461	maggi	vegetable basket	0.002494	0.219512	7.766858
220	chicken	milk	0.002078	0.277778	7.652672
407	chicken	milk	0.002078	0.277778	7.652672