

```
In [11]: #importing the required datasets
import numpy as np
import pandas as pd
from apyori import apriori
```

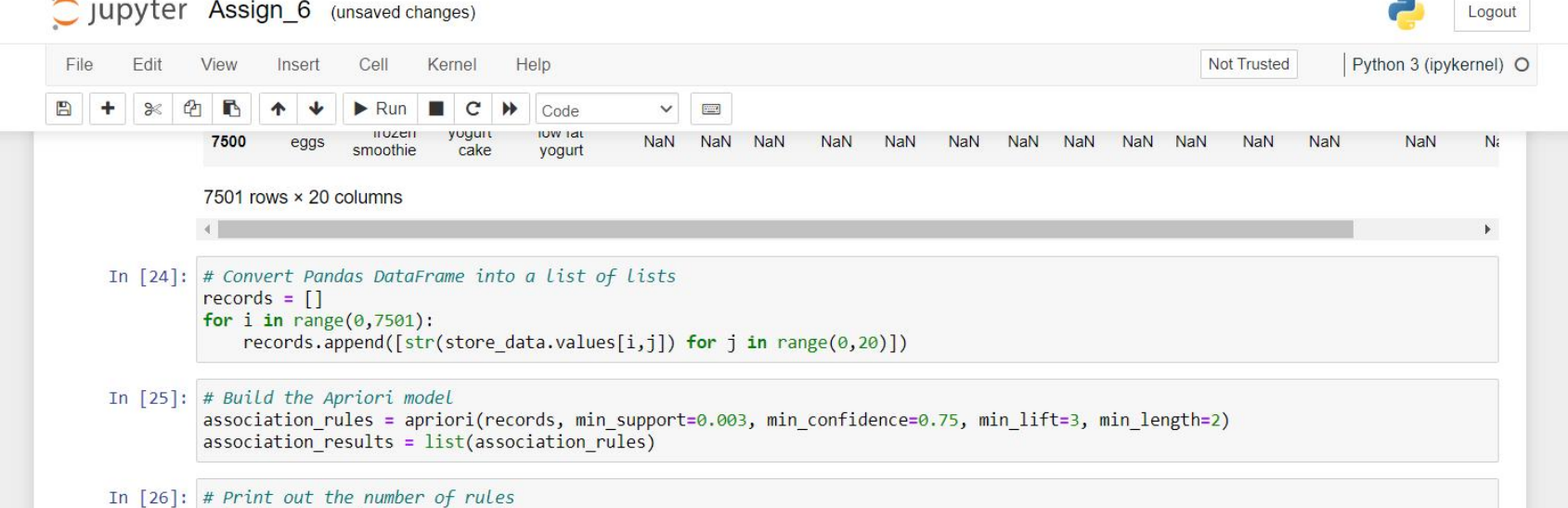
```
In [12]: store_data = pd.read_csv('Market_Basket_Optimisation.csv', header = None)
```

```
In [13]: #Have a glance at the records
store_data
```

```
Out[13]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	froz smooth
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
7496	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7501 rows x 20 columns



The screenshot shows a Jupyter Notebook titled "Assign_6 (unsaved changes)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Help), a toolbar with icons for saving, running, and other actions, and a status bar indicating "Not Trusted" and "Python 3 (ipykernel)".

The notebook content includes a Pandas DataFrame with 7501 rows and 20 columns. The columns are: eggs, frozen smoothie, yogurt cake, low fat yogurt, and 17 columns of NaN values. The first row of data is shown.

Below the DataFrame, there are four code cells:

```
In [24]: # Convert Pandas DataFrame into a list of lists
records = []
for i in range(0,7501):
    records.append([str(store_data.values[i,j]) for j in range(0,20)])
```

```
In [25]: # Build the Apriori model
association_rules = apriori(records, min_support=0.003, min_confidence=0.75, min_lift=3, min_length=2)
association_results = list(association_rules)
```

```
In [26]: # Print out the number of rules
print(len(association_results))
```

2

```
In [27]: # Glancing at the first rule
print(association_results)
```

[RelationRecord(items=frozenset({'frozen vegetables', 'soup', 'milk', 'mineral water'}), support=0.0030662578322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'frozen vegetables', 'soup', 'milk'}), items_add=frozenset({'mineral water'}), confidence=0.7666666666666666, lift=3.21631245339299)], RelationRecord(items=frozenset({'frozen vegetables', 'mineral water', 'nan', 'soup', 'milk'}), support=0.0030662578322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'frozen vegetables', 'soup', 'milk'}), items_add=frozenset({'mineral water', 'nan'}), confidence=0.7666666666666666, lift=3.218112292482745), OrderedStatistic(items_base=frozenset({'frozen vegetables', 'soup', 'milk', 'nan'}), items_add=frozenset({'mineral water'}), confidence=0.7666666666666666, lift=3.21631245339299)])]

```
In [28]: for item in association_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]))
```

LP1_NBA Lab Manual_2022-23.p...Desktop/ML/ML_LAB/Assign_6 - Jupyter Notebook

localhost:8888/notebooks/Desktop/ML/ML_LAB/Assign_6.ipynb

Logout

Jupyter Assign_6 (unsaved changes)

FileEditViewInsertCellKernelHelp

Not TrustedPython 3 (ipykernel)

2

In [27]:

Glancng at the first rule
print(association_results)

[RelationRecord(items=frozenset({'frozen vegetables', 'soup', 'milk', 'mineral water'}), support=0.0030662578322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'frozen vegetables', 'soup', 'milk'}), items_add=frozenset({'mineral water', 'nan', 'soup', 'milk'}), support=0.0030662578322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'frozen vegetables', 'soup', 'milk'}), items_add=frozenset({'mineral water', 'nan'}), confidence=0.7666666666666666, lift=3.218112292482745), OrderedStatistic(items_base=frozenset({'frozen vegetables', 'soup', 'milk', 'nan'}), items_add=frozenset({'mineral water'}), confidence=0.7666666666666666, lift=3.21631245339299)])]

In [28]:

for item in association_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: frozen vegetables -> soup
Support: 0.0030662578322890282
Confidence: 0.7666666666666666
Lift: 3.21631245339299
=====
Rule: frozen vegetables -> mineral water
Support: 0.0030662578322890282
Confidence: 0.7666666666666666
Lift: 3.218112292482745
=====

In []: