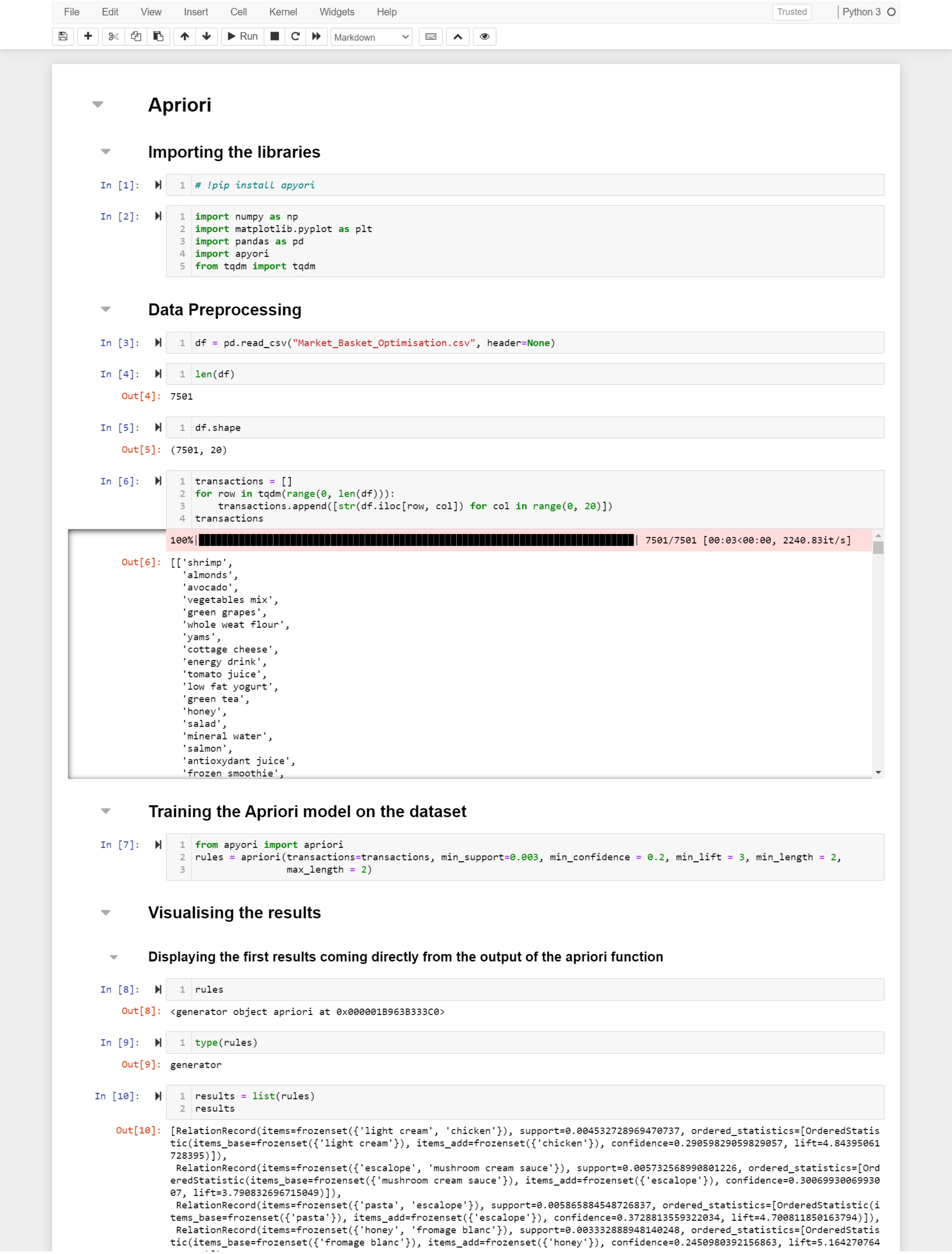

Apriori

Importing the libraries

In [1]:
```
# !pip install apyori
```

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import apyori
from tqdm import tqdm
```

Data Preprocessing

In [3]:
```
df = pd.read_csv("Market_Basket_Optimisation.csv", header=None)
```

In [4]:
```
len(df)
```

Out[4]:
```
7501
```

In [5]:
```
df.shape
```

Out[5]:
```
(7501, 20)
```

In [6]:
```
transactions = []
for row in tqdm(range(0, len(df))):
    transactions.append([str(df.iloc[row, col]) for col in range(0, 20)])
transactions
```

```
100%|██████████████████████████████████████████████████████████████████████████| 7501/7501 [00:03<00:00, 2240.83it/s]
```

Out[6]:
```
[['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',
```

Training the Apriori model on the dataset

In [7]:
```
from apyori import apriori
rules = apriori(transactions=transactions, min_support=0.003, min_confidence = 0.2, min_lift = 3, min_length = 2,
                max_length = 2)
```

Visualising the results

Displaying the first results coming directly from the output of the apriori function

In [8]:
```
rules
```

Out[8]:
```
<generator object apriori at 0x000001B963B333C0>
```

In [9]:
```
type(rules)
```

Out[9]:
```
generator
```

In [10]:
```
results = list(rules)
results
```

Out[10]:
```
[RelationRecord(items=frozenset({'light cream', 'chicken'}), support=0.004532728969470737, ordered_statistics=[OrderedStatis
tic(items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}), confidence=0.29059829059829057, lift=4.84395061
728395)]),
 RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[Ord
eredStatistic(items_base=frozenset({'mushroom cream sauce'}), items_add=frozenset({'escalope'}), confidence=0.30069930069930
07, lift=3.790832696715049)]),
 RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.005865884548726837, ordered_statistics=[OrderedStatistic(i
tems_base=frozenset({'pasta'}), items_add=frozenset({'escalope'}), confidence=0.3728813559322034, lift=4.700811850163794)]),
 RelationRecord(items=frozenset({'honey', 'fromage blanc'}), support=0.003332888948140248, ordered_statistics=[OrderedStatis
tic(items_base=frozenset({'fromage blanc'}), items_add=frozenset({'honey'}), confidence=0.245098039215686, lift=5.164270764
485569)]),
 RelationRecord(items=frozenset({'herb & pepper', 'ground beef'}), support=0.015997866951073192, ordered_statistics=[Ordered
Statistic(items_base=frozenset({'herb & pepper'}), items_add=frozenset({'ground beef'}), confidence=0.3234501347708895, lift
=3.2919938411349285)]),
 RelationRecord(items=frozenset({'tomato sauce', 'ground beef'}), support=0.005332622317024397, ordered_statistics=[OrderedS
tatistic(items_base=frozenset({'tomato sauce'}), items_add=frozenset({'ground beef'}), confidence=0.3773584905660377, lift=
3.840659481324083)]),
 RelationRecord(items=frozenset({'light cream', 'olive oil'}), support=0.003199573390214638, ordered_statistics=[OrderedStat
istic(items_base=frozenset({'light cream'}), items_add=frozenset({'olive oil'}), confidence=0.20512820512820515, lift=3.1147
098515519573)]),
 RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[Order
edStatistic(items_base=frozenset({'whole wheat pasta'}), items_add=frozenset({'olive oil'}), confidence=0.2714932126696833,
lift=4.122410097642296)]),
 RelationRecord(items=frozenset({'pasta', 'shrimp'}), support=0.005065991201173177, ordered_statistics=[OrderedStatistic(ite
ms_base=frozenset({'pasta'}), items_add=frozenset({'shrimp'}), confidence=0.3220338983050847, lift=4.506672147735896)])]
```

In [11]:

```
list(results[0][2][0][0])[0]
```

Out[11]:

```
'light cream'
```

In [12]:

```
list(results[0][2][0][1])[0]
```

Out[12]:

```
'chicken'
```

Putting the results well organised into a Pandas DataFrame

In [13]:

```
def inspect(results):
    lhs         = [list(result[2][0][0])[0] for result in results]
    rhs         = [list(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(lhs, rhs, supports, confidences, lifts))
resultsinDataFrame = pd.DataFrame(inspect(results), columns = ['Left Hand Side', 'Right Hand Side', 'Support', 'Confiden
```

Displaying the results non sorted

In [14]:

```
resultsinDataFrame
```

Out[14]:

	Left Hand Side	Right Hand Side	Support	Confidence	Lift
0	light cream	chicken	0.004533	0.290598	4.843951
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
2	pasta	escalope	0.005866	0.372881	4.700812
3	fromage blanc	honey	0.003333	0.245098	5.164271
4	herb & pepper	ground beef	0.015998	0.323450	3.291994
5	tomato sauce	ground beef	0.005333	0.377358	3.840659
6	light cream	olive oil	0.003200	0.205128	3.114710
7	whole wheat pasta	olive oil	0.007999	0.271493	4.122410
8	pasta	shrimp	0.005066	0.322034	4.506672

Displaying the results sorted by descending lifts

In [15]:

```
resultsinDataFrame.nlargest(n=10, columns="Lift")
```

Out[15]:

	Left Hand Side	Right Hand Side	Support	Confidence	Lift
3	fromage blanc	honey	0.003333	0.245098	5.164271
0	light cream	chicken	0.004533	0.290598	4.843951
2	pasta	escalope	0.005866	0.372881	4.700812
8	pasta	shrimp	0.005066	0.322034	4.506672
7	whole wheat pasta	olive oil	0.007999	0.271493	4.122410
5	tomato sauce	ground beef	0.005333	0.377358	3.840659
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
4	herb & pepper	ground beef	0.015998	0.323450	3.291994
6	light cream	olive oil	0.003200	0.205128	3.114710

In [16]:

```
resultsinDataFrame.sort_values(by=["Lift"], axis=0, ascending=False)
```

Out[16]:

	Left Hand Side	Right Hand Side	Support	Confidence	Lift
3	fromage blanc	honey	0.003333	0.245098	5.164271
0	light cream	chicken	0.004533	0.290598	4.843951
2	pasta	escalope	0.005866	0.372881	4.700812
8	pasta	shrimp	0.005066	0.322034	4.506672
7	whole wheat pasta	olive oil	0.007999	0.271493	4.122410
5	tomato sauce	ground beef	0.005333	0.377358	3.840659
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
4	herb & pepper	ground beef	0.015998	0.323450	3.291994

6	light cream	olive oil	0.003200	0.205128	3.114710

Eclat

Importing the libraries

In [17]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import apyori
from tqdm import tqdm
```

Data Preprocessing

In [18]:
```
df = pd.read_csv("Market_Basket_Optimisation.csv", header=None)
```

In [19]:
```
len(df)
```

Out[19]:
```
7501
```

In [20]:
```
df.shape
```

Out[20]:
```
(7501, 20)
```

In [21]:
```
transactions = []
for row in tqdm(range(0, len(df))):
    transactions.append([str(df.iloc[row, col]) for col in range(0, 20)])
transactions
```

```
100%|██████████████████████████████████████████████████████████████████████████| 7501/7501 [00:03<00:00, 2209.57it/s]
```

Out[21]:
```
[['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',
```

Training the Eclat model on the dataset

In [22]:
```
from apyori import apriori
rules = apriori(transactions=transactions, min_support=0.003, min_confidence = 0.2, min_lift = 3, min_length = 2,
                max_length = 2)
```

Visualising the results

Displaying the first results coming directly from the output of the Eclat function

In [23]:
```
rules
```

Out[23]:
```
<generator object apriori at 0x000001B9646BB2E0>
```

In [24]:
```
type(rules)
```

Out[24]:
```
generator
```

In [25]:
```
results = list(rules)
results
```

Out[25]:
```
[RelationRecord(items=frozenset({'light cream', 'chicken'}), support=0.004532728969470737, ordered_statistics=[OrderedStatis
tic(items_base=frozenset({'light cream'}), items_add=frozenset({'chicken'}), confidence=0.29059829059829057, lift=4.84395061
728395)]),
 RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[Ord
eredStatistic(items_base=frozenset({'mushroom cream sauce'}), items_add=frozenset({'escalope'}), confidence=0.30069930069930
07, lift=3.790832696715049)]),
 RelationRecord(items=frozenset({'pasta', 'escalope'}), support=0.005865884548726837, ordered_statistics=[OrderedStatistic(i
tems_base=frozenset({'pasta'}), items_add=frozenset({'escalope'}), confidence=0.3728813559322034, lift=4.700811850163794)]),
 RelationRecord(items=frozenset({'honey', 'fromage blanc'}), support=0.003332888948140248, ordered_statistics=[OrderedStatis
tic(items_base=frozenset({'fromage blanc'}), items_add=frozenset({'honey'}), confidence=0.245098039215686３, lift=5.164270764
485569)]),
 RelationRecord(items=frozenset({'herb & pepper', 'ground beef'}), support=0.015997866951073192, ordered_statistics=[Ordered
Statistic(items_base=frozenset({'herb & pepper'}), items_add=frozenset({'ground beef'}), confidence=0.323450134770889５, lift
=3.2919938411349285)]),
 RelationRecord(items=frozenset({'tomato sauce', 'ground beef'}), support=0.005332622317024397, ordered_statistics=[OrderedS
tatistic(items_base=frozenset({'tomato sauce'}), items_add=frozenset({'ground beef'}), confidence=0.3773584905660377, lift=
3.840659481324083)]),
 RelationRecord(items=frozenset({'light cream', 'olive oil'}), support=0.003199573390214638, ordered_statistics=[OrderedStat
istic(items_base=frozenset({'light cream'}), items_add=frozenset({'olive oil'}), confidence=0.2051282051282515, lift=3.1147
098515519573)]),
 RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[Order
edStatistic(items_base=frozenset({'whole wheat pasta'}), items_add=frozenset({'olive oil'}), confidence=0.2714932126696833,
lift=4.122410097642296)]),
 RelationRecord(items=frozenset({'pasta', 'shrimp'}), support=0.005065991201173177, ordered_statistics=[OrderedStatistic(ite
ms_base=frozenset({'pasta'}), items_add=frozenset({'shrimp'}), confidence=0.3220338983050847, lift=4.506672147735896)])]
```

In [26]:

```python
list(results[0][2][0][0])[0]
```

Out[26]:

```
'light cream'
```

In [27]:

```python
list(results[0][2][0][1])[0]
```

Out[27]:

```
'chicken'
```

Putting the results well organised into a Pandas DataFrame

In [28]:

```python
def inspect(results):
    lhs         = [tuple(result[2][0][0])[0] for result in results]
    rhs         = [tuple(result[2][0][1])[0] for result in results]
    supports    = [result[1] for result in results]
    return list(zip(lhs, rhs, supports))
resultsinDataFrame = pd.DataFrame(inspect(results), columns = ['Product 1', 'Product 2', 'Support'])
```

Displaying the results non sorted

In [29]:

```python
resultsinDataFrame
```

Out[29]:

	Product 1	Product 2	Support
0	light cream	chicken	0.004533
1	mushroom cream sauce	escalope	0.005733
2	pasta	escalope	0.005866
3	fromage blanc	honey	0.003333
4	herb & pepper	ground beef	0.015998
5	tomato sauce	ground beef	0.005333
6	light cream	olive oil	0.003200
7	whole wheat pasta	olive oil	0.007999
8	pasta	shrimp	0.005066

Displaying the results sorted by descending lifts

In [30]:

```python
resultsinDataFrame.nlargest(n=10, columns="Support")
```

Out[30]:

	Product 1	Product 2	Support
4	herb & pepper	ground beef	0.015998
7	whole wheat pasta	olive oil	0.007999
2	pasta	escalope	0.005866
1	mushroom cream sauce	escalope	0.005733
5	tomato sauce	ground beef	0.005333
8	pasta	shrimp	0.005066
0	light cream	chicken	0.004533
3	fromage blanc	honey	0.003333
6	light cream	olive oil	0.003200

In [31]:

```python
resultsinDataFrame.sort_values(by=["Support"], axis=0, ascending=False)
```

Out[31]:

	Product 1	Product 2	Support
4	herb & pepper	ground beef	0.015998
7	whole wheat pasta	olive oil	0.007999
2	pasta	escalope	0.005866
1	mushroom cream sauce	escalope	0.005733
5	tomato sauce	ground beef	0.005333
8	pasta	shrimp	0.005066
0	light cream	chicken	0.004533
3	fromage blanc	honey	0.003333
6	light cream	olive oil	0.003200

Apriori is more powerful than Eclat and Eclat is subset of Apriori