AssociationRule_learning_ap_kelvin

September 11, 2022

1 Learning Association Rules of Transactions from a Retail Shop

1.0.1 In this project I used apriori association rule machine learning on a large list of transactions from a retail shop to easily identify a set of products frequently bought together. This tool seeks to guide any business owner on decisions relating to continuos production, promotional pricing, product placement, fraud detection etc. This machine learning algorithm serves as a recommendation system which answers the question... "People who bought .... also bought that...."

1.1 Importing the libraries

[2]:
```
# Install apyori library
# !pip install apyori
```

[3]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

1.2 Data Preprocessing

[4]:
```
dataset = pd.read_csv('Market_Basket_Optimisation.csv', header = None) # no␣
 ↪headers in dataframe

#convert pandas dataframe to a list of transactions
transactions = []
for i in range(0, 7501):
  transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])
```

1.3 Training the Apriori model on the dataset

[5]:
```
from apyori import apriori #import apriori library
rules = apriori(transactions = transactions,
                min_support = 0.003, # product appears in atleast 3% of the time
                min_confidence = 0.2,
                min_lift = 3,  # lift = rule of thumb = lifts below 3 makes␣
 ↪rule irrelevant
                min_length = 2, max_length = 2)  # two products in our rules
```

```
# min_support = 21/7501 = 0.00279 : products that appear in atleast 3 times a␣
 ↪day =  21 times a week:
# min_confidence = precision for rule to be correct = accurracy of correct rule
# lift = rule of thumb = lifts below 3 makes rule irrelevant
# min & max length = buy one get one free % = two products in our rule
```

1.4 Visualising the results

1.5 Displaying the first results (raw results) coming directly from the output of the apriori function

[6]:
```
results = list(rules)
```

[7]:
```
results
```

[7]:
```
[RelationRecord(items=frozenset({'chicken', 'light cream'}),
support=0.004532728969470737,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}),
items_add=frozenset({'chicken'}), confidence=0.29059829059829057,
lift=4.84395061728395)]),
 RelationRecord(items=frozenset({'mushroom cream sauce', 'escalope'}),
support=0.005732568990801226,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'mushroom cream
sauce'}), items_add=frozenset({'escalope'}), confidence=0.3006993006993007,
lift=3.790832696715049)]),
 RelationRecord(items=frozenset({'pasta', 'escalope'}),
support=0.005865884548726837,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}),
items_add=frozenset({'escalope'}), confidence=0.3728813559322034,
lift=4.700811850163794)]),
 RelationRecord(items=frozenset({'honey', 'fromage blanc'}),
support=0.003332888948140248,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'fromage blanc'}),
items_add=frozenset({'honey'}), confidence=0.2450980392156863,
lift=5.164270764485569)]),
 RelationRecord(items=frozenset({'ground beef', 'herb & pepper'}),
support=0.015997866951073192,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'herb & pepper'}),
items_add=frozenset({'ground beef'}), confidence=0.3234501347708895,
lift=3.2919938411349285)]),
 RelationRecord(items=frozenset({'ground beef', 'tomato sauce'}),
support=0.005332622317024397,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'tomato sauce'}),
items_add=frozenset({'ground beef'}), confidence=0.3773584905660377,
lift=3.840659481324083)]),
 RelationRecord(items=frozenset({'olive oil', 'light cream'}),
```

```
support=0.003199573390214638,
ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}),
items_add=frozenset({'olive oil'}), confidence=0.20512820512820515,
lift=3.1147098515519573)]),
 RelationRecord(items=frozenset({'olive oil', 'whole wheat pasta'}),
support=0.007998933475536596,
ordered_statistics=[OrderedStatistic(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(items_base=frozenset({'pasta'}),
items_add=frozenset({'shrimp'}), confidence=0.3220338983050847,
lift=4.506672147735896)])]
```

1.5.1 Putting the results well organised into a Pandas DataFrame

[8]:
```
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]
    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', 'Confidence', 'Lift'])
```

1.5.2 Displaying the results non sorted

[9]:
```
resultsinDataFrame
```

[9]:

	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

1.5.3 Displaying the results sorted by descending lifts

[10]:
```
resultsinDataFrame.nlargest(n = 10, columns = 'Lift')
```

[10]:

	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

2 Conclusion:

2.0.1 Customers who bought "fromage blanc" also bought honey with approximately 25% chance, this rule appears in 0.0033 of the transactions with a lift value of 5.164.

2.0.2 NOTE: "LIFT" is the most important metric to measure strength of a rule.