

Assignment on Association Rule Learning

Download Market Basket Optimization dataset from below link. Data Set:

<https://www.kaggle.com/hemanthkumar05/market-basket-optimization>.

This dataset comprises the list of transactions of a retail company over the period of one week. It contains a total of 7501 transaction records where each record consists of the list of items sold in one transaction. Using this record of transactions and items in each transaction, find the association rules between items. There is no header in the dataset and the first row contains the first transaction, so mentioned header = None here while loading dataset. Follow following steps:

1. Data Preprocessing
2. Generate the list of transactions from the dataset
3. Train Apriori algorithm on the dataset
4. Visualize the list of rules
5. Generated rules depend on the values of hyper parameters. By increasing the minimum confidence value and find the rules accordingly

Importing Libraries

```
In [2]: import pandas as pd
```

Loading the dataset

```
In [3]: A = pd.read_csv(r"C:\Users\RASIKA\Downloads\archive (10)\Market_Basket_Optimisation")
```

```
In [4]: A.head()
```

```
Out[4]:
```

	0	1	2	3	4	5	6	7	8	9	10
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	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

```
In [5]: len(A)
```

```
Out[5]: 7501
```

Data Preprocessing

In [6]: `A.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7501 entries, 0 to 7500
Data columns (total 20 columns):
#   Column  Non-Null Count  Dtype
---  -
0    0      7501 non-null     object
1    1      5747 non-null     object
2    2      4389 non-null     object
3    3      3345 non-null     object
4    4      2529 non-null     object
5    5      1864 non-null     object
6    6      1369 non-null     object
7    7       981 non-null     object
8    8       654 non-null     object
9    9       395 non-null     object
10   10      256 non-null     object
11   11      154 non-null     object
12   12       87 non-null     object
13   13       47 non-null     object
14   14       25 non-null     object
15   15        8 non-null   object
16   16        4 non-null   object
17   17        4 non-null   object
18   18        3 non-null   object
19   19        1 non-null   object
dtypes: object(20)
memory usage: 1.1+ MB
```

In [7]: `A.describe()`

```
Out[7]:
```

	0	1	2	3	4	5	6	7	8	9	10	11
count	7501	5747	4389	3345	2529	1864	1369	981	654	395	256	154
unique	115	117	115	114	110	106	102	98	88	80	66	51
top	mineral water	mineral water	mineral water	mineral water	green tea	french fries	green tea	green tea	green tea	green tea	low fat yogurt	green tea
freq	577	484	375	201	153	107	96	67	57	31	22	15

Generating Transaction List

```
In [8]: transactions = []
for i in range(0, 7501):
    transactions.append([str(A.values[i,j]) for j in range(0, 20)])
```

Creating apriori model

```
In [9]: from apyori import apriori
tran_rules = apriori(transactions, min_support = 0.003, min_confidence = 0.6, min_lift = 1, min_rule_len = 2)
```

Visualising the results

```
In [10]: results = list(tran_rules)
print(results[:10])
```

```
[RelationRecord(items=frozenset({'spaghetti', 'ground beef', 'cereals'}), support=
0.0030662578322890282, ordered_statistics=[OrderedStatistic(items_base=frozenset
({'ground beef', 'cereals'}), items_add=frozenset({'spaghetti'}), confidence=0.676
4705882352942, lift=3.8853031258445188)]), RelationRecord(items=frozenset({'spaghe
tti', 'tomatoes', 'olive oil'}), support=0.004399413411545127, ordered_statistics=
[OrderedStatistic(items_base=frozenset({'tomatoes', 'olive oil'}), items_add=froze
nset({'spaghetti'}), confidence=0.6111111111111112, lift=3.5099115194827295)]), Re
lationRecord(items=frozenset({'spaghetti', 'ground beef', 'nan', 'cereals'}), supp
ort=0.0030662578322890282, ordered_statistics=[OrderedStatistic(items_base=frozens
et({'ground beef', 'cereals'}), items_add=frozenset({'spaghetti', 'nan'}), confide
nce=0.6764705882352942, lift=3.8853031258445188), OrderedStatistic(items_base=froz
enset({'ground beef', 'nan', 'cereals'}), items_add=frozenset({'spaghetti'}), conf
idence=0.6764705882352942, lift=3.8853031258445188)]), RelationRecord(items=frozen
set({'milk', 'soup', 'frozen vegetables', 'mineral water'}), support=0.00306625783
22890282, ordered_statistics=[OrderedStatistic(items_base=frozenset({'milk', 'sou
p', 'frozen vegetables'}), items_add=frozenset({'mineral water'}), confidence=0.76
6666666666666666, lift=3.21631245339299), OrderedStatistic(items_base=frozenset({'so
up', 'frozen vegetables', 'mineral water'}), items_add=frozenset({'milk'}), confid
ence=0.6052631578947368, lift=4.670863114576565)]), RelationRecord(items=frozenset
({'spaghetti', 'nan', 'tomatoes', 'olive oil'}), support=0.004399413411545127, ord
ered_statistics=[OrderedStatistic(items_base=frozenset({'tomatoes', 'olive oil'}),
items_add=frozenset({'spaghetti', 'nan'}), confidence=0.6111111111111112, lift=3.5
099115194827295), OrderedStatistic(items_base=frozenset({'nan', 'tomatoes', 'olive
oil'}), items_add=frozenset({'spaghetti'}), confidence=0.6111111111111112, lift=3.
5099115194827295)]), RelationRecord(items=frozenset({'soup', 'frozen vegetables',
'milk', 'nan', 'mineral water'}), support=0.0030662578322890282, ordered_statistic
s=[OrderedStatistic(items_base=frozenset({'milk', 'soup', 'frozen vegetables'}), i
tems_add=frozenset({'nan', 'mineral water'}), confidence=0.7666666666666666, lift=
3.218112292482745), OrderedStatistic(items_base=frozenset({'soup', 'frozen vegetab
les', 'mineral water'}), items_add=frozenset({'milk', 'nan'}), confidence=0.605263
1578947368, lift=4.670863114576565), OrderedStatistic(items_base=frozenset({'mil
k', 'soup', 'frozen vegetables', 'nan'}), items_add=frozenset({'mineral water'}),
confidence=0.7666666666666666, lift=3.21631245339299), OrderedStatistic(items_base
=frozenset({'soup', 'frozen vegetables', 'nan', 'mineral water'}), items_add=froze
nset({'milk'}), confidence=0.6052631578947368, lift=4.670863114576565)])]
```

```
In [11]: 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'])
```

```
In [12]: resultsinDataFrame.nlargest(n = 10, columns = 'Confidence')
```

```
Out[12]:
```

	Left Hand Side	Right Hand Side	Support	Confidence	Lift
3	milk	mineral water	0.003066	0.766667	3.216312
5	milk	nan	0.003066	0.766667	3.218112
0	ground beef	spaghetti	0.003066	0.676471	3.885303
2	ground beef	spaghetti	0.003066	0.676471	3.885303
1	tomatoes	spaghetti	0.004399	0.611111	3.509912
4	tomatoes	spaghetti	0.004399	0.611111	3.509912

