

Data Analytics Laboratory

Task 2

Association Mining: Finding Association Rules using Apriori principle

Introduction

- Association rules is an unsupervised learning method.
- This is a descriptive, not predictive, method often used to discover interesting relationships hidden in a large dataset.
- The identified relationship is usually called as frequent itemset.
- Association rules are mostly used in mining transaction in databases.
- Each of the uncovered rules is in the form $X \rightarrow Y$, meaning that when item X is observed, item Y is also observed.
- Using association rules, patterns can be discovered from the data.
- Association rules are sometimes referred to as market basket analysis.
- An itemset containing k items is called a k-itemset. Use curly braces like {item 1, item 2, . . . item k} to denote a k-itemset.
- Apriori as the main focus of the discussion of association rules.
- Apriori is one of the earliest and the most fundamental algorithms for generating association rules.

Prerequisites

Define support. If 80% of all transactions contain itemset {bread}, then what is the support value of {bread}?

Define Confidence. {mobile phone, mobile cover, scratch card} has a support count of 0.34. {mobile phone, mobile cover} has a support count of 0.67. What is the Confidence for the above example?

Define Lift. Assuming 1,000 transactions, {milk, eggs} appearing in 300 of them, {milk} appearing in 500, {eggs} appearing in 400, Calculate Lift for the relation (milk \rightarrow eggs).

Install the Apriori library

```
In [ ]: pip install apyori
```

```
Collecting apyori
  Downloading https://files.pythonhosted.org/packages/5e/62/5ffde5c473ea4b033490617ec5caa80d59804875ad3c3c57c0976533a21a/apyori-1.1.2.tar.gz
  (https://files.pythonhosted.org/packages/5e/62/5ffde5c473ea4b033490617ec5caa80d59804875ad3c3c57c0976533a21a/apyori-1.1.2.tar.gz)
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py) ... done
  Created wheel for apyori: filename=apyori-1.1.2-cp36-none-any.whl size=5975 sha256=b89afc9a2adefce72db8d4fcf172774b081ba7a97a08ad6cb55caed758f50b6f
  Stored in directory: /root/.cache/pip/wheels/5d/92/bb/474bbadbcb8c0062b9eb168f69982a0443263f8ab1711a8cad0
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

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

```
In [ ]: url='https://raw.githubusercontent.com/ArunkumarGoge/DataAnalyticsL
df = pd.read_csv(url)
```

```
In [ ]: df.head()
```

Out[57]:

	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
In [ ]: df = pd.read_csv(url, header=None)
```

```
In [ ]: df.head()
```

```
Out[61]:
```

	0	1	2	3	4	5	6	7	8	9	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	yc
1	burgers	meatballs	eggs	0	0	0	0	0	0	0	
2	chutney	0	0	0	0	0	0	0	0	0	
3	turkey	avocado	0	0	0	0	0	0	0	0	
4	mineral water	milk	energy bar	whole wheat rice	green tea	0	0	0	0	0	

```
In [ ]: df.fillna(0,inplace=True)
```

```
In [ ]: df.head()
```

```
Out[64]:
```

	0	1	2	3	4	5	6	7	8	9	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	yc
1	burgers	meatballs	eggs	0	0	0	0	0	0	0	
2	chutney	0	0	0	0	0	0	0	0	0	
3	turkey	avocado	0	0	0	0	0	0	0	0	
4	mineral water	milk	energy bar	whole wheat rice	green tea	0	0	0	0	0	

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

```
Out[65]: (7501, 20)
```

```
In [ ]: #for using aprori need to convert data in list format..
# transaction = [['apple','almonds'],['apple'],['banana','apple']].
transactions = []
for i in range(0,len(df)):
    transactions.append([str(df.values[i,j]) for j in range(0,20) i
```

```
In [ ]: transactions[0]
```

```
Out[67]: ['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',
          'spinach',
          'olive oil']
```

```
In [ ]: rules = apriori(transactions,min_support=0.003,min_confidance=0.2,m
```

```
In [ ]: Results = list(rules)
Results
```

```
Out[69]: [RelationRecord(items=frozenset({'cottage cheese', 'brownies'}), s
support=0.0034662045060658577, ordered_statistics=[OrderedStatistic
(items_base=frozenset({'brownies'}), items_add=frozenset({'cottage
cheese'}), confidence=0.10276679841897232, lift=3.225329518580382)
, OrderedStatistic(items_base=frozenset({'cottage cheese'}), items
_add=frozenset({'brownies'}), confidence=0.10878661087866107, lift
=3.2253295185803816)]),
RelationRecord(items=frozenset({'light cream', 'chicken'}), suppo
rt=0.004532728969470737, ordered_statistics=[OrderedStatistic(item
s_base=frozenset({'chicken'}), items_add=frozenset({'light cream'}
), confidence=0.07555555555555556, lift=4.843950617283951), Ordere
dStatistic(items_base=frozenset({'light cream'}), items_add=frozen
set({'chicken'}), confidence=0.29059829059829057, lift=4.843950617
28395)]),
RelationRecord(items=frozenset({'mushroom cream sauce', 'escalope
'}), support=0.005732568990801226, ordered_statistics=[OrderedStat
istic(items_base=frozenset({'escalope'}), items_add=frozenset({'mu
shroom cream sauce'}), confidence=0.0722689075630252, lift=3.79083
26967150496), OrderedStatistic(items_base=frozenset({'mushroom cre
am sauce'}), items_add=frozenset({'escalope'}), confidence=0.20060
```

```
In [ ]: #convert result in a dataframe for further operation...
df_results = pd.DataFrame(Results)
```

```
In [ ]: df_results.head()
```

Out[71]:

	items	support	ordered_statistics
0	(cottage cheese, brownies)	0.003466	[((brownies), (cottage cheese), 0.102766798418...
1	(light cream, chicken)	0.004533	[((chicken), (light cream), 0.0755555555555555...
2	(mushroom cream sauce, escalope)	0.005733	[((escalope), (mushroom cream sauce), 0.072268...
3	(pasta, escalope)	0.005866	[((escalope), (pasta), 0.07394957983193277, 4....
4	(tomato juice, fresh bread)	0.004266	[((fresh bread), (tomato juice), 0.09907120743...

Apply Apriori Algorithm on the below given dataset.

1. Find which one item repeated mostly in all the transactions.
2. How many pairs of items are there in the above dataset satisfying support value > 0.2 .
3. How many pairs of items are there in the above dataset satisfying confidence value > 0.02 .

<https://bit.ly/3qx5rrG>

Results

The program is implemented in python and the output is observed.

Faculty Signature