

Knowledge Discovery & Data Mining Lab5

Name : Kunal Sanjay Patil

PRN : 20190802025

AIM :

To implement the Apriori Algorithm using WEKA and python.

Apriori Algorithm using WEKA :-

Dataset used :-

	A	B	C	D	E	F	G
1	Transaction Id	Biscuits	Rolls	Cake	Coffee	Drink	Dessert
2	1	YES		YES	YES		
3	2			YES		YES	YES
4	3	YES			YES	YES	
5	4	YES	YES				YES
6	5		YES		YES	YES	YES

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter: Choose **None** [Apply] [Stop]

Current relation: data-weka.filters.unsupervised.attribute.Remove-R1
 Relation: data-weka.filters.unsupervised.attribute.Remove-R1
 Instances: 5
 Attributes: 6
 Sum of weights: 5

Attributes: All | None | Invert | Pattern

No. | Name

1	<input checked="" type="checkbox"/> Biscuits
2	<input type="checkbox"/> Rolls
3	<input type="checkbox"/> Cake
4	<input type="checkbox"/> Coffee
5	<input type="checkbox"/> Drink
6	<input type="checkbox"/> Dessert

Remove

Selected attribute: Name: Biscuits
 Missing: 2 (40%)
 Distinct: 1
 Type: Nominal
 Unique: 0 (0%)

No.	Label	Count	Weight
1	YES	3	3

Class: Dessert (Nom) [v] [Visualize All]

3

Status: OK

Log x 0

Weka Explorer

Preprocess Classify Cluster **Associate** Select attributes Visualize

Associator

Choose **Apriori** -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Start Stop

Result list (right-click for ...)

203250 - Apriori

213526 - Apriori

Associator output

```

=== Run information ===

Scheme:      weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:    data-weka.filters.unsupervised.attribute.Remove-R1
Instances:   5
Attributes:  6
             Biscuits
             Rolls
             Cake
             Coffee
             Drink
             Dessert

=== Associator model (full training set) ===

Apriori
=====

Minimum support: 0.3 (1 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 14

Size of set of large itemsets L(3): 8

Size of set of large itemsets L(4): 1

Best rules found:

1. Rolls=YES 2 ==> Dessert=YES 2    <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
2. Biscuits =YES Dessert=YES 1 ==> Rolls=YES 1    <conf:(1)> lift:(2.5) lev:(0.12) [0] conv:(0.6)
3. Biscuits =YES Rolls=YES 1 ==> Dessert=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
4. Cake=YES Coffee=YES 1 ==> Biscuits =YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)

```

Status OK

Log x0

Weka Explorer

Preprocess Classify Cluster **Associate** Select attributes Visualize

Associator

Choose **Apriori** -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Start Stop

Result list (right-click for ...)

203250 - Apriori

213526 - Apriori

Associator output

```

=== Associator model (full training set) ===

Apriori
=====

Minimum support: 0.3 (1 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 14

Size of set of large itemsets L(3): 8

Size of set of large itemsets L(4): 1

Best rules found:

1. Rolls=YES 2 ==> Dessert=YES 2    <conf:(1)> lift:(1.67) lev:(0.16) [0] conv:(0.8)
2. Biscuits =YES Dessert=YES 1 ==> Rolls=YES 1    <conf:(1)> lift:(2.5) lev:(0.12) [0] conv:(0.6)
3. Biscuits =YES Rolls=YES 1 ==> Dessert=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
4. Cake=YES Coffee=YES 1 ==> Biscuits =YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
5. Biscuits =YES Cake=YES 1 ==> Coffee=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
6. Biscuits =YES Drink=YES 1 ==> Coffee=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
7. Rolls=YES Drink=YES 1 ==> Coffee=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
8. Rolls=YES Coffee=YES 1 ==> Drink=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)
9. Coffee=YES Dessert=YES 1 ==> Rolls=YES 1    <conf:(1)> lift:(2.5) lev:(0.12) [0] conv:(0.6)
10. Rolls=YES Coffee=YES 1 ==> Dessert=YES 1    <conf:(1)> lift:(1.67) lev:(0.08) [0] conv:(0.4)

```

Status OK

Log x0

Apriori Algorithm using python :-

In [1]:

```
pip install apyori
```

Requirement already satisfied: apyori in c:\users\kunal\anaconda3\lib\site-packages (1.1.2)

Note: you may need to restart the kernel to use updated packages.

In [2]:

```
import pandas as pd
from apyori import apriori
```

In [3]:

```
data = pd.read_csv('Market_Basket_Optimisation.csv', header=None)
```

In [4]:

```
data.head()
```

Out[4]:

	0	1	2	3	4	5	6	7	8	9	10
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat 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]:

```
data.shape
```

Out[5]:

```
(7501, 20)
```

In [6]:

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

In [7]:

```
association_rules = apriori(transactions, min_support=0.005, min_confidence=0.2,min_lift =
association_results = list(association_rules)
```

In [8]:

```
print(len(association_results))
```

38

In [9]:

```
association_results
```

Out[9]:

```
[RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), 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({'herb & pepper', 'ground beef'}), 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({'tomato sauce', 'ground beef'}), 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', 'whole wheat pasta'}), support=0.007998933475536596, ordered_statistics=[OrderedStatistic(items_base=frozenset({'olive oil'}), items_add=frozenset({'whole wheat pasta'}), confidence=0.2714932100000001, lift=1.340659481324083)])]
```

In [10]:

```
results = pd.DataFrame(association_results)
results.head()
```

Out[10]:

	items	support	ordered_statistics
0	(escalope, mushroom cream sauce)	0.005733	[((mushroom cream sauce), (escalope), 0.300699...]
1	(pasta, escalope)	0.005866	[((pasta), (escalope), 0.3728813559322034, 4.7...
2	(herb & pepper, ground beef)	0.015998	[((herb & pepper), (ground beef), 0.3234501347...
3	(tomato sauce, ground beef)	0.005333	[((tomato sauce), (ground beef), 0.37735849056...
4	(olive oil, whole wheat pasta)	0.007999	[((whole wheat pasta), (olive oil), 0.27149321...