

# Practicum Problems

## 2.1 Problem 1

Load the market-basket sample dataset into the Orange application, and run both frequent item set as well as association rule modules. Set the support threshold to 10% and observe the antecedent in the rules with the highest lift. What item is observed to be there, and what is its support? Is this a valuable association rule? Why or why not?

### Answer:

When the support threshold is set to 10% we obtain Eggs and Eggs, Diaper as antecedents with highest lift=2.5 and support=20%.

Yes, the association rule is valuable because lift highlights rules which are rare but informative. Also, high-confidence rules can sometimes be misleading because the confidence measure ignores the support of the itemset appearing in the rule consequent. One way to address this problem is by applying a metric known as lift.

\*\*\* Association Rules

Info	Supp	Conf	Covr	Strg	Lift	Levr	Antecedent	Consequent
Number of rules: 38	0.20	1.00	0.20	2.00	2.50	0.12	Eggs=1 →	Bread=1, Beer=1
Filtered rules: 38	0.20	1.00	0.20	2.00	2.50	0.12	Diapers=1, Eggs=1 →	Bread=1, Beer=1
Selected rules: 4	0.20	1.00	0.20	2.00	2.50	0.12	Eggs=1 →	Bread=1, Diapers=1, Beer=1
Selected examples: 1	0.20	1.00	0.20	3.00	1.67	0.08	Eggs=1 →	Bread=1, Diapers=1
Find association rules	0.20	1.00	0.20	3.00	1.67	0.08	Eggs=1 →	Beer=1
Minimal support: 10%	0.20	1.00	0.20	3.00	1.67	0.08	Bread=1, Eggs=1 →	Beer=1
Minimal confidence: 100%	0.20	1.00	0.20	3.00	1.67	0.08	Diapers=1, Eggs=1 →	Beer=1
Max. number of rules: 10000	0.20	1.00	0.20	3.00	1.67	0.08	Eggs=1 →	Diapers=1, Beer=1
<input type="checkbox"/> Induce classification (itemset → class) rules	0.20	1.00	0.20	3.00	1.67	0.08	Beer=1, Eggs=1 →	Bread=1, Diapers=1
<input type="checkbox"/> Find rules	0.20	1.00	0.20	3.00	1.67	0.08	Bread=1, Diapers=1, Eggs=1 →	Beer=1
	0.20	1.00	0.20	3.00	1.67	0.08	Bread=1, Eggs=1 →	Diapers=1, Beer=1
	0.20	1.00	0.20	3.00	1.67	0.08	Bread=1, Cola=1 →	Milk=1, Diapers=1
	0.20	1.00	0.20	3.00	1.67	0.08	Beer=1, Cola=1 →	Milk=1, Diapers=1
	0.40	1.00	0.40	1.50	1.67	0.16	Cola=1 →	Milk=1, Diapers=1
	0.20	1.00	0.20	4.00	1.25	0.04	Bread=1 Milk=1 Beer=1 →	Diapers=1

## 2.2 Problem 2

Load the Extended Bakery dataset (75000-out2-final.csv) into the Orange application, and run both frequent itemset as well as association rule modules. Set the support threshold to 1% and the confidence threshold to 90%. Observe the association rules containing the Cherry Tart item within the antecedent. What other item appears with it? When the confidence threshold is lowered to 45%, does the Cherry Tart item now appear without another item in the antecedent? Is the same consequent observed in both cases? How did lowering the confidence threshold lead to this change? Hint: Reference the Simpson's Paradox section of the text.

**Answer:**

### Case 1:

When the support threshold is set to 1% and the confidence threshold is set to 90%.

Opera Cake appears along with Cherry Tart as an antecedent.

Apricot Danish is observed as consequent.

\*\*\* Association Rules

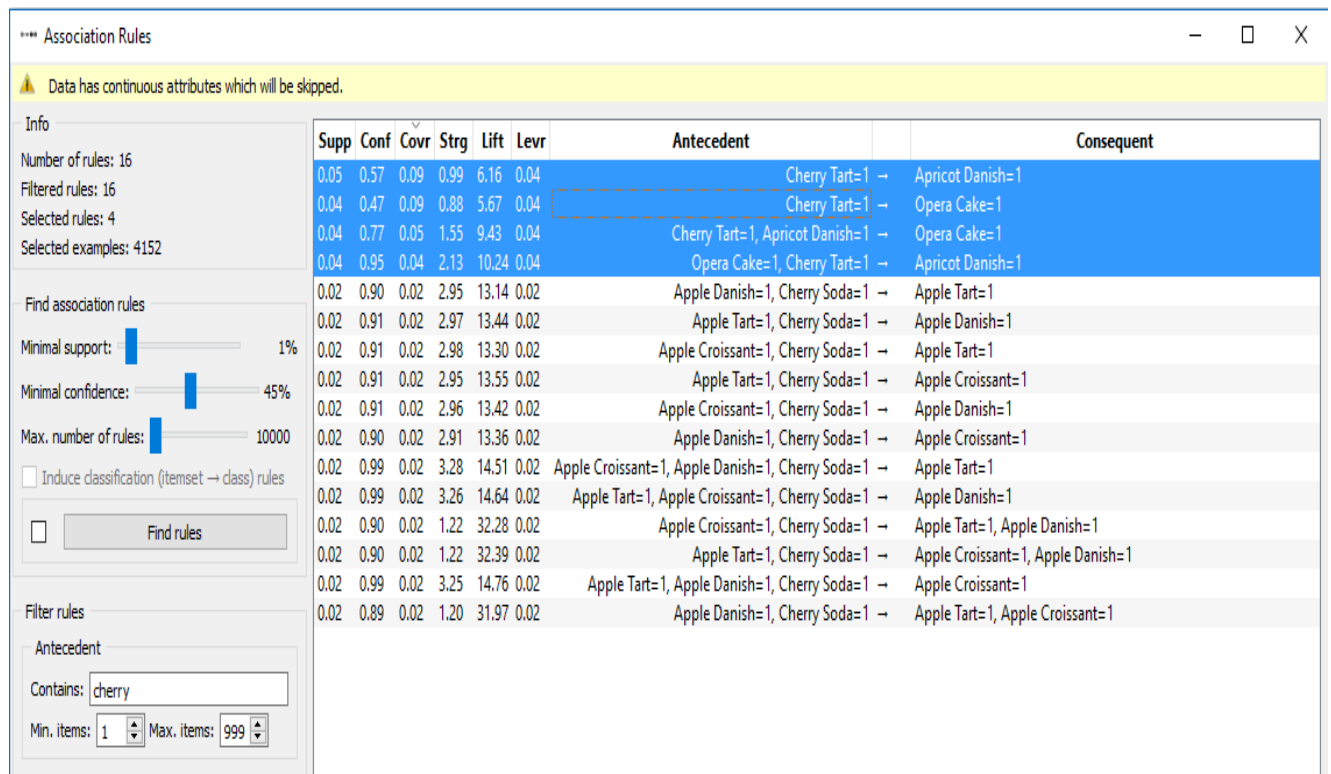
⚠ Data has continuous attributes which will be skipped.

Info	Supp	Conf	Covr	Strg	Lift	Levr	Antecedent	Consequent
Number of rules: 9	0.04	0.95	0.04	2.13	10.24	0.04	Opera Cake=1, Cherry Tart=1 →	Apricot Danish=1
Filtered rules: 9	0.02	0.91	0.02	2.97	13.44	0.02	Apple Tart=1, Cherry Soda=1 →	Apple Danish=1
Selected rules: 1	0.02	0.91	0.02	2.98	13.30	0.02	Apple Croissant=1, Cherry Soda=1 →	Apple Tart=1
Selected examples: 3083	0.02	0.91	0.02	2.95	13.55	0.02	Apple Tart=1, Cherry Soda=1 →	Apple Croissant=1
Find association rules	0.02	0.91	0.02	2.96	13.42	0.02	Apple Croissant=1, Cherry Soda=1 →	Apple Danish=1
Minimal support: 1%	0.02	0.99	0.02	3.28	14.51	0.02	Apple Croissant=1, Apple Danish=1, Cherry Soda=1 →	Apple Tart=1
Minimal confidence: 90%	0.02	0.99	0.02	3.26	14.64	0.02	Apple Tart=1, Apple Croissant=1, Cherry Soda=1 →	Apple Danish=1
Max. number of rules: 10000	0.02	0.90	0.02	1.22	32.39	0.02	Apple Tart=1, Cherry Soda=1 →	Apple Croissant=1, Apple Danish=1
<input type="checkbox"/> Induce classification (itemset → class) rules	0.02	0.99	0.02	3.25	14.76	0.02	Apple Tart=1, Apple Danish=1, Cherry Soda=1 →	Apple Croissant=1
<input type="checkbox"/> Find rules								

### Case 2:

When the confidence threshold is lowered to 45%

Cherry Tart appears with Opera Cake and Apricot Danish as an antecedent and twice without any other antecedents. The consequents have changed to Opera Cake and Apricot Danish.



When Opera Cake and Cherry Tart are considered as antecedents with Apricot Danish as consequent then the confidence is 95%. However, when Opera Cake to Apricot Danish is considered the confidence is 52% and for Cherry Tart to Apricot Danish we get 57% confidence. We find that the Cherry Tart, Opera Cake and Apricot Danish is positively correlated in the combined data but is negatively correlated in the stratified data.

Here the presence of hidden variables may have caused the observed relationship between Cherry Tart, Opera Cake and Apricot Danish to reverse its direction, due to the phenomenon known as Simpson's paradox.

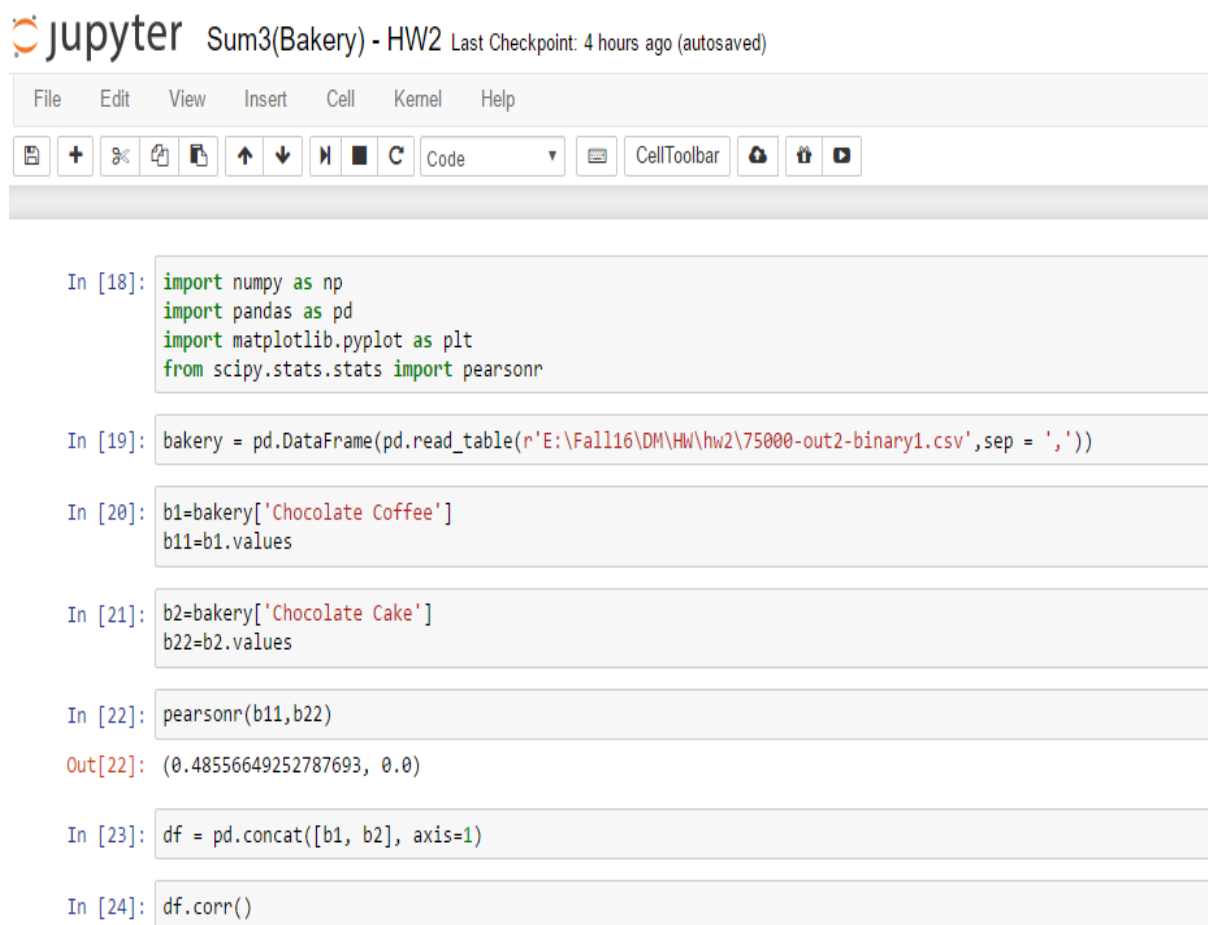
Hence, proper stratification is needed to avoid generating spurious patterns resulting from Simpson's paradox.

Supp	Conf	Covr	Strg	Lift	Levr	Antecedent	Consequent
0.04	0.53	0.08	1.13	5.67	0.04	Opera Cake=1	Cherry Tart=1
0.04	0.52	0.08	1.13	5.66	0.04	Opera Cake=1	Apricot Danish=1
0.04	0.50	0.08	0.65	9.43	0.04	Opera Cake=1	Cherry Tart=1, Apricot Danish=1
0.04	0.96	0.04	2.17	10.26	0.04	Opera Cake=1, Apricot Danish=1	Cherry Tart=1
0.04	0.95	0.04	2.13	10.24	0.04	Opera Cake=1, Cherry Tart=1	Apricot Danish=1

## 2.3 Problem 3

Load the Extended Bakery dataset (75000-out2-binary.csv) into Python using a Pandas dataframe. Calculate the binary correlation coefficient for the Chocolate Coffee and Chocolate Cake items. Show whether the two items are symmetric binary variables via their co-presence and co-absence. Would an association rule between these items as antecedent and consequent have a high confidence level? Why or why not?

**Answer:**



```
Sum3(Bakery) - HW2 Last Checkpoint: 4 hours ago (autosaved)
File Edit View Insert Cell Kernel Help
[Icons] [Code] CellToolbar [Icons]

In [18]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats.stats import pearsonr

In [19]: bakery = pd.DataFrame(pd.read_table(r'E:\Fall16\DM\HW\hw2\75000-out2-binary1.csv', sep = ','))

In [20]: b1=bakery['Chocolate Coffee']
b11=b1.values

In [21]: b2=bakery['Chocolate Cake']
b22=b2.values

In [22]: pearsonr(b11,b22)
Out[22]: (0.48556649252787693, 0.0)

In [23]: df = pd.concat([b1, b2], axis=1)

In [24]: df.corr()
```

File Edit View Insert Cell Kernel Help

Save Add Reload Close Copy Paste Undo Redo Code CellToolbar Help

```
In [24]: df.corr()
```

```
Out[24]:
```

	Chocolate Coffee	Chocolate Cake
Chocolate Coffee	1.000000	0.485566
Chocolate Cake	0.485566	1.000000

```
In [26]: b2=bakery['Chocolate Cake']
```

```
In [27]: newdf =(b1==0) & (b2==0)
```

```
In [28]: newdf
```

```
Out[28]:
```

0	True
1	True
2	True
3	True
4	True
5	True
6	True
7	True
8	True
9	True
10	True
11	True
12	True
13	True
14	True

File Edit View Insert Cell Kernel Help

Save Add Reload Close Copy Paste Undo Redo Code CellToolbar Help

```
74993 False
74994 True
74995 True
74996 True
74997 True
74998 True
74999 True
dtype: bool
```

```
In [29]: gb = newdf.groupby(newdf)
Coab = gb.get_group(True)
CoAbsence= Coab.sum()
```

```
In [30]: CoAbsence
```

```
Out[30]: 65802
```

```
In [31]: newdf1 =(b1==1) & (b2==1)
```

```
In [32]: newdf1
```

```
Out[32]:
```

0	False
1	False
2	False
3	False
4	False
5	False
6	False
7	False
8	False
9	False
10	False
11	False
12	False
13	False
14	False

```
File Edit View Insert Cell Kernel Help
[Icons] [Code] CellToolbar [Icons]

74988 False
74989 False
74990 False
74991 False
74992 False
74993 False
74994 False
74995 False
74996 False
74997 False
74998 False
74999 False
dtype: bool

In [33]: gb = newdf1.groupby(newdf1)
        Cop = gb.get_group(True)
        CoPresence= Cop.sum()

In [34]: CoPresence

Out[34]: 3303
```

After swapping columns Chocolate Cake and Chocolate Coffee -

```
In [21]: newdf =(b2==0) & (b1==0)
        gb = newdf.groupby(newdf)
        Coab = gb.get_group(True)
        CoAbsence= Coab.sum()
        CoAbsence

Out[21]: 65802

In [22]: newdf1 =(b2==1) & (b1==1)
        gb = newdf1.groupby(newdf1)
        Cop = gb.get_group(True)
        CoPresence= Cop.sum()
        CoPresence

Out[22]: 3303
```

Yes, they are symmetric variables because even if the columns are swapped the correlation value and co-presence and co-absence values do not change.

No, the association rule between Chocolate Coffee and Chocolate Cake won't have high confidence because of high value of co-absence.