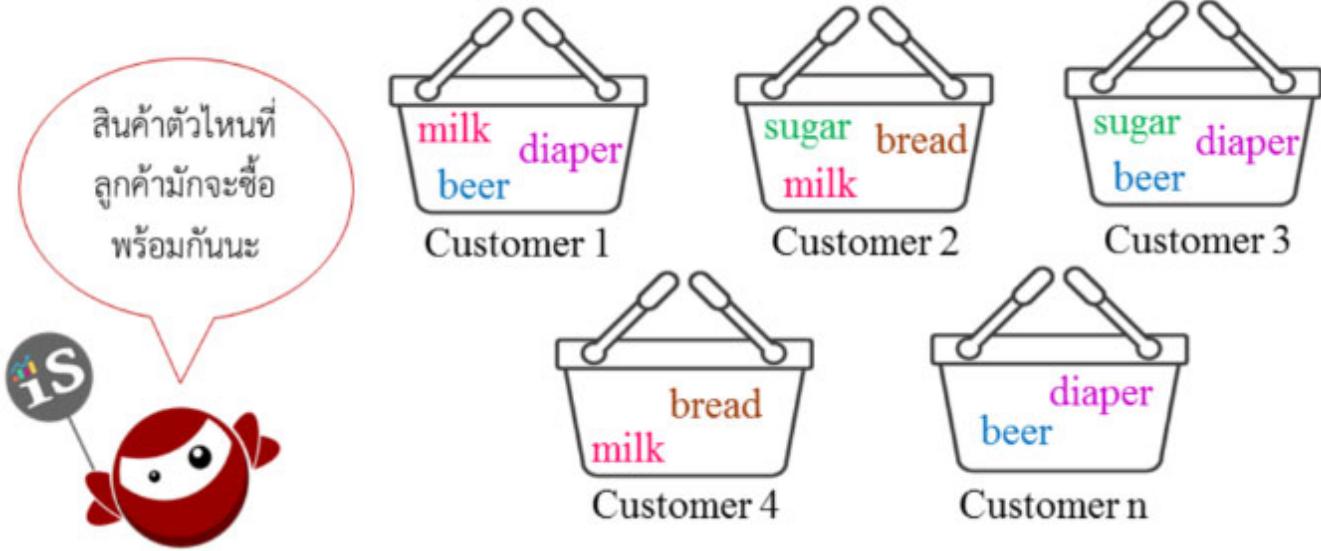


Basket Analysis



- by  MLxtend

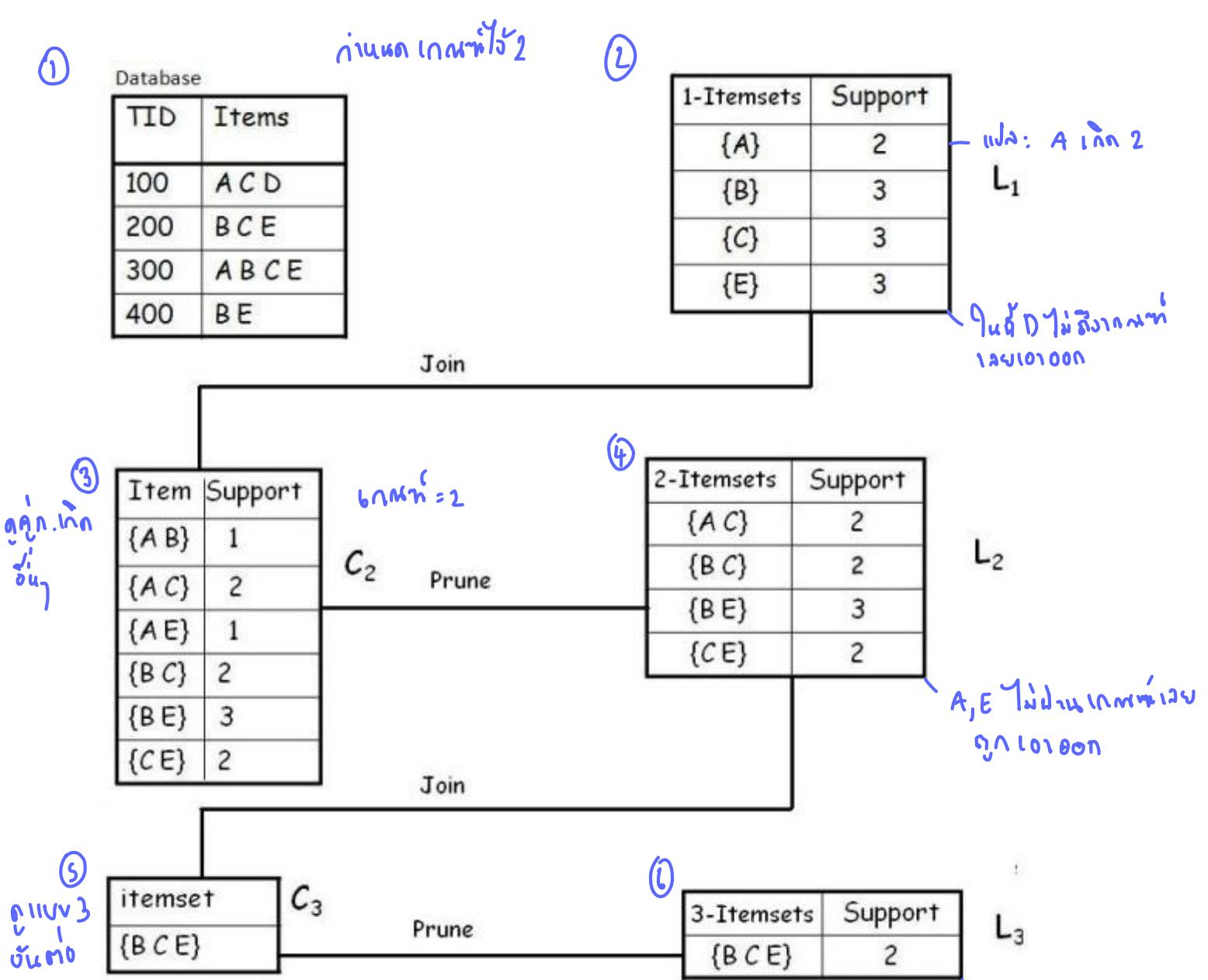
unasso, rvl e ajo w j

Apriori Algorithm

อัลกอริทึม Apriori เป็นเทคนิคที่ถูกนำมาใช้ในการขุดหา association rule ในฐานข้อมูลที่มีการทำธุกรรม มีประสิทธิภาพในการระบุชุดรายการที่เป็นที่นิยมและสร้าง association rule อัลกอริทึมนี้ใช้วิธีการ "bottom-up" และมีพื้นฐานบนหลัก Apriori

Key Steps: "ກ່ຽວຂ້ອງ ນາງນັນຢູ່ ກົມນາ"

- Generate Frequent Itemsets:** อัลกอริทึมเริ่มด้วยการค้นหา frequent itemsets ซึ่งเป็นชุดของสินค้าที่เกิดขึ้นพร้อมกันในการทำธุรกรรมมากกว่า **minimum support threshold** = $\frac{\text{จำนวนที่ซื้อของชุด}}{\text{จำนวนรายการทั้งหมด}} \times 100\%$ (e.g. transac 100 จำนวนที่ซื้อของชุด 5 รายการใน transaction 50% ของจำนวน transaction) ที่กำหนดไว้
 - Join Step:** ในขั้นตอนนี้อัลกอริทึมสร้าง candidate itemsets โดยการรวม frequent itemsets จากการทำซ้ำก่อนหน้า \rightarrow ถ้า item ที่มี support มากกว่า "minimum support threshold" ที่กำหนดไว้ ให้รวมเข้าด้วยกัน
 - Prune Step:** candidate itemsets ที่ไม่ตรงตามเกณฑ์ minimum support threshold จะถูกตัดทิ้งเพื่อลดพื้นที่ในการค้นหา ตัวอย่างใน transaction ที่มี item 5 รายการ ถ้า item ที่มี support ต่ำกว่า 15% ถูกตัดออก
 - Repeat the Join and Prune Steps:** อัลกอริทึมทำซ้ำในกระบวนการรวมและตัดจนกว่าจะไม่สามารถสร้าง frequent itemsets ใหม่ได้ แล้วถ้า item ที่มี support ต่ำกว่า 15% ถูกตัดออก
 - Generate Association Rules:** เมื่อได้รับ frequent itemsets แล้ว association rules จะถูกสร้างขึ้นจากชุดรายการเหล่านี้ กฎประกอบด้วย antecedent (ด้านซ้าย) และส่วน consequent (ด้านขวา)
 - Calculate Metrics:** ตัวชี้วัด เช่น support, confidence, และ lift ถูกคำนวณสำหรับแต่ละ association rule เพื่อวัดความแข็งแกร่งและความสำคัญของความสัมพันธ์



▼ Extract frequent itemsets

เริ่มจากการหา frequent itemsets

- ใช้ dataset จากไฟล์ Groceries_dataset.csv
 - อัปโหลดขึ้น sample_data

▼ สร้าง DataFrame

- เรียกใช้ pandas

```
import pandas as pd
```

- อ่านไฟล์ `Groceries_dataset.csv` จาก `sample_data` โดยคลิกขวาที่ชื่อไฟล์ และเลือก `Copy path`

```
data = pd.read_csv('/content/sample_data/Groceries_dataset.csv')
data.head()
```

	Member_number	Date	itemDescription
0	1808	21-07-2015	tropical fruit
1	2552	05-01-2015	whole milk
2	2300	19-09-2015	pip fruit
3	1187	12-12-2015	other vegetables
4	3037	01-02-2015	whole milk

```
data.shape
```

row col
(38765, 3)

```
data.sort_values(['Member_number', 'Date'])
```

	Member_number	Date	itemDescription
4843	1000	15-03-2015	sausage
8395	1000	15-03-2015	whole milk
20992	1000	15-03-2015	semi-finished bread
24544	1000	15-03-2015	yogurt
13331	1000	24-06-2014	whole milk
...
3578	5000	10-02-2015	soda
19727	5000	10-02-2015	root vegetables
34885	5000	10-02-2015	semi-finished bread
9340	5000	16-11-2014	bottled beer
25489	5000	16-11-2014	other vegetables

38765 rows × 3 columns

```
len(data.groupby(['Member_number', 'Date']))
```

14963

- ดูรายการสินค้าและความถี่ *

```
data['itemDescription'].value_counts()
```

whole milk	2502	win
other vegetables	1898	
rolls/buns	1716	
soda	1514	
yogurt	1334	
	...	
rubbing alcohol	5	
bags	4	
baby cosmetics	3	
kitchen utensil	1	
preservation products	1	now

— ວິຊາການ ເສັ້າກັ່ນແມ່ນ 16 ຮາຊາກວ

- ทำให้อยู่ในรูปของ list of transactions

Group 0122 ရန်မြန်း၊ ရန်

```
list(data.groupby(['Member_number', 'Date']))[:3]
```

0.750.500
ԱՆՎԻՇԱԽԵ-
ՅԱ

```

→ [((1000, '15-03-2015'),
      Member_number           Date      itemDescription
  4843           1000  15-03-2015           sausage
  8395           1000  15-03-2015       whole milk
 20992           1000  15-03-2015 semi-finished bread
 24544           1000  15-03-2015        yogurt),
→ ((1000, '24-06-2014'),
      Member_number           Date      itemDescription
 13331           1000  24-06-2014       whole milk
 29480           1000  24-06-2014        pastry
 32851           1000  24-06-2014  salty snack),
→ ((1000, '24-07-2015'),
      Member_number           Date      itemDescription
 2047           1000  24-07-2015    canned beer
 18196           1000  24-07-2015  misc. beverages)]

```

```
list(data.groupby(['Member_number', 'Date']))[0]
```

```
((1000, '15-03-2015'),
    Member_number           Date      itemDescription
4843             1000  15-03-2015          sausage
8395             1000  15-03-2015      whole milk
20992            1000  15-03-2015
                           semi-finished bread
24544            1000  15-03-2015          yogurt)
```

transaction: key, value
value: value or set (dataframe)

```
list(data.groupby(['Member_number', 'Date']))[0][1]
```

0:00 នៅម៉ោង

data frame

var train
 \nou{t}isn

Member_number	Date	itemDescription
4843	1000	15-03-2015
8395	1000	15-03-2015
20992	1000	15-03-2015
24544	1000	15-03-2015

```
list(data.groupby(['Member_number', 'Date']))[0][1]['itemDescription']
```

ກົດໃຫຍ້
trnsac.

4843	sausage
8395	whole milk
20992	semi-finished bread
24544	yogurt

Name: itemDescription, dtype: object

ດີດ້ວຍໃຫຍ້
col.

```
list(data.groupby(
```

['Member_number', 'Date']))[0][1]['itemDescription'].tolist() ໄຟລ່າງທີ່ຈະໃຫຍ້

ດີດ້ວຍ.. ['sausage', 'whole milk', 'semi-finished bread', 'yogurt']

```
transactions = [
    a[1]['itemDescription'].tolist()
    for a in list(data.groupby(['Member_number', 'Date']))]
transactions[:10] ແມ່ນ trnsac.
```

ດີດ້ວຍສັບ
ໂຈງໄລ່ຈະ trnsac.

```
[['sausage', 'whole milk', 'semi-finished bread', 'yogurt'],
 ['whole milk', 'pastry', 'salty snack'],
 ['canned beer', 'misc. beverages'],
 ['sausage', 'hygiene articles'],
 ['soda', 'pickled vegetables'],
 ['frankfurter', 'curd'],
 ['sausage', 'whole milk', 'rolls/buns'],
 ['whole milk', 'soda'],
 ['beef', 'white bread'],
 ['frankfurter', 'soda', 'whipped/sour cream']]
```

▼ ປັບຮູບແບບ transactions

transactions ຈະຕ້ອງອີ່ມໃນຮູບແບບຂອງຕາງໆ ທີ່ແຕ່ລະແກ່ແສດງການມີ (True) ອີ່ມີ (False) ສິນຄ້າໃນ 1
ຮາຍການຂອງ transaction

ຮູບແບບ
matrix

- ໃໝ່ TransactionEncoder

∴ ອົກການຂໍມູນຕາງໆ 15,000 ໄດ້

ໄລຍະ 175 col.

```
from mlxtend.preprocessing import TransactionEncoder
```

ແກ່ລະແກ່ວ່າ: ບອກວ່າດ້ວຍສິນຄ້າຕັ້ງໄຟນະ
ກົດເປັນ true ດ້ວຍບໍ່ກົດ false

```
te = TransactionEncoder()
```

```
→ te.fit(transactions)
```

```
t_encode = te.transform(transactions)
```

```
t_encode
```

```
array([[False, False, False, ..., True, True, False],
       [False, False, False, ..., True, False, False],
       [False, False, False, ..., False, False, False],
       ...,
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False]])
```

ດີ

colon. fit

```
df = pd.DataFrame(t_encode, columns=te.columns_)
```

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics
0		False	False	False	False
1		False	False	False	False
2		False	False	False	False
3		False	False	False	False
4		False	False	False	False
...
14958		False	False	False	False
14959		False	False	False	False
14960		False	False	False	False
14961		False	False	False	False
14962		False	False	False	False

▼ Apriori

- ہے,

หา frequent itemsets ด้วย Apriori

- เรียกใช้ apriori library

```
from mlxtend.frequent_patterns import apriori
```

```
import warnings
warnings.filterwarnings('ignore', category=DeprecationWarning)
```

- ໜ frequent itemsets

15000*.001

15.0

100

17. min
(minimum support)

```
fia = apriori(df, min_support=0.001, use_colnames=True)
```

9. សំណើលេខ 0-1
និងអនុគមនាគារ នៃសំណើលេខ 0-1
e.g. $0.067 \rightarrow 1000 / 16000$
(សំណើលេខ 16000)

	support	itemsets
0	0.004010	(Instant food products)
1	<u>0.021386</u> ^{2% มาก} ₁₅₀₀₀	(UHT-milk)
2	0.001470	(abrasive cleaner)
3	0.001938	(artif. sweetener)
4	0.008087	(baking powder)
...
745	0.001136	(rolls/buns, sausage, whole milk)
746	0.001002	(rolls/buns, soda, whole milk)
747	0.001337	(rolls/buns, yogurt, whole milk)
748	0.001069	(whole milk, sausage, soda)
749	0.001470	(whole milk, sausage, yogurt)

- ผลลัพธ์ที่ได้จะเป็น DataFrame ^{750 รายการหนึ่งรายการ}
- สามารถปรับแต่งได้ตามต้องการ เช่น เพิ่ม Series ที่แสดงจำนวนรายการใน itemset

↙ e.g.

```
fia['length'] = fia['itemsets'].apply(lambda x: len(x))
fia
```

	support	itemsets	length
0	0.004010	(Instant food products)	1
1	0.021386	(UHT-milk)	1
2	0.001470	(abrasive cleaner)	1
3	0.001938	(artif. sweetener)	1
4	0.008087	(baking powder)	1
...
745	0.001136	(rolls/buns, sausage, whole milk)	3
746	0.001002	(rolls/buns, soda, whole milk)	3
747	0.001337	(rolls/buns, yogurt, whole milk)	3
748	0.001069	(whole milk, sausage, soda)	3
749	0.001470	(whole milk, sausage, yogurt)	3

750 rows × 3 columns

- สามารถค้นหา itemsets ที่ต้องการได้

```
fia[fia['length']==2]
```

	support	itemsets	length
149	0.001069	(bottled water, UHT-milk)	2
150	0.002139	(other vegetables, UHT-milk)	2
151	0.001804	(rolls/buns, UHT-milk)	2
152	0.001002	(root vegetables, UHT-milk)	2
153	0.001136	(sausage, UHT-milk)	2
...
736	0.002941	(yogurt, whipped/sour cream)	2
737	0.003141	(whole milk, white bread)	2
738	0.001069	(yogurt, white bread)	2
739	0.001270	(whole milk, white wine)	2
740	0.011161	(whole milk, yogurt)	2

592 rows × 3 columns

1% ≈ 150 รายการ.

```
fia[(fia['length']==2) & (fia['support']>=0.01)]
```

	support	itemsets	length
609	0.010559	(rolls/buns, other vegetables)	2
625	0.014837	(whole milk, other vegetables)	2
677	0.013968	(rolls/buns, whole milk)	2
717	0.011629	(soda, whole milk)	2
740	0.011161	(whole milk, yogurt)	2

```
fia[fia['itemsets'] == {'yogurt', 'whole milk'}]
```

=

	support	itemsets	length
740	0.011161	(whole milk, yogurt)	2

จัด itemset ใหม่-เริ่มต้นที่ itemset ที่มี item ก่อนหน้า



FP-Growth Algorithm

= Algor. นี้

นำไปใช้กับ Apriori

อัลกอริทึม FP-Growth (Frequent Pattern-Growth) เป็นทางเลือกในการคัดหา frequent itemsets แทนอัลกอริทึม Apriori ในชุดข้อมูลขนาดใหญ่

Key Steps:

1. Build FP-Tree: อัลกอริทึมสร้าง FP-Tree → เอามาใช้ ทั้งในเก็บข้อมูลใน memory (มีข้อบกพร่องที่: 1, 2, 3 itemset บล็อก bottom-up ที่ไม่สามารถใช้กับ Apriori)

2. Construct Conditional Pattern Bases: สร้าง conditional pattern base จาก FP-Tree

3. Mine Frequent Itemsets: หา Frequent itemsets จากการสำรวจ Conditional Pattern Bases โดยไม่ต้องสร้างและติดรายการตัวเลือก

4. Generate Association Rules: เช่นเดียวกับอัลกอริทึม Apriori กว่าความสัมพันธ์จะถูกสร้างขึ้นจาก frequent itemsets

FP-Growth

หา frequent itemsets ด้วย FP-Growth

- เรียกใช้ fpgrowth library

```
from mlxtend.frequent_patterns import fpgrowth
```

- หา frequent itemsets

```
fif = fpgrowth(df, min_support=0.001, use_colnames=True)  
fif
```

	support	itemsets
0	0.157923	(whole milk)
1	0.085879	(yogurt)
2	0.060349	(sausage)
3	0.009490	(semi-finished bread)
4	0.051728	(pastry)
...
745	0.001403	(chewing gum, yogurt)
746	0.001069	(chewing gum, other vegetables)
747	0.001002	(chewing gum, soda)
748	0.001069	(whole milk, pasta)
749	0.001002	(rolls/buns, seasonal products)

750 rows × 2 columns

▷ ผู้อ.นักศึกษาท่านใดที่ต้องการเรียนรู้ FP-growth และ Apriori ก็สามารถหาน้ำหน้า

```
fif[fif['itemsets'] == {'sausage', 'UHT-milk'}]
```

	support	itemsets
565	0.001136	(sausage, UHT-milk)

```
fia[fia['length']==2]
```

	support	itemsets	length
149	0.001069	(bottled water, UHT-milk)	2
150	0.002139	(other vegetables, UHT-milk)	2
151	0.001804	(rolls/buns, UHT-milk)	2
152	0.001002	(root vegetables, UHT-milk)	2
153	0.001136	(sausage, UHT-milk)	2
...
736	0.002941	(yogurt, whipped/sour cream)	2
737	0.003141	(whole milk, white bread)	2
738	0.001069	(yogurt, white bread)	2
739	0.001270	(whole milk, white wine)	2
740	0.011161	(whole milk, yogurt)	2

592 rows × 3 columns

ດាស់ងុំ ឲ្យចិត្តុបន្តិ៍
ດាស់ងុំបន្តិ៍ ឲ្យចិត្តុុង

▼ Association Rules

Guassso, rule ၅၁ ကုသံလုပ်မှူး ၁။မျှော်ဘား၏သိသုတေသနများ

Association rules เป็นการแสดงความสัมพันธ์ทางตรรกะระหว่างสินค้าโดยขึ้นอยู่กับการเกิดขึ้นร่วมกันในการทำธุรกรรม กฎเหล่านี้ช่วยให้มีความเข้าใจในพฤติกรรมและความชอบของลูกค้า เพื่อช่วยธุรกิจในการตัดสินใจอย่างมีข้อมูล

Format of Association Rule:

三

- Antecedent (Left-hand side) -> Consequent (Right-hand side)
ก็... \rightarrow ॥ กว่า..

e.g. ດ້ວຍເນື່ອໃຈ່າຍ

▼ Metrics 9ນາງ.ກົດມະນີເນື້ອງການຮຽນຮູ້ນີ້ ຊະບັບ 5 ຂີ້ (ມີວິທີກົດມີກີ້)

1. Support: (នៅទៅនូវការណ៍)

Support is the proportion of transactions that contain both the antecedent and the consequent of an association rule.

Support(A->B) = (Number of Transactions containing A and B) / (Total Number of Transactions)

వ → శంక్రాన్తి A, B 40 టాబ్లూస్

Example: Suppose out of 100 transactions, 40 contain both items A and B.

$$\text{Support}(A \rightarrow B) = 40 / 100 = 0.4$$

2. Confidence:

Confidence is the probability of finding the consequent in a transaction given that the antecedent is already present in that transaction. *ពិនិត្យការងារថា ពីការបានដាក់មិនមែន*

$\text{Confidence}(A \rightarrow B) = (\text{Number of Transactions containing } A \text{ and } B) / (\text{Number of Transactions containing } A)$

នៅលើ 100 នៅទាំង

មាន A

នៅក្នុង 40 នៅលើ 60 នៅទាំង

(ការងារ A នៅលើ 60 នៅលើ 100 នៅទាំង B)

Example: Out of the 60 transactions containing item A, 40 also contain item B.

នៅមុននេះ នៅលើ 60 នៅទាំង

ដើម្បី

$\text{Confidence}(A \rightarrow B) = 40 / 60 = \underline{0.6667}$

នៅមុននេះ នៅលើ 60 នៅទាំង

3. Lift:

Lift measures the ratio of observed support to expected support for an association rule. It indicates how much more likely the items in the antecedent and consequent are to be bought together compared to if they were bought independently.

នៅក្នុង នៅក្នុង Antecedent
នៅក្នុង Consequent

នូវ support

$\text{Lift}(A \rightarrow B) = \frac{\text{Support}(A \rightarrow B)}{\text{Support}(A) * \text{Support}(B)}$

Example:

Suppose we have the following information:

- Total number of transactions: 1000
- Transactions containing item A: 600
- Transactions containing item B: 400
- Transactions containing both A and B: 300

Calculate Lift for Rule A->B:

1. Calculate Support:

- $\text{Support}(A) = \text{Transactions containing } A / \text{Total transactions} = 600 / 1000 = 0.6$
- $\text{Support}(B) = \text{Transactions containing } B / \text{Total transactions} = 400 / 1000 = 0.4$
- $\text{Support}(A \rightarrow B) = \text{Transactions containing } A \text{ and } B / \text{Total transactions} = 300 / 1000 = 0.3$

2. Calculate Lift:

- $\text{Lift}(A \rightarrow B) = \text{Support}(A \rightarrow B) / (\text{Support}(A) * \text{Support}(B))$
- $\text{Lift}(A \rightarrow B) = 0.3 / (0.6 * 0.4) = \underline{1.25}$

→ Interpretation:

- ការងារ $\text{Lift}(A \rightarrow B) = 1$ ផ្តល់ព័ត៌មានថា ការកែត្រាវិធី A និង B មិនកំពុងក្រោមគ្នា
- ការងារ $\text{Lift}(A \rightarrow B) > 1$ ផ្តល់ព័ត៌មានថា ការកែត្រាវិធី A ធ្វើការជូន B
- ការងារ $\text{Lift}(A \rightarrow B) < 1$ ផ្តល់ព័ត៌មានថា ការកែត្រាវិធី A ធ្វើការជូន B

(A ការងារក្នុង B)

ในตัวอย่างนี้ Lift(A->B) มากกว่า 1 ชี้งแสดงถึงการเขื่อมโยงบวกระหว่างรายการ A และ B นี้ เป็นการบ่งชี้ว่า ลูกค้าที่ซื้อรายการ A มีโอกาสที่จะซื้อรายการ B ด้วย และการมี A เพิ่มโอกาสในการซื้อ B

4. Leverage:

Leverage measures the difference between the actual occurrence of both the antecedent and the consequent in transactions and the expected occurrence if they were independent.

$$\text{Leverage}(A \rightarrow B) = \text{Support}(A \rightarrow B) - (\text{Support}(A) * \text{Support}(B))$$

=

คลับ กับ lift 1.26
1.98 ลบ

Example:

Suppose we have the following information:

- Total number of transactions: 1000
- Transactions containing item A: 600
- Transactions containing item B: 400
- Transactions containing both A and B: 200

Calculate Leverage for Rule A->B:

1. Calculate Support:

- $\text{Support}(A) = \text{Transactions containing A} / \text{Total transactions} = 600 / 1000 = 0.6$
- $\text{Support}(B) = \text{Transactions containing B} / \text{Total transactions} = 400 / 1000 = 0.4$
- $\text{Support}(A \rightarrow B) = \text{Transactions containing A and B} / \text{Total transactions} = 200 / 1000 = 0.2$

2. Calculate Leverage:

- $\text{Leverage}(A \rightarrow B) = \text{Support}(A \rightarrow B) - (\text{Support}(A) * \text{Support}(B))$
- $\text{Leverage}(A \rightarrow B) = 0.2 - (0.6 * 0.4) = 0.2 - 0.24 = -0.04$

→ Interpretation:

- หาก $\text{Leverage}(A \rightarrow B) = 0$ แสดงว่าการเกิดของ A และ B เป็นไปตามคาดหวัง ถ้า A และ B เป็นอิสระต่อ กัน
- หาก $\text{Leverage}(A \rightarrow B) > 0$ แสดงว่าการเกิดของ A และ B เกินค่าคาดหวัง
- หาก $\text{Leverage}(A \rightarrow B) < 0$ แสดงว่าการเกิดของ A และ B น้อยกว่าค่าคาดหวัง

5. Conviction:

Conviction measures the ratio of the expected frequency that the antecedent implies the consequent to the observed frequency of the implication.

ก. เชื่อมต่อ B

$$\text{Conviction}(A \rightarrow B) = (1 - \underline{\text{Support}(B)}) / (1 - \text{Confidence}(A \rightarrow B))$$

ค. เชื่อมต่อ ไม่ซึ่ง B

Example:

Suppose we have the following information:

- Total number of transactions: 1000
- Transactions containing item A: 600
- Transactions containing item B: 400
- Transactions containing both A and B: 300

Calculate Conviction for Rule A->B:

1. Calculate Support and Confidence (same as in previous examples).

2. Calculate Conviction:

- Conviction(A->B) = $(1 - \text{Support}(B)) / (1 - \text{Confidence}(A \rightarrow B))$
- Conviction(A->B) = $(1 - 0.4) / (1 - 0.5) = 0.6 / 0.5 = 1.2$

→ Interpretation:

คลับกัน

- หาก Conviction(A->B) = 1 แสดงว่า A และ B เป็นอิสระกัน
- หาก Conviction(A->B) > 1 แสดงว่าการเกิดของ A และ B ร่วมกันมากกว่าที่คาดหวัง
- หาก Conviction(A->B) < 1 แสดงว่าการเกิดของ A และ B ร่วมกันน้อยกว่าที่คาดหวัง

- สร้างจาก frequent itemsets
- เรียกใช้ association_rules library

from mlxtend.frequent_patterns import association_rules

from mlxtend.frequent_patterns import association_rules

- กำหนดเกณฑ์ในการพิจารณากฎ ห้า 5 ค่า
- supported metrics are 'support', 'confidence', 'lift', 'leverage', and 'conviction'
 - support(A->C) = support(A+C) [aka 'support'], range: [0, 1]
 - confidence(A->C) = support(A+C) / support(A), range: [0, 1]
 - lift(A->C) = confidence(A->C) / support(C), range: [0, inf]
 - leverage(A->C) = support(A->C) - support(A)*support(C), range: [-1, 1]
 - conviction = $[1 - \text{support}(C)] / [1 - \text{confidence}(A \rightarrow C)]$, range: [0, inf]

rules = association_rules(fif, metric='confidence', min_threshold=0.2)

confidence ต้องต่ำกว่า 0.2
min_threshold (20%) หมายความว่า กฎต้องมีความน่าเชื่อถือ 20% ขึ้นไป

rules[rules['lift'] > 1]

	antecedents	consequents	antecedent support	consequent support	support	confidence
0	(sausage, yogurt)	(whole milk)	0.005748	0.157923	0.001470	0.2558
1	(rolls/buns, sausage)	(whole milk)	0.005347	0.157923	0.001136	0.2125

- จึงได้ผลลัพธ์ rules กด 0
= กฎที่มีความน่าเชื่อถือ 20% ขึ้นไป คือ (sausage, yogurt) หรือ (whole milk) ต้องมีความน่าเชื่อถือ 20% ขึ้นไป

- แต่ถ้าต้องการ lift กด 1

! หมายความว่า กฎที่มีความน่าเชื่อถือ 20% ขึ้นไป ต้องมีความน่าเชื่อถือ 20% ขึ้นไป

ตัววัดประสิทธิภาพของกฎ

ค่า Confidence เป็นตัววัดประสิทธิภาพสำหรับ Association Rule โดยเป็นตัวเลขที่แสดงความน่าจะเป็นที่กลุ่มของสินค้า RHS จะถูกหยิบเข้าตะกร้าด้วย หลังจากที่กลุ่มของสินค้า LHS ถูกหยิบเข้าตะกร้าไปแล้ว

หรืออาจกล่าวได้ว่า “หากเราพิจารณาเฉพาะตะกร้าการซื้อขายที่มีกลุ่มสินค้า LHS มีตะกร้าการซื้อขายที่มี RHS อยู่ด้วยเป็นสัดส่วนเท่าไหร่”

ในตัวอย่างร้านค้าของเรา หากพิจารณา $LHS = \{\text{Cereal}\}$ และ $RHS = \{\text{Apple}\}$ จะได้ว่า $\text{Confidence}(\{\text{Cereal}\} \Rightarrow \{\text{Apple}\}) = 2/3$ (ภายใน 3 ตะกร้าที่มีการซื้อขาย Cereal มีจำนวน 2 ตะกร้าที่มีการซื้อขาย Apple ด้วย)

หาก RHS เป็นกลุ่มของสินค้าที่พบเจอในแทบทุกตะกร้าการซื้อขาย (a very frequent itemset) ไม่ว่า LHS จะเป็นกลุ่มของสินค้าใดก็ตาม $\text{Confidence}(LHS \Rightarrow RHS)$ จะมีค่าค่อนข้างสูงเสมอ ด้วยเหตุนี้ จึงมีอีกหนึ่งตัววัดประสิทธิภาพที่พิจารณาความถี่ในการพบกลุ่มของสินค้า RHS ร่วมกับค่า confidence ด้วย

