In [1]:
```
# 如果發生ModuleNotFoundError: No module named 'pandas'
# 安裝套件：!pip install pandas
import pandas as pd
import numpy as np
```

In [2]:
```
# 1.讀取購物車資料
import csv
dataList = []
with open('groceries.csv', newline = '') as f:
    csvReader = csv.reader(f)

    for i in csvReader:
        dataList.append(list(i))

dataList
```

Out[2]:
```
[['citrus fruit', 'semi-finished bread', 'margarine', 'ready soups'],
 ['tropical fruit', 'yogurt', 'coffee'],
 ['whole milk'],
 ['pip fruit', 'yogurt', 'cream cheese ', 'meat spreads'],
 ['other vegetables',
  'whole milk',
  'condensed milk',
  'long life bakery product'],
 ['whole milk', 'butter', 'yogurt', 'rice', 'abrasive cleaner'],
 ['rolls/buns'],
 ['other vegetables',
  'UHT-milk',
  'rolls/buns',
  'bottled beer',
  'liquor (appetizer)'],
 ['pot plants'],
 ['whole milk', 'cereals'],
 ['tropical fruit',
  'other vegetables',
```

In [3]:
```
# 如果沒安裝過mlxtend，安裝套件: conda install -c conda-forge mlxtend
# mlxtend:machine learning extensions（機器學習擴展模組套作，用來轉換真值表）
from mlxtend.preprocessing import TransactionEncoder
# 宣告模型
te = TransactionEncoder()
# 用 fit 訓練資料, 可以知道原始資料有多少唯一值
# 再用 transform 進行轉換
te_ary = te.fit(dataList).transform(dataList)
te_ary
```

Out[3]:
```
array([[False, False, False, ..., False, False, False],
       [False, False, False, ..., False,  True, False],
       [False, False, False, ...,  True, False, False],
       ...,
       [False, False, False, ..., False,  True, False],
       [False, False, False, ..., False, False, False],
       [False, False, False, ..., False, False, False]])
```

In [4]:
```
# 查看欄位
te.columns_
```

Out[4]:
```
['Instant food products',
 'UHT-milk',
 'abrasive cleaner',
 'artif. sweetener',
 'baby cosmetics',
 'baby food',
 'bags',
 'baking powder',
 'bathroom cleaner',
 'beef',
 'berries',
 'beverages',
 'bottled beer',
 'bottled water',
 'brandy',
 'brown bread',
 'butter',
 'butter milk',
 'cake bar',
```

In [5]:
```
# 轉換為 pandas 的 DataFrame(2維)
df = pd.DataFrame(te_ary, columns = te.columns_)
df
# 這種編碼方式叫 One-hot encoding (獨熱編碼)
# 對機器學習而言,相對方便計算, 但若項目很多, 則會面對一個非常巨大的表格
```

Out[5]:

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics	baby food	bags	baking powder	bathroom cleaner	beef	...	turkey	vinegar
0	False	False	False	False	False	False	False	False	False	False	...	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9830	False	False	False	False	False	False	False	False	False	True	...	False	False
9831	False	False	False	False	False	False	False	False	False	False	...	False	False
9832	False	False	False	False	False	False	False	False	False	False	...	False	False
9833	False	False	False	False	False	False	False	False	False	False	...	False	False
9834	False	False	False	False	False	False	False	False	False	False	...	False	True

9835 rows × 169 columns

In [6]:
```
# 2.產生頻繁項目集
# 引用 apriori函數, 並設定最小支持度(min_support)為 0.04
# 將 itemsets改為商品種類的名稱顯示, 呼叫 apriori 函數時, 加入 use_colnames= True
from mlxtend.frequent_patterns import apriori
frequent_itemsets = apriori(df, min_support = 0.04, use_colnames = True)
frequent_itemsets
# 其中 itemsets 代表商品種類的索引編號
```

Out[6]:

	support	itemsets
0	0.052466	(beef)
1	0.080529	(bottled beer)
2	0.110524	(bottled water)
3	0.064870	(brown bread)
4	0.055414	(butter)
5	0.077682	(canned beer)
6	0.042908	(chicken)
7	0.049619	(chocolate)
8	0.082766	(citrus fruit)
9	0.058058	(coffee)
10	0.053279	(curd)
11	0.063447	(domestic eggs)
12	0.058973	(frankfurter)
13	0.048094	(frozen vegetables)
14	0.072293	(fruit/vegetable juice)
15	0.058566	(margarine)
16	0.052364	(napkins)
17	0.079817	(newspapers)
18	0.193493	(other vegetables)
19	0.088968	(pastry)
20	0.075648	(pip fruit)
21	0.057651	(pork)
22	0.183935	(rolls/buns)
23	0.108998	(root vegetables)
24	0.093950	(sausage)
25	0.098526	(shopping bags)
26	0.174377	(soda)
27	0.104931	(tropical fruit)
28	0.071683	(whipped/sour cream)
29	0.042095	(white bread)
30	0.255516	(whole milk)
31	0.139502	(yogurt)
32	0.042603	(rolls/buns, other vegetables)
33	0.047382	(other vegetables, root vegetables)
34	0.074835	(whole milk, other vegetables)
35	0.043416	(yogurt, other vegetables)
36	0.056634	(rolls/buns, whole milk)
37	0.048907	(whole milk, root vegetables)
38	0.040061	(soda, whole milk)
39	0.042298	(whole milk, tropical fruit)
40	0.056024	(yogurt, whole milk)

In [7]:
```
type(frequent_itemsets)
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [8]:
```
# 增加 itemsets的長度 length 欄位
# .apply() 的功能與迴圈（Loop）相似，讓你沿著某個維度（Axis）重複執行某項工作
# lambda是一種沒有函數名稱的匿名函數寫法, 透過 len(x)計算x長度後將值回傳
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x:len(x))
frequent_itemsets
```

Out[8]:

	support	itemsets	length
0	0.052466	(beef)	1
1	0.080529	(bottled beer)	1
2	0.110524	(bottled water)	1
3	0.064870	(brown bread)	1
4	0.055414	(butter)	1
5	0.077682	(canned beer)	1
6	0.042908	(chicken)	1
7	0.049619	(chocolate)	1
8	0.082766	(citrus fruit)	1
9	0.058058	(coffee)	1
10	0.053279	(curd)	1
11	0.063447	(domestic eggs)	1
12	0.058973	(frankfurter)	1
13	0.048094	(frozen vegetables)	1
14	0.072293	(fruit/vegetable juice)	1
15	0.058566	(margarine)	1
16	0.052364	(napkins)	1
17	0.079817	(newspapers)	1
18	0.193493	(other vegetables)	1
19	0.088968	(pastry)	1
20	0.075648	(pip fruit)	1
21	0.057651	(pork)	1
22	0.183935	(rolls/buns)	1
23	0.108998	(root vegetables)	1
24	0.093950	(sausage)	1
25	0.098526	(shopping bags)	1
26	0.174377	(soda)	1
27	0.104931	(tropical fruit)	1
28	0.071683	(whipped/sour cream)	1
29	0.042095	(white bread)	1
30	0.255516	(whole milk)	1
31	0.139502	(yogurt)	1
32	0.042603	(rolls/buns, other vegetables)	2
33	0.047382	(other vegetables, root vegetables)	2
34	0.074835	(whole milk, other vegetables)	2
35	0.043416	(yogurt, other vegetables)	2
36	0.056634	(rolls/buns, whole milk)	2
37	0.048907	(whole milk, root vegetables)	2
38	0.040061	(soda, whole milk)	2
39	0.042298	(whole milk, tropical fruit)	2
40	0.056024	(yogurt, whole milk)	2

In [9]:
```
# 3.產生關聯規則
# 信心水準(Confidence): 兩物品同時出現的條件機率，整條關聯規則的信心度
# 最小信心度: 0.4
# 以上兩個值由人為設定, 通常由資料科學家依過去經驗決定的
from mlxtend.frequent_patterns import association_rules
association_rules(frequent_itemsets, metric = 'confidence', min_threshold = 0.4)
# antecedents: 前項; consequents: 後項
# support: 支持度; confidence: 信心度; lift:提升度
```

Out[9]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhan
0	(root vegetables)	(other vegetables)	0.108998	0.193493	0.047382	0.434701	2.246605	0.026291	1.426693	
1	(root vegetables)	(whole milk)	0.108998	0.255516	0.048907	0.448694	1.756031	0.021056	1.350401	
2	(tropical fruit)	(whole milk)	0.104931	0.255516	0.042298	0.403101	1.577595	0.015486	1.247252	
3	(yogurt)	(whole milk)	0.139502	0.255516	0.056024	0.401603	1.571735	0.020379	1.244132	

In [10]:
```
# 另一個角度分析: 提升度(Lift):衡量項目集之間的相關性/依賴，整條關聯規則的提升度
# Lift(A -> B) = Confidence(A -> B) / P(B) = P(B|A) / P(B)
# 或 = P(A∪B)/(P(A)*P(B))
# 出現商品B的機率之下, 出現商品A的情況下，出現商品B的機率
# 提升度 > 1 : 表示數據間越相關，呈正相關 (這兩個項目集提高了彼此的可能性)
# 提升度 = 1 : 表示兩數據獨立，不相關 (A 和 B 是獨立的)
# 提升度 < 1 : 表示兩數據呈負相關 (顧客往往不會同時購買 A 和 B)
rules = association_rules(frequent_itemsets, metric = 'lift', min_threshold = 1.7)
rules
```

Out[10]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhan
0	(other vegetables)	(root vegetables)	0.193493	0.108998	0.047382	0.244877	2.246605	0.026291	1.179941	
1	(root vegetables)	(other vegetables)	0.108998	0.193493	0.047382	0.434701	2.246605	0.026291	1.426693	
2	(whole milk)	(root vegetables)	0.255516	0.108998	0.048907	0.191405	1.756031	0.021056	1.101913	
3	(root vegetables)	(whole milk)	0.108998	0.255516	0.048907	0.448694	1.756031	0.021056	1.350401	

In [11]:
```
# 增加 antecedents_len 欄位
rules['antecedent_len'] = rules['antecedents'].apply(lambda x:len(x))
rules
```

Out[11]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhan
0	(other vegetables)	(root vegetables)	0.193493	0.108998	0.047382	0.244877	2.246605	0.026291	1.179941	
1	(root vegetables)	(other vegetables)	0.108998	0.193493	0.047382	0.434701	2.246605	0.026291	1.426693	
2	(whole milk)	(root vegetables)	0.255516	0.108998	0.048907	0.191405	1.756031	0.021056	1.101913	
3	(root vegetables)	(whole milk)	0.108998	0.255516	0.048907	0.448694	1.756031	0.021056	1.350401	

In [12]:
```
# 關聯規則：X => Y[support,confidence,lift]
# 最小支持度(Minimum support)= 4% 與最小信心度(Minimum confidence) = 40% 的條件下
# 增加篩選條件
rules[(rules['antecedent_len'] >= 1) & (rules['confidence'] >= 0.4) & (rules['lift'] >= 1.7)
```

Out[12]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhan
1	(root vegetables)	(other vegetables)	0.108998	0.193493	0.047382	0.434701	2.246605	0.026291	1.426693	
3	(root vegetables)	(whole milk)	0.108998	0.255516	0.048907	0.448694	1.756031	0.021056	1.350401	

In [13]:
```
import csv
with open('gege.csv', 'w', newline='') as file:
    writer = csv.writer(file, delimiter=',')
    writer.writerows(te.inverse_transform(te_ary))
file.close()
```

In [ ]:
```

```