Data Mining

Exercise of Association Rule Learning

Jia-Wei Chang

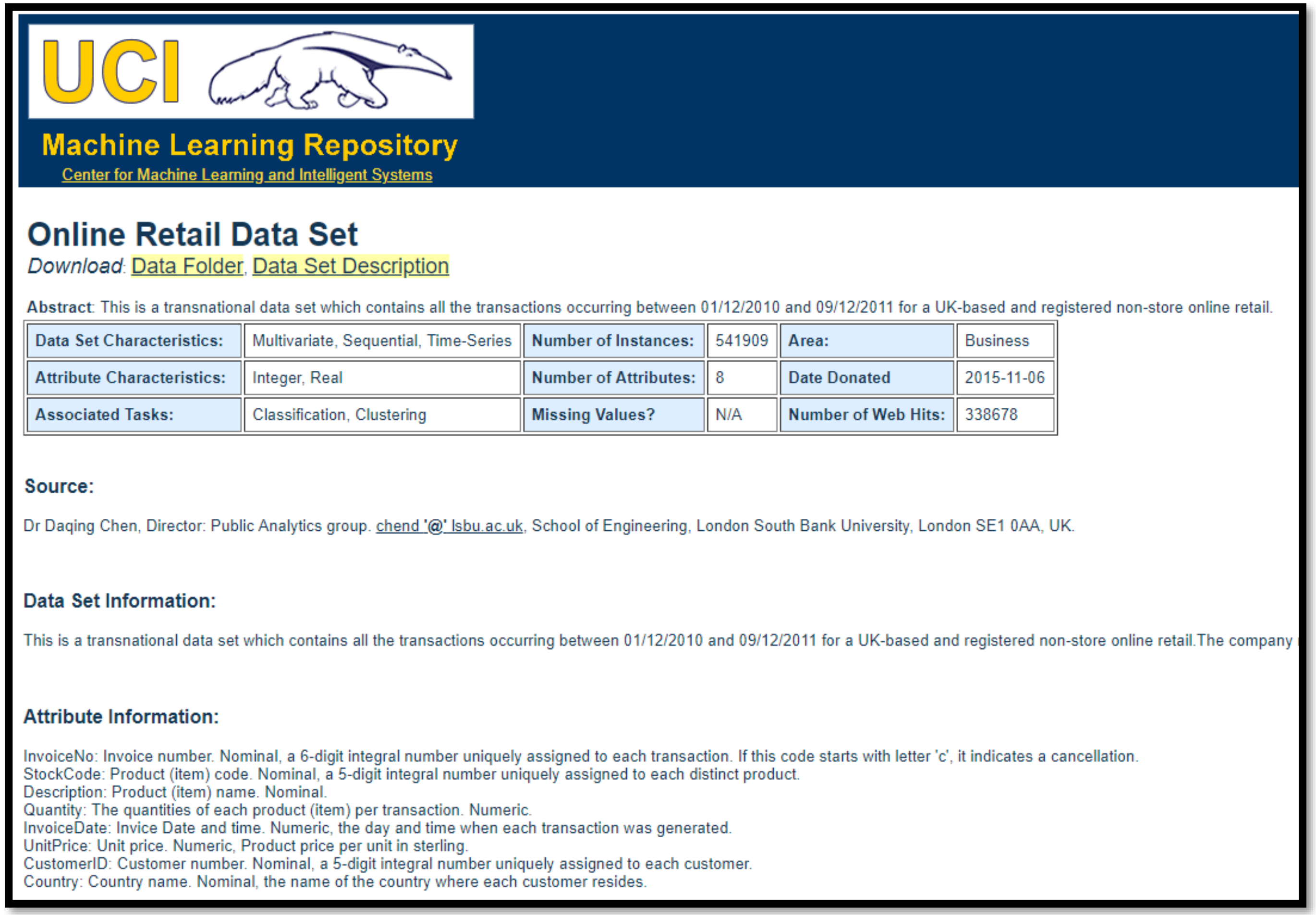
Department of Computer Science and Information Engineering

National Taichung University of Science and Technology

Apriori

Case 1

Data Description



UCI

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Online Retail Data Set

Download: Data Folder, Data Set Description

Abstract: This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.

Data Set Characteristics:	Multivariate, Sequential, Time-Series	**Number of Instances:**	541909	**Area:**	Business
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	8	**Date Donated**	2015-11-06
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	338678

Source:

Dr Daqing Chen, Director: Public Analytics group. chend '@' lsbu.ac.uk, School of Engineering, London South Bank University, London SE1 0AA, UK.

Data Set Information:

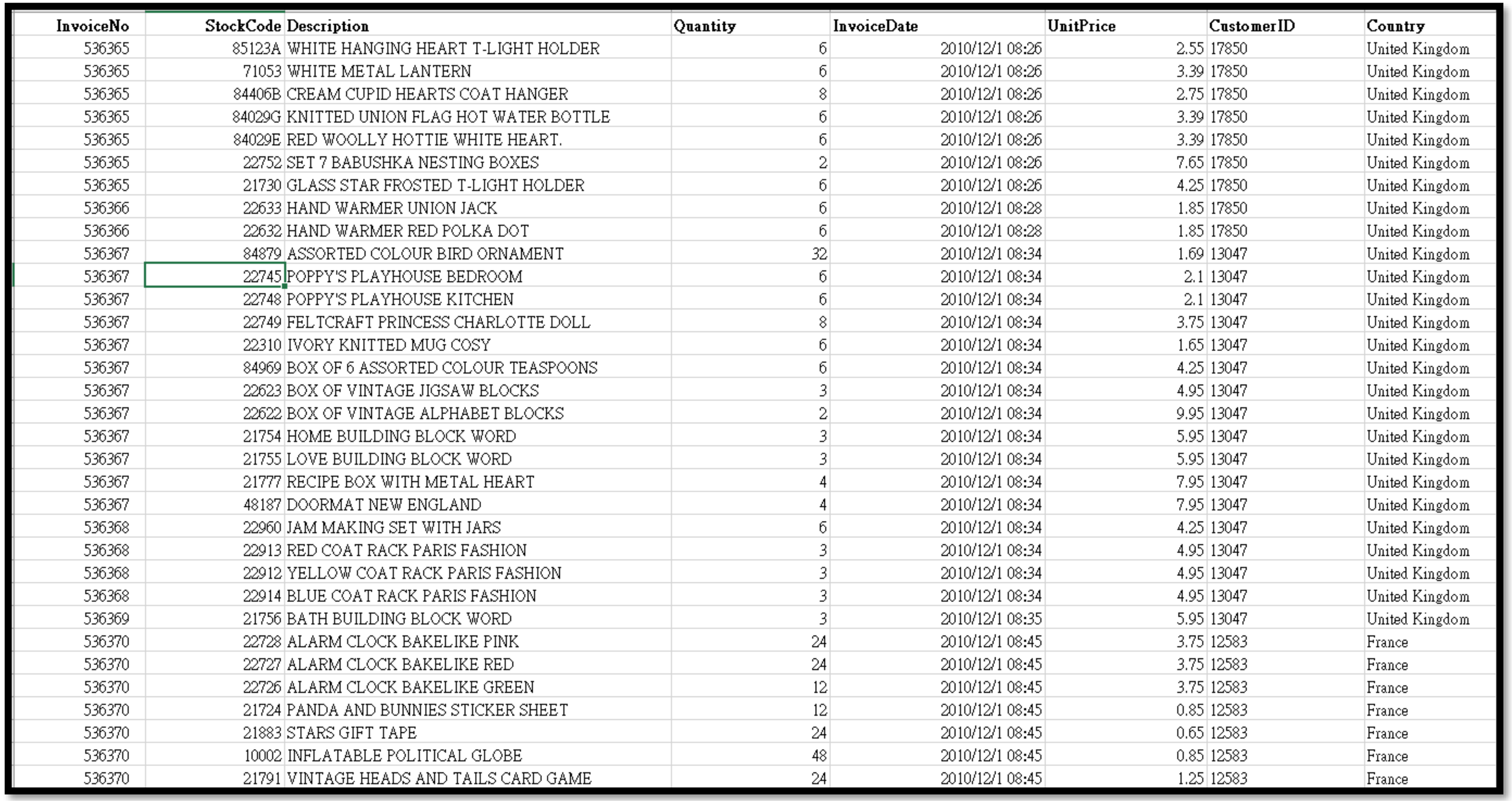
This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company

Attribute Information:

InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
Description: Product (item) name. Nominal.
Quantity: The quantities of each product (item) per transaction. Numeric.
InvoiceDate: Invice Date and time. Numeric, the day and time when each transaction was generated.
UnitPrice: Unit price. Numeric, Product price per unit in sterling.
CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
Country: Country name. Nominal, the name of the country where each customer resides.

https://archive.ics.uci.edu/ml/datasets/Online%20Retail

Data Description

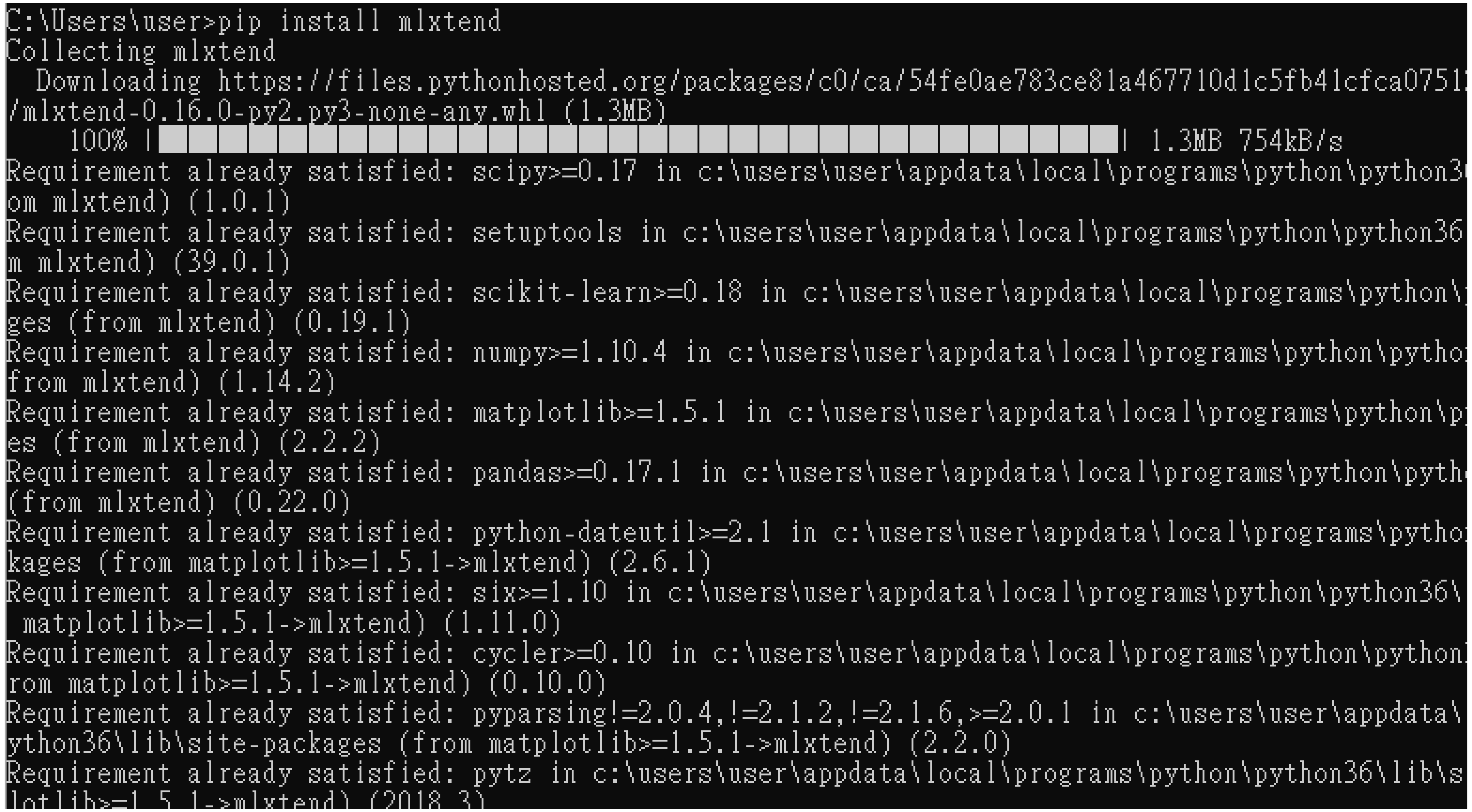


InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010/12/1 08:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	2010/12/1 08:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010/12/1 08:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010/12/1 08:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010/12/1 08:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010/12/1 08:26	7.65	17850	United Kingdom
536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010/12/1 08:26	4.25	17850	United Kingdom
536366	22633	HAND WARMER UNION JACK	6	2010/12/1 08:28	1.85	17850	United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	2010/12/1 08:28	1.85	17850	United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010/12/1 08:34	1.69	13047	United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	2010/12/1 08:34	2.1	13047	United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	2010/12/1 08:34	2.1	13047	United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	2010/12/1 08:34	3.75	13047	United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	2010/12/1 08:34	1.65	13047	United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	2010/12/1 08:34	4.25	13047	United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	2010/12/1 08:34	4.95	13047	United Kingdom
536367	22622	BOX OF VINTAGE ALPHABET BLOCKS	2	2010/12/1 08:34	9.95	13047	United Kingdom
536367	21754	HOME BUILDING BLOCK WORD	3	2010/12/1 08:34	5.95	13047	United Kingdom
536367	21755	LOVE BUILDING BLOCK WORD	3	2010/12/1 08:34	5.95	13047	United Kingdom
536367	21777	RECIPE BOX WITH METAL HEART	4	2010/12/1 08:34	7.95	13047	United Kingdom
536367	48187	DOORMAT NEW ENGLAND	4	2010/12/1 08:34	7.95	13047	United Kingdom
536368	22960	JAM MAKING SET WITH JARS	6	2010/12/1 08:34	4.25	13047	United Kingdom
536368	22913	RED COAT RACK PARIS FASHION	3	2010/12/1 08:34	4.95	13047	United Kingdom
536368	22912	YELLOW COAT RACK PARIS FASHION	3	2010/12/1 08:34	4.95	13047	United Kingdom
536368	22914	BLUE COAT RACK PARIS FASHION	3	2010/12/1 08:34	4.95	13047	United Kingdom
536369	21756	BATH BUILDING BLOCK WORD	3	2010/12/1 08:35	5.95	13047	United Kingdom
536370	22728	ALARM CLOCK BAKELIKE PINK	24	2010/12/1 08:45	3.75	12583	France
536370	22727	ALARM CLOCK BAKELIKE RED	24	2010/12/1 08:45	3.75	12583	France
536370	22726	ALARM CLOCK BAKELIKE GREEN	12	2010/12/1 08:45	3.75	12583	France
536370	21724	PANDA AND BUNNIES STICKER SHEET	12	2010/12/1 08:45	0.85	12583	France
536370	21883	STARS GIFT TAPE	24	2010/12/1 08:45	0.65	12583	France
536370	10002	INFLATABLE POLITICAL GLOBE	48	2010/12/1 08:45	0.85	12583	France
536370	21791	VINTAGE HEADS AND TAILS CARD GAME	24	2010/12/1 08:45	1.25	12583	France

https://archive.ics.uci.edu/ml/datasets/Online%20Retail

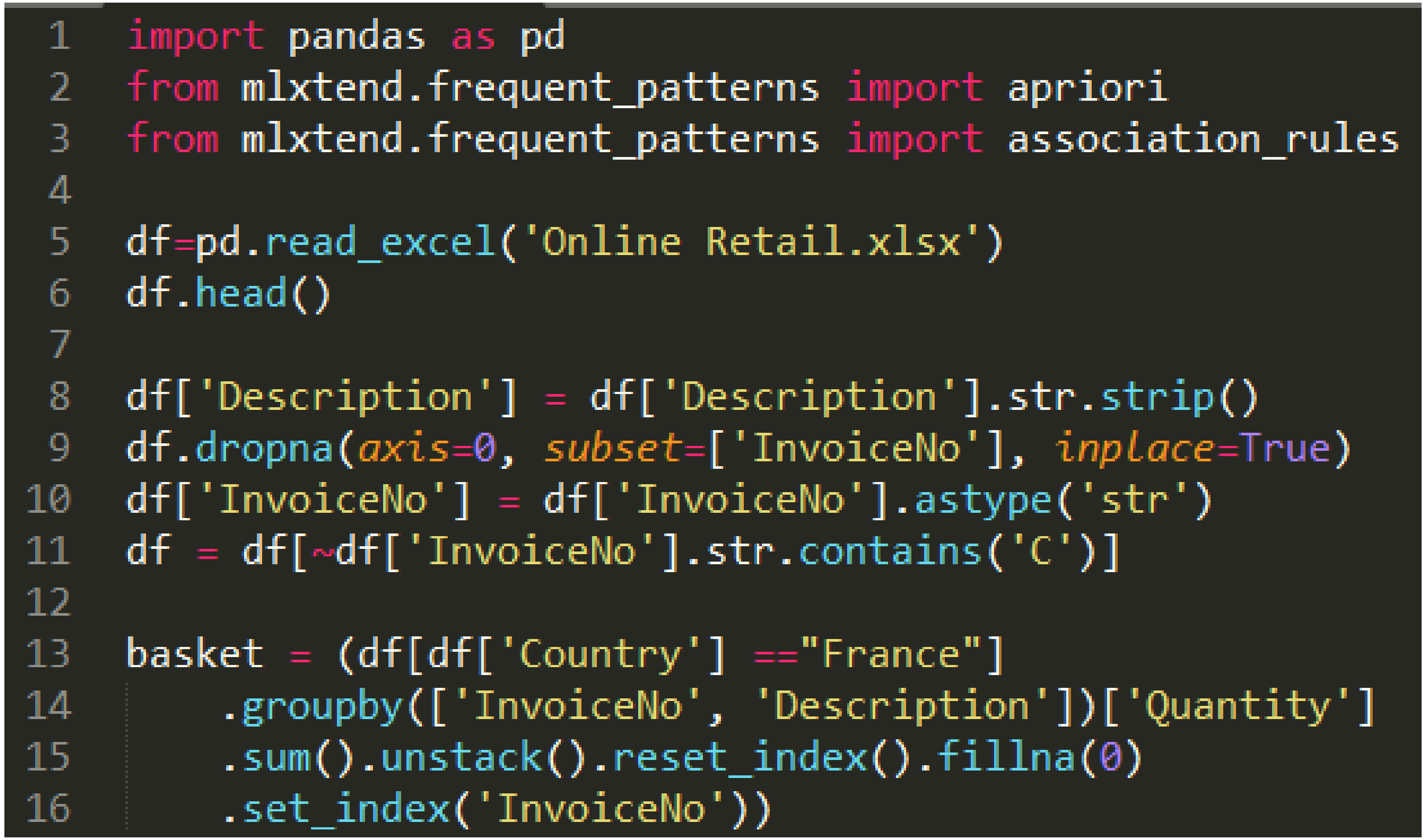
Step 1 - Install mlxtend

- pip install mlxtend xlrd



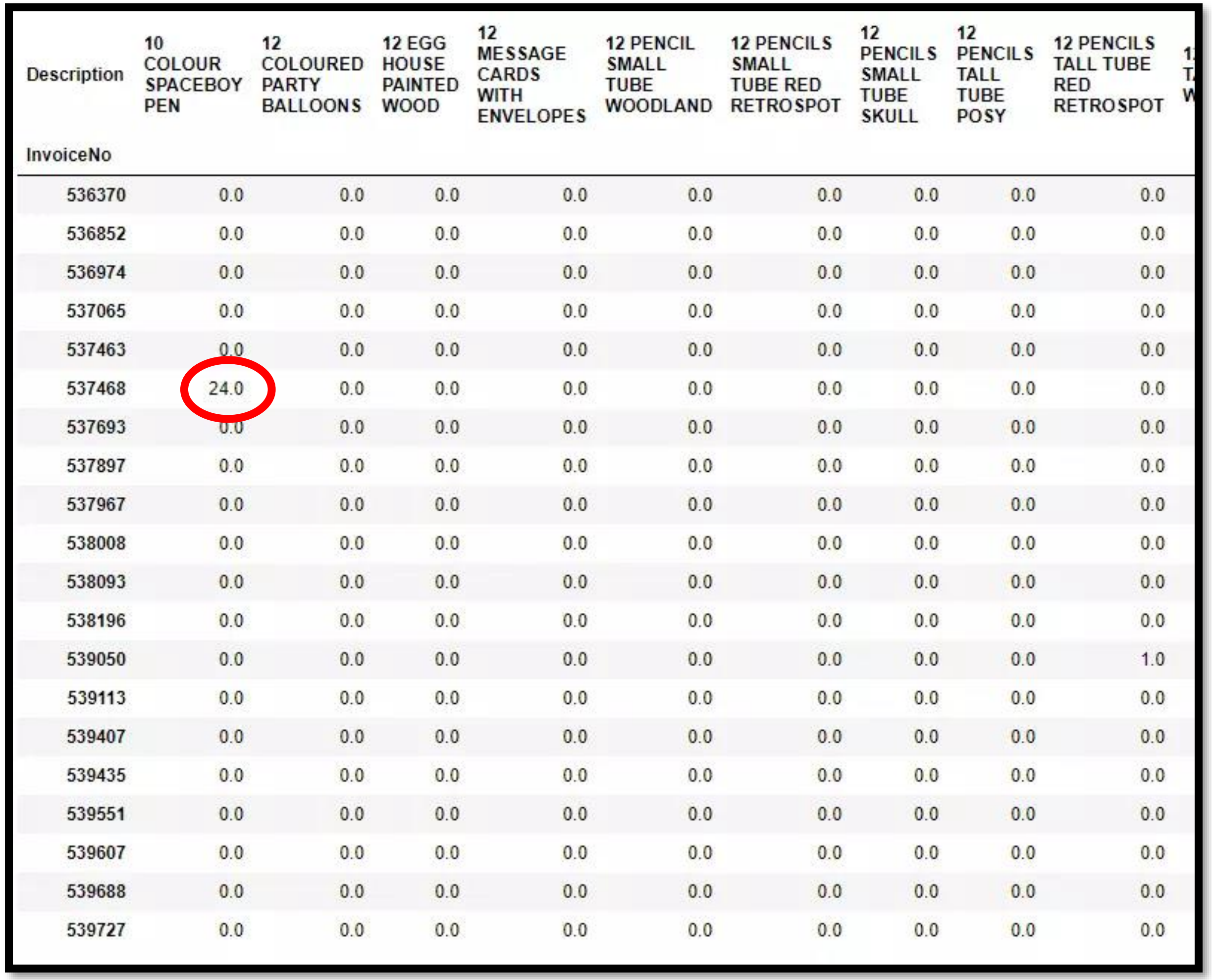
```
C:\Users\user>pip install mlxtend
Collecting mlxtend
  Downloading https://files.pythonhosted.org/packages/c0/ca/54fe0ae783ce81a467710d1c5fb41cfca0751
/mlxtend-0.16.0-py2.py3-none-any.whl (1.3MB)
    100% |████████████████████████████████| 1.3MB 754kB/s
Requirement already satisfied: scipy>=0.17 in c:\users\user\appdata\local\programs\python\python3
om mlxtend) (1.0.1)
Requirement already satisfied: setuptools in c:\users\user\appdata\local\programs\python\python36
m mlxtend) (39.0.1)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\user\appdata\local\programs\python\
ges (from mlxtend) (0.19.1)
Requirement already satisfied: numpy>=1.10.4 in c:\users\user\appdata\local\programs\python\pytho
from mlxtend) (1.14.2)
Requirement already satisfied: matplotlib>=1.5.1 in c:\users\user\appdata\local\programs\python\p
es (from mlxtend) (2.2.2)
Requirement already satisfied: pandas>=0.17.1 in c:\users\user\appdata\local\programs\python\pyth
(from mlxtend) (0.22.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\user\appdata\local\programs\pytho
kages (from matplotlib>=1.5.1->mlxtend) (2.6.1)
Requirement already satisfied: six>=1.10 in c:\users\user\appdata\local\programs\python\python36\
 matplotlib>=1.5.1->mlxtend) (1.11.0)
Requirement already satisfied: cycler>=0.10 in c:\users\user\appdata\local\programs\python\python
rom matplotlib>=1.5.1->mlxtend) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\user\appdata\
ython36\lib\site-packages (from matplotlib>=1.5.1->mlxtend) (2.2.0)
Requirement already satisfied: pytz in c:\users\user\appdata\local\programs\python\python36\lib\s
lotlib>=1.5.1->mlxtend) (2018.3)
```

Step 2 – Import Libs & Data Preprocessing



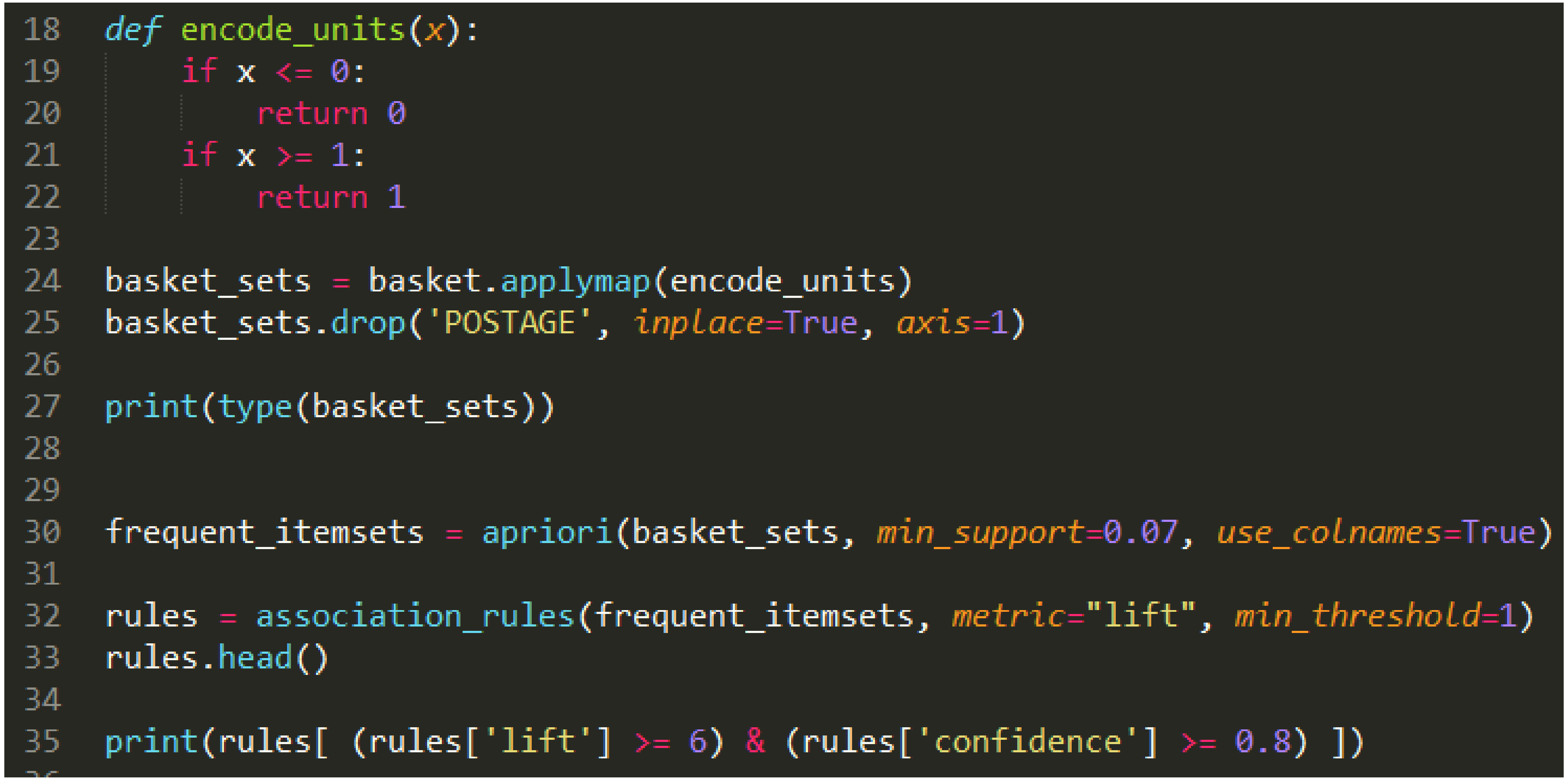
```python
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

df=pd.read_excel('Online Retail.xlsx')
df.head()

df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]

basket = (df[df['Country'] =="France"]
    .groupby(['InvoiceNo', 'Description'])['Quantity']
    .sum().unstack().reset_index().fillna(0)
    .set_index('InvoiceNo'))
```

Step 2 – Import Libs & Data Preprocessing



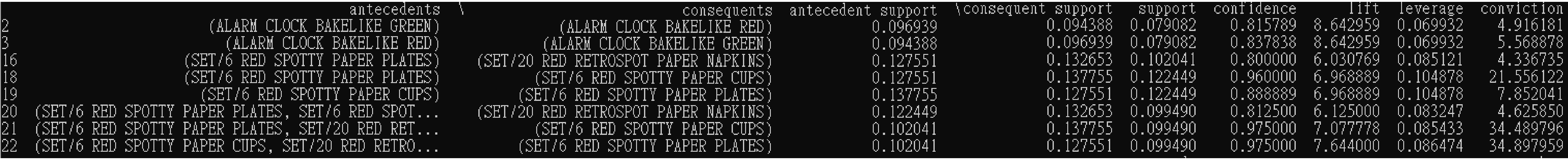
Description InvoiceNo	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 EGG HOUSE PAINTED WOOD	12 MESSAGE CARDS WITH ENVELOPES	12 PENCIL SMALL TUBE WOODLAND	12 PENCILS SMALL TUBE RED RETROSPOT	12 PENCILS SMALL TUBE SKULL	12 PENCILS TALL TUBE POSY	12 PENCILS TALL TUBE RED RETROSPOT	1 T W
536370	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
536852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
536974	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
537065	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
537463	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
537468	24.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
537693	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
537897	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
537967	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
538008	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
538093	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
538196	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539050	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
539113	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539407	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539435	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539551	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539607	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539688	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
539727	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Step 3 – Data Preprocessing & Association Rule Learning



```
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units)
basket_sets.drop('POSTAGE', inplace=True, axis=1)

print(type(basket_sets))


frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head()

print(rules[ (rules['lift'] >= 6) & (rules['confidence'] >= 0.8) ])
```

Step 3 – Results

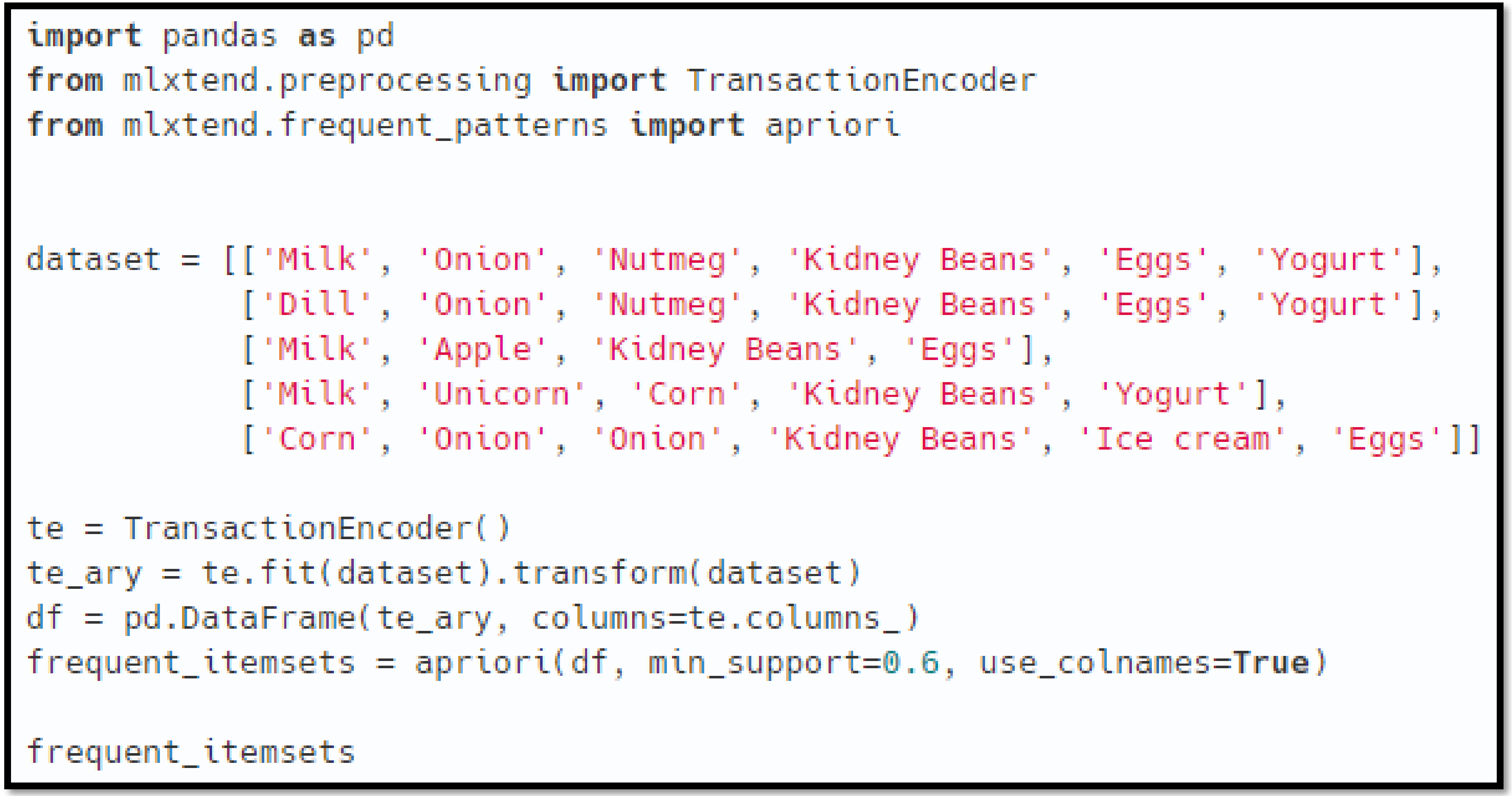


```
                                          antecedents  \                           consequents  antecedent support  \consequent support   support  confidence      lift  leverage  conviction
2                        (ALARM CLOCK BAKELIKE GREEN)               (ALARM CLOCK BAKELIKE RED)            0.096939             0.094388  0.079082    0.815789  8.642959  0.069932    4.916181
3                          (ALARM CLOCK BAKELIKE RED)             (ALARM CLOCK BAKELIKE GREEN)            0.094388             0.096939  0.079082    0.837838  8.642959  0.069932    5.568878
16                    (SET/6 RED SPOTTY PAPER PLATES)     (SET/20 RED RETROSPOT PAPER NAPKINS)            0.127551             0.132653  0.102041    0.800000  6.030769  0.085121    4.336735
18                    (SET/6 RED SPOTTY PAPER PLATES)            (SET/6 RED SPOTTY PAPER CUPS)            0.127551             0.137755  0.122449    0.960000  6.968889  0.104878   21.556122
19                      (SET/6 RED SPOTTY PAPER CUPS)          (SET/6 RED SPOTTY PAPER PLATES)            0.137755             0.127551  0.122449    0.888889  6.968889  0.104878    7.852041
20  (SET/6 RED SPOTTY PAPER PLATES, SET/6 RED SPOT...     (SET/20 RED RETROSPOT PAPER NAPKINS)            0.122449             0.132653  0.099490    0.812500  6.125000  0.083247    4.625850
21  (SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...            (SET/6 RED SPOTTY PAPER CUPS)            0.102041             0.137755  0.099490    0.975000  7.077778  0.085433   34.489796
22  (SET/6 RED SPOTTY PAPER CUPS, SET/20 RED RETRO...          (SET/6 RED SPOTTY PAPER PLATES)            0.102041             0.127551  0.099490    0.975000  7.644000  0.086474   34.897959
```

Apriori

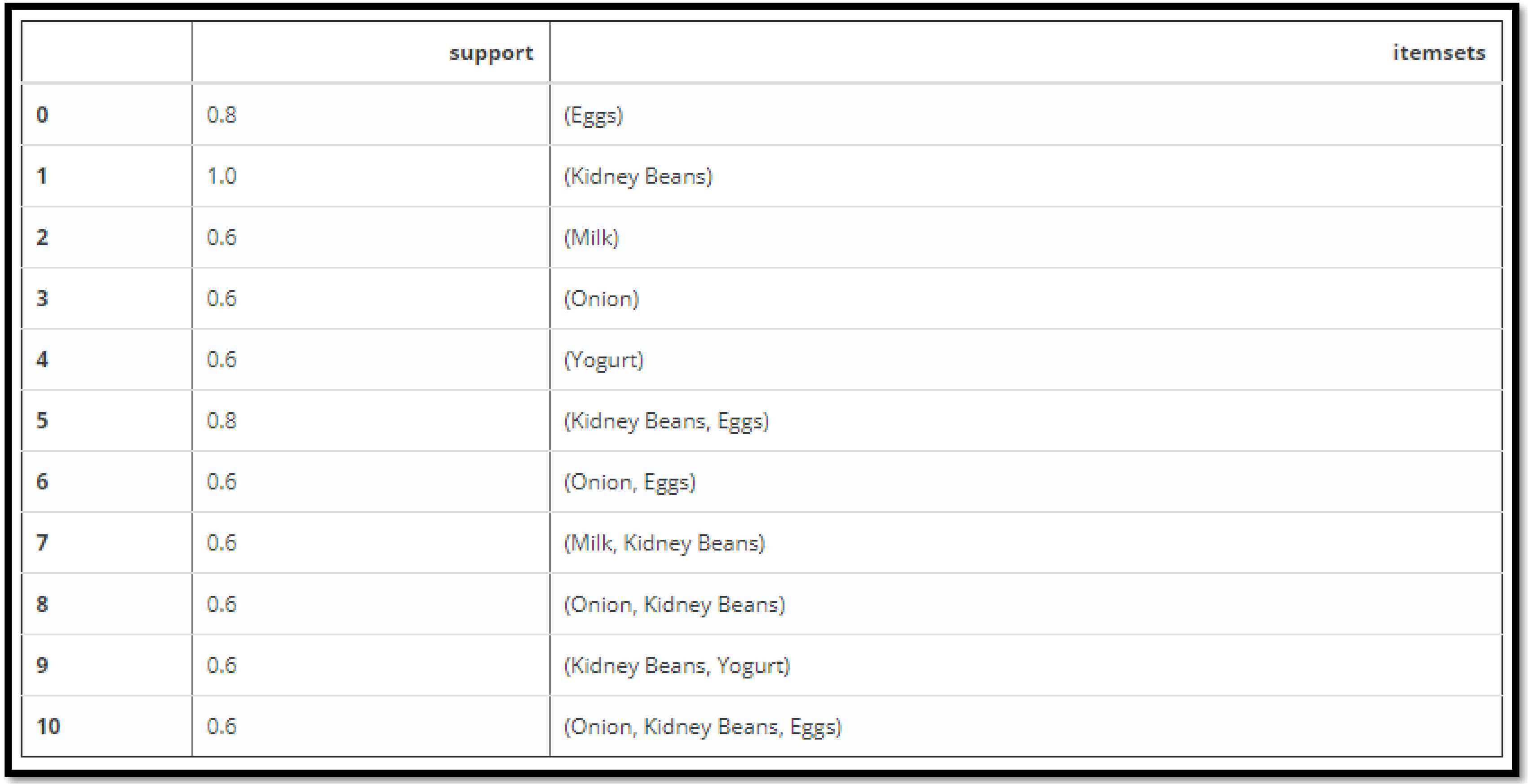
Case 2

Generating Association Rules from Frequent Itemsets



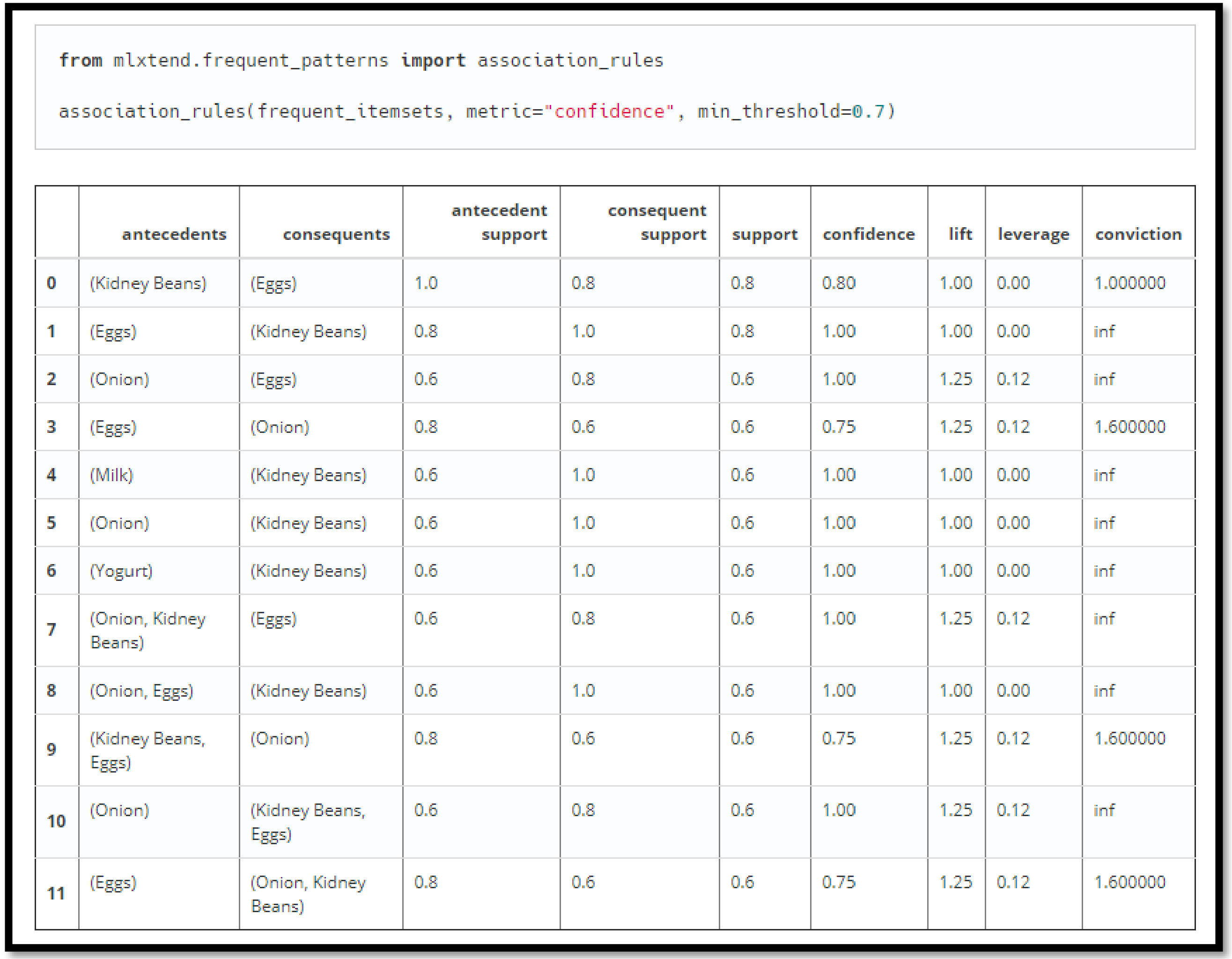
```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori


dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)

frequent_itemsets
```

Generating Association Rules from Frequent Itemsets



	support	itemsets
0	0.8	(Eggs)
1	1.0	(Kidney Beans)
2	0.6	(Milk)
3	0.6	(Onion)
4	0.6	(Yogurt)
5	0.8	(Kidney Beans, Eggs)
6	0.6	(Onion, Eggs)
7	0.6	(Milk, Kidney Beans)
8	0.6	(Onion, Kidney Beans)
9	0.6	(Kidney Beans, Yogurt)
10	0.6	(Onion, Kidney Beans, Eggs)

Generating Association Rules from Frequent Itemsets

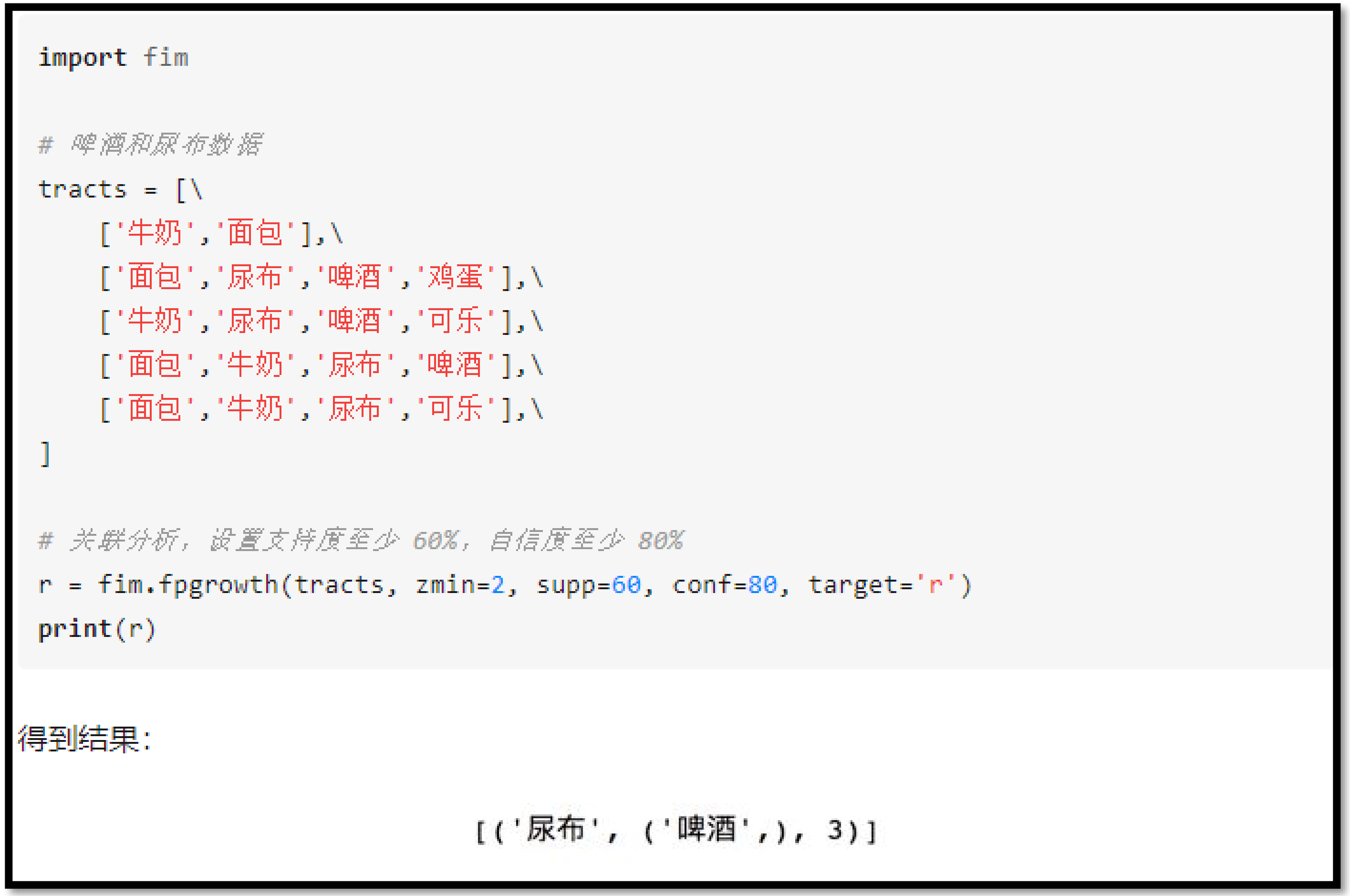


```
from mlxtend.frequent_patterns import association_rules

association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Kidney Beans)	(Eggs)	1.0	0.8	0.8	0.80	1.00	0.00	1.000000
1	(Eggs)	(Kidney Beans)	0.8	1.0	0.8	1.00	1.00	0.00	inf
2	(Onion)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
3	(Eggs)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000
4	(Milk)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
5	(Onion)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
6	(Yogurt)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
7	(Onion, Kidney Beans)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
8	(Onion, Eggs)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
9	(Kidney Beans, Eggs)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000
10	(Onion)	(Kidney Beans, Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
11	(Eggs)	(Onion, Kidney Beans)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000

Other Tools

- PyFIM
- http://www.borgelt.net/pyfim.html



```
import fim

# 啤酒和尿布数据
tracts = [\
    ['牛奶','面包'],\
    ['面包','尿布','啤酒','鸡蛋'],\
    ['牛奶','尿布','啤酒','可乐'],\
    ['面包','牛奶','尿布','啤酒'],\
    ['面包','牛奶','尿布','可乐'],\
]

# 关联分析，设置支持度至少 60%，自信度至少 80%
r = fim.fpgrowth(tracts, zmin=2, supp=60, conf=80, target='r')
print(r)
```

得到结果：

```
[('尿布', ('啤酒',), 3)]
```

Thank you

Reference

1. http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/
2. https://pbpython.com/market-basket-analysis.html
3. https://zhuanlan.zhihu.com/p/30600248