

```
In [2]: import numpy as np
import pandas as pd
from mlxtend.frequent_patterns import apriori, association_rules
```

```
In [4]: data=pd.read_excel("C:\\Users\\om\\Downloads\\online Retail.xlsx")
data.head(10)
```

```
Out[4]:
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047.0	United Kingdom

```
In [5]: data.shape
```

```
Out[5]: (541909, 8)
```

```
In [6]: data.columns
```

```
Out[6]: Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
              'UnitPrice', 'CustomerID', 'Country'],
              dtype='object')
```

```
In [8]:
```

```
data.Country.unique()
```

```
Out[8]: array(['United Kingdom', 'France', 'Australia', 'Netherlands', 'Germany',
      'Norway', 'EIRE', 'Switzerland', 'Spain', 'Poland', 'Portugal',
      'Italy', 'Belgium', 'Lithuania', 'Japan', 'Iceland',
      'Channel Islands', 'Denmark', 'Cyprus', 'Sweden', 'Austria',
      'Israel', 'Finland', 'Bahrain', 'Greece', 'Hong Kong', 'Singapore',
      'Lebanon', 'United Arab Emirates', 'Saudi Arabia',
      'Czech Republic', 'Canada', 'Unspecified', 'Brazil', 'USA',
      'European Community', 'Malta', 'RSA'], dtype=object)
```

```
In [9]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   InvoiceNo        541909 non-null object
1   StockCode       541909 non-null object
2   Description      540455 non-null object
3   Quantity        541909 non-null int64
4   InvoiceDate      541909 non-null datetime64[ns]
5   UnitPrice       541909 non-null float64
6   CustomerID      406829 non-null float64
7   Country         541909 non-null object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

```
In [10]: data.describe()
```

```
Out[10]:
```

	Quantity	UnitPrice	CustomerID
<b>count</b>	541909.000000	541909.000000	406829.000000
<b>mean</b>	9.552250	4.611114	15287.690570
<b>std</b>	218.081158	96.759853	1713.600303
<b>min</b>	-80995.000000	-11062.060000	12346.000000
<b>25%</b>	1.000000	1.250000	13953.000000
<b>50%</b>	3.000000	2.080000	15152.000000
<b>75%</b>	10.000000	4.130000	16791.000000
<b>max</b>	80995.000000	38970.000000	18287.000000

```
In [11]: data['Description']=data['Description'].str.strip()
```

```
In [13]: data.dropna(axis=0,subset=['InvoiceNo'],inplace=True)
data['InvoiceNo']=data['InvoiceNo'].astype('str')
data=data[~data['InvoiceNo'].str.contains('C')]
```

```
In [14]: data.shape
```

Out[14]: (532621, 8)

In [15]: `print(541909-532621)`

9288

In [17]: `basket_France=(data[data['Country']=='France'].groupby(['InvoiceNo','Description'])  
 ['Quantity'].sum().unstack().reset_index().fillna(0).set_index('InvoiceNo'  
basket_UK=(data[data['Country']=='United Kingdom'].groupby(['InvoiceNo','Description'])  
 ['Quantity'].sum().unstack().reset_index().fillna(0).set_index('InvoiceNo'  
basket_Por=(data[data['Country']=='Portugal'].groupby(['InvoiceNo','Description'])  
 ['Quantity'].sum().unstack().reset_index().fillna(0).set_index('InvoiceNo'  
basket_Sweden=(data[data['Country']=='Sweden'].groupby(['InvoiceNo','Description'])  
 ['Quantity'].sum().unstack().reset_index().fillna(0).set_index('InvoiceNo`

In [19]: `def hot_encode(x):  
 if x<=0:  
 return 0  
 else:  
 return 1  
basket_encoded = basket_France.applymap(hot_encode)  
basket_France = basket_encoded  
basket_encoded = basket_UK.applymap(hot_encode)  
basket_UK = basket_encoded  
basket_encoded = basket_Por.applymap(hot_encode)  
basket_Por = basket_encoded  
basket_encoded = basket_Sweden.applymap(hot_encode)  
basket_Sweden = basket_encoded`

In [21]: `frq_items= apriori(basket_France,min_support=0.05,use_colnames=True)  
rules=association_rules(frq_items,metric='lift',min_threshold=1)  
rules=rules.sort_values(['confidence','lift'],ascending=[False,False])  
rules.head()`

Out[21]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
44	(JUMBO BAG WOODLAND ANIMALS)	(POSTAGE)	0.076531	0.765306	0.076531	1.000	1.306667	0.017961	
258	(RED TOADSTOOL LED NIGHT LIGHT, PLASTERS IN TI...	(POSTAGE)	0.051020	0.765306	0.051020	1.000	1.306667	0.011974	
270	(PLASTERS IN TIN WOODLAND ANIMALS, RED TOADSTO...	(POSTAGE)	0.053571	0.765306	0.053571	1.000	1.306667	0.012573	

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conv
<b>301</b>	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.127551	0.099490	0.975	7.644000	0.086474	34.8
<b>300</b>	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.137755	0.099490	0.975	7.077778	0.085433	34.4

```
In [22]: frq_items= apriori(basket_UK,min_support=0.05,use_colnames=True)
rules=association_rules(frq_items,metric='lift',min_threshold=1)
rules=rules.sort_values(['confidence','lift'],ascending=[False,False])
rules.head()
```

```
Out[22]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
--	-------------	-------------	--------------------	--------------------	---------	------------	------	----------	------------

```
In [23]: frq_items= apriori(basket_Por,min_support=0.05,use_colnames=True)
rules=association_rules(frq_items,metric='lift',min_threshold=1)
rules=rules.sort_values(['confidence','lift'],ascending=[False,False])
rules.head()
```

```
Out[23]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	cc
<b>1170</b>	(SET 12 COLOUR PENCILS SPACEBOY)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	
<b>1171</b>	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET 12 COLOUR PENCILS SPACEBOY)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	
<b>1172</b>	(SET OF 4 KNICK KNACK TINS LONDON)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	
<b>1173</b>	(SET 12 COLOUR PENCILS DOLLY GIRL)	(SET OF 4 KNICK KNACK TINS LONDON)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	
<b>1174</b>	(SET OF 4 KNICK KNACK TINS POPPIES)	(SET 12 COLOUR PENCILS DOLLY GIRL)	0.051724	0.051724	0.051724	1.0	19.333333	0.049049	

In [24]:

```
frq_items= apriori(basket_Sweden,min_support=0.05,use_colnames=True)
rules=association_rules(frq_items,metric='lift',min_threshold=1)
rules=rules.sort_values(['confidence','lift'],ascending=[False,False])
rules.head()
```

Out[24]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	convictic
0	(PACK OF 72 SKULL CAKE CASES)	(12 PENCILS SMALL TUBE SKULL)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	i
1	(12 PENCILS SMALL TUBE SKULL)	(PACK OF 72 SKULL CAKE CASES)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	i
4	(36 DOILIES DOLLY GIRL)	(ASSORTED BOTTLE TOP MAGNETS)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	i
5	(ASSORTED BOTTLE TOP MAGNETS)	(36 DOILIES DOLLY GIRL)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	i
180	(CHILDRENS CUTLERY CIRCUS PARADE)	(CHILDRENS CUTLERY DOLLY GIRL)	0.055556	0.055556	0.055556	1.0	18.0	0.052469	i



In [ ]: