Market Basket Analysis

In [1]:

```
import numpy as np
import pandas as pd
import mlxtend
from mlxtend.frequent_patterns import apriori, fpgrowth
from mlxtend.frequent_patterns import association_rules
# import matplotlib.pyplot as plt
# import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
# sns.set_style('whitegrid')
```

In [2]:

```
df = pd.read_excel('http://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%2
df.head()
```

Out[2]:

	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

Data Cleaning

In [ ]:

```
# sns.heatmap(df.isnull(),yticklabels = False, cbar = False, cmap = 'viridis')
```

In [ ]:

```
# (df['CustomerID'].isna().sum()/len(df['CustomerID']))*100
```

24.93% of the CustomerID values are null.

In [3]:

```
df['Description'] = df['Description'].str.strip() #removes spaces from beginning and end
#df.dropna(axis=0, subset=['InvoiceNo'], inplace=True) #removes duplicate invoice
df['InvoiceNo'] = df['InvoiceNo'].astype('str') #converting invoice number to be string
df = df[~df['InvoiceNo'].str.contains('C')] #remove the credit transactions
df.head()
```

Out[3]:

	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

In [4]:

```
df['Country'].value_counts()
```

Out[4]:

```
United Kingdom          487622
Germany                   9042
France                    8408
EIRE                      7894
Spain                     2485
Netherlands               2363
Belgium                   2031
Switzerland               1967
Portugal                  1501
Australia                 1185
Norway                    1072
Italy                      758
Channel Islands            748
Finland                    685
Cyprus                     614
Sweden                     451
Unspecified                446
Austria                    398
Denmark                    380
Poland                     330
Japan                      321
Israel                     295
Hong Kong                  284
Singapore                  222
Iceland                    182
USA                        179
Canada                     151
Greece                     145
Malta                      112
United Arab Emirates        68
European Community          60
RSA                         58
Lebanon                     45
Lithuania                   35
Brazil                      32
Czech Republic              25
Bahrain                     18
Saudi Arabia                 9
Name: Country, dtype: int64
```

In [ ]:

```
# mybasket = (df[df['Country'] =="Germany"]
#           .groupby(['InvoiceNo', 'Description'])['Quantity']
#           .sum().unstack().reset_index().fillna(0)
#           .set_index('InvoiceNo'))
# mybasket.head()
```

In [5]:

```
mybasket = (df[df['Country'] =="United Kingdom"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
mybasket.head()
```

Out[5]:

Description	*Boombox Ipod Classic	*USB Office Mirror Ball	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 DAISY PEGS IN WOOD BOX	12 EGG HOUSE PAINTED WOOD	12 HANGING EGGS HAND PAINTED	12 IVOF ROS PE PLAC SETTINC
InvoiceNo								
536365	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
536366	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
536367	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
536368	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
536369	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

5 rows × 4175 columns

In [7]:

```
#converting all positive vaues to 1 and everything else to 0
def my_encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

my_basket_sets = mybasket.applymap(my_encode_units)
my_basket_sets.drop('POSTAGE', inplace=True, axis=1) #Remove "postage" as an item
```

Training Model

Apriori

In [12]:

```
#Generatig frequent itemsets
my_frequent_itemsets = apriori(my_basket_sets, min_support=0.03, use_colnames=True)
```

In [13]:

```
my_frequent_itemsets.sort_values('support', ascending = False)
```

Out[13]:

	support	itemsets
118	0.116034	(WHITE HANGING HEART T-LIGHT HOLDER)
52	0.103820	(JUMBO BAG RED RETROSPOT)
95	0.090266	(REGENCY CAKESTAND 3 TIER)
84	0.085391	(PARTY BUNTING)
69	0.074570	(LUNCH BAG RED RETROSPOT)
...	...	...
126	0.030535	(JUMBO BAG RED RETROSPOT, JUMBO BAG BAROQUE B...
96	0.030428	(RETROSPOT HEART HOT WATER BOTTLE)
18	0.030214	(DOORMAT HEARTS)
123	0.030160	(ALARM CLOCK BAKELIKE RED, ALARM CLOCK BAKELIK...
20	0.030107	(DOORMAT NEW ENGLAND)

131 rows × 2 columns

In [14]:

```
#generating rules
my_rules_apriori = association_rules(my_frequent_itemsets, metric="lift", min_threshold=1)
```

In [15]:

```
#viewing top 100 rules
my_rules_apriori.head(100)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	convic
0	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.049821	0.046928	0.030160	0.605376	12.900183	0.027822	2.415
1	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.046928	0.049821	0.030160	0.642694	12.900183	0.027822	2.659
2	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.050035	0.037660	0.030910	0.617773	16.403939	0.029026	2.517

Recommendations

In [16]:

```
my_basket_sets['ROUND SNACK BOXES SET OF4 WOODLAND'].sum()
```

Out[16]:

```
428
```

In [17]:

```
my_basket_sets['SPACEBOY LUNCH BOX'].sum()
```

Out[17]:

```
701
```

In [18]:

```
#Filtering rules based on condition
my_rules_apriori[ (my_rules_apriori['lift'] >= 3) &
       (my_rules_apriori['confidence'] >= 0.3) ]
```

4	TEACUP AND SAUCER)	TEACUP AND SAUCER)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.050035	0.051267	0.037553	0.750535	14.639752	0.034988	3.803
7	(JUMBO BAG BAROQUE BLACK WHITE)	(JUMBO BAG RED RETROSPOT)	0.048749	0.103820	0.030535	0.626374	6.033290	0.025474	2.398
8	(JUMBO BAG RED RETROSPOT)	(JUMBO BAG PINK POLKADOT)	0.103820	0.062088	0.042053	0.405057	6.523895	0.035607	1.576
9	(JUMBO BAG PINK	(JUMBO BAG RED	0.062088	0.103820	0.042053	0.677308	6.523895	0.035607	2.777

FPG

In [19]:

```
frequent_itemsets = fpgrowth(my_basket_sets, min_support=0.03, use_colnames=True)
```

In [20]:

```
frequent_itemsets.sort_values('support', ascending = False)
```

Out[20]:

	support	itemsets
0	0.116034	(WHITE HANGING HEART T-LIGHT HOLDER)
35	0.103820	(JUMBO BAG RED RETROSPOT)
73	0.090266	(REGENCY CAKESTAND 3 TIER)
96	0.085391	(PARTY BUNTING)
11	0.074570	(LUNCH BAG RED RETROSPOT)
...	...	...
125	0.030535	(JUMBO BAG RED RETROSPOT, JUMBO BAG BAROQUE B...
41	0.030428	(RETROSPOT HEART HOT WATER BOTTLE)
63	0.030214	(DOORMAT HEARTS)
127	0.030160	(ALARM CLOCK BAKELIKE RED, ALARM CLOCK BAKELIK...
4	0.030107	(DOORMAT NEW ENGLAND)

131 rows × 2 columns

In [23]:

```
my_rules_fpg = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
```

In [24]:

```
#Filtering rules based on condition
my_rules_fpg[ (my_rules_fpg['lift'] >= 3) &
       (my_rules_fpg['confidence'] >= 0.3) ]
```

Out[24]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
0	(JUMBO BAG RED RETROSPOT)	(JUMBO BAG PINK POLKADOT)	0.103820	0.062088	0.042053	0.405057	6.523895	0.0
1	(JUMBO BAG PINK POLKADOT)	(JUMBO BAG RED RETROSPOT)	0.062088	0.103820	0.042053	0.677308	6.523895	0.0
2	(JUMBO STORAGE BAG SUKI)	(JUMBO BAG RED RETROSPOT)	0.060535	0.103820	0.037392	0.617699	5.949737	0.0
3	(JUMBO BAG RED RETROSPOT)	(JUMBO STORAGE BAG SUKI)	0.103820	0.060535	0.037392	0.360165	5.949737	0.0
5	(JUMBO BAG BAROQUE BLACK WHITE)	(JUMBO BAG RED RETROSPOT)	0.048749	0.103820	0.030535	0.626374	6.033290	0.0
6	(JUMBO BAG RED RETROSPOT)	(JUMBO SHOPPER VINTAGE RED PAISLEY)	0.103820	0.060695	0.035196	0.339009	5.585425	0.0
7	(JUMBO SHOPPER VINTAGE RED PAISLEY)	(JUMBO BAG RED RETROSPOT)	0.060695	0.103820	0.035196	0.579876	5.585425	0.0
8	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.049821	0.046928	0.030160	0.605376	12.900183	0.0
9	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.046928	0.049821	0.030160	0.642694	12.900183	0.0
10	(LUNCH BAG RED RETROSPOT)	(LUNCH BAG BLACK SKULL.)	0.074570	0.065142	0.032517	0.436063	6.694072	0.0
11	(LUNCH BAG BLACK SKULL.)	(LUNCH BAG RED RETROSPOT)	0.065142	0.074570	0.032517	0.499178	6.694072	0.0
12	(ROSES REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.051267	0.050035	0.037553	0.732497	14.639752	0.0
13	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.050035	0.051267	0.037553	0.750535	14.639752	0.0

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lev
14	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.050035	0.037660	0.030910	0.617773	16.403939	0.0:
15	(PINK REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.037660	0.050035	0.030910	0.820768	16.403939	0.0:

So we can see that both apriori and fp growth analysis have same results, but in pracitce, fp growth is a faster algorithm to work on

Possible recommendations to customers when buying certain products (antecedents)

In [26]:

```
#Filtering rules based on condition
my_rules_fpg[ (my_rules_fpg['lift'] >= 5) &
       (my_rules_fpg['confidence'] >= 0.5) ]
```

Out[26]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lever
1	(JUMBO BAG PINK POLKADOT)	(JUMBO BAG RED RETROSPOT)	0.062088	0.103820	0.042053	0.677308	6.523895	0.035
2	(JUMBO STORAGE BAG SUKI)	(JUMBO BAG RED RETROSPOT)	0.060535	0.103820	0.037392	0.617699	5.949737	0.031
5	(JUMBO BAG BAROQUE BLACK WHITE)	(JUMBO BAG RED RETROSPOT)	0.048749	0.103820	0.030535	0.626374	6.033290	0.025
7	(JUMBO SHOPPER VINTAGE RED PAISLEY)	(JUMBO BAG RED RETROSPOT)	0.060695	0.103820	0.035196	0.579876	5.585425	0.028
8	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.049821	0.046928	0.030160	0.605376	12.900183	0.027
9	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.046928	0.049821	0.030160	0.642694	12.900183	0.027
12	(ROSES REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.051267	0.050035	0.037553	0.732497	14.639752	0.034
13	(GREEN REGENCY TEACUP AND SAUCER)	(ROSES REGENCY TEACUP AND SAUCER)	0.050035	0.051267	0.037553	0.750535	14.639752	0.034
14	(GREEN REGENCY TEACUP AND SAUCER)	(PINK REGENCY TEACUP AND SAUCER)	0.050035	0.037660	0.030910	0.617773	16.403939	0.029
15	(PINK REGENCY TEACUP AND SAUCER)	(GREEN REGENCY TEACUP AND SAUCER)	0.037660	0.050035	0.030910	0.820768	16.403939	0.029

These consquents can be recommended to customers once they buy the products in antecedents.