CDS503 - Machine Learning

Market Basket Analysis (Apriori)

A Simple Classification (Market Basket Analysis) using Python

We are going to start demonstrating Python with a simple Market Basket Analysis. In this lab, we are going to explore the Online Retail data set. This work is based on an article written by Chris Moffitt in [https://pbpython.com/market-basket-analysis.html (https://pbpython.com/market-basket-analysis.html)](https://pbpython.com/market-basket-analysis.html)

In this Lab we are going to use the CRISP-DM model.

```
In [22]: rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
         rules
Out[22]:
                       antecedants                        consequents   support  confidence      lift
0  (PLASTERS IN TIN WOODLAND ANIMALS)     (PLASTERS IN TIN CIRCUS PARADE)  0.170918    0.597015  3.545907
1     (PLASTERS IN TIN CIRCUS PARADE)  (PLASTERS IN TIN WOODLAND ANIMALS)  0.168367    0.606061  3.545907
2     (PLASTERS IN TIN CIRCUS PARADE)        (PLASTERS IN TIN SPACEBOY)  0.168367    0.530303  3.849607
3          (PLASTERS IN TIN SPACEBOY)     (PLASTERS IN TIN CIRCUS PARADE)  0.137755    0.648148  3.849607
4  (PLASTERS IN TIN WOODLAND ANIMALS)        (PLASTERS IN TIN SPACEBOY)  0.170918    0.611940  4.442233
5          (PLASTERS IN TIN SPACEBOY)  (PLASTERS IN TIN WOODLAND ANIMALS)  0.137755    0.759259  4.442233
```

Association rules are normally written like this: {Diapers} -> {Beer} which means that there is a strong relationship between customers that purchased diapers and also purchased beer in the same transaction.

In the above example, the {Diaper} is the **antecedent** and the {Beer} is the **consequent**. Both antecedents and consequents can have multiple items. In other words, {Diaper, Gum} -> {Beer, Chips} is a valid rule.

Support is the relative frequency that the rules show up. In many instances, you may want to look for high support in order to make sure it is a useful relationship. However, there may be instances where a low support is useful if you are trying to find “hidden” relationships.

Confidence is a measure of the reliability of the rule. A confidence of .5 in the above example would mean that in 50% of the cases where Diaper and Gum were purchased, the purchase also included Beer and Chips. For product recommendation, a 50% confidence may be perfectly acceptable but in a medical situation, this level may not be high enough.

Lift is the ratio of the observed support to that expected if the two rules were independent (see wikipedia). The basic rule of thumb is that a lift value close to 1 means the rules were completely independent. Lift values > 1 are generally more “interesting” and could be indicative of a useful rule pattern.

Step1: Business Understanding

The specific data for this article comes from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Online+Retail (http://archive.ics.uci.edu/ml/datasets/Online+Retail)) and represents transactional data from a UK retailer from 2010-2011. This mostly represents sales to wholesalers so it is slightly different from consumer purchase patterns but is still a useful case study. You should make a folder in the lab or your own computer to save data and your own work.

This example will use MLxtend library. MLxtend can be installed using pip, so make sure that is done before trying to execute any of the code below. Once it is installed, the code below shows how to get it up and running. I have made the notebook available so feel free to follow along with the examples below.

Use the command given to install MLxtend package using Anaconda Prompt: **pip install mlxtend**

In [1]:
```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

df = pd.read_csv("input/Online Retail.csv", encoding= 'unicode_escape')
df.head()
```

Out[1]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

In [2]:
```
df.shape
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Out[2]:
```
(541909, 8)
```

In [3]:
```
#describe the data
df.describe()
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
C:\Users\7327152\Anaconda3\lib\site-packages\pandas\core\nanops.py:1558: DeprecationWarning: the `interpolation=` argument to percentile was
renamed to `method=`, which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encouraged to review the method they. (Deprecated NumPy 1.22)
  return np.percentile(values, q, axis=axis, interpolation=interpolation)
C:\Users\7327152\Anaconda3\lib\site-packages\pandas\core\nanops.py:1558: DeprecationWarning: the `interpolation=` argument to percentile was
renamed to `method=`, which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encouraged to review the method they. (Deprecated NumPy 1.22)
  return np.percentile(values, q, axis=axis, interpolation=interpolation)
C:\Users\7327152\Anaconda3\lib\site-packages\pandas\core\nanops.py:1498: DeprecationWarning: the `interpolation=` argument to percentile was
renamed to `method=`, which has additional options.
Users of the modes 'nearest', 'lower', 'higher', or 'midpoint' are encouraged to review the method they. (Deprecated NumPy 1.22)
  return np.percentile(values, q, interpolation=interpolation)
```

Out[3]:

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

There is a little cleanup, we need to do. First, some of the descriptions have spaces that need to be removed. We’ll also drop the rows that don’t have invoice numbers and remove the credit transactions (those with invoice numbers containing C).

In [4]:
```
#remove space
df['Description'] = df['Description'].str.strip()
#drop rows that contain 'C' in invoice no
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df = df[~df['InvoiceNo'].str.contains('C')]
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

After the cleanup, we need to consolidate the items into 1 transaction per row with each product 1 hot encoded. For the sake of keeping the data set small, we only looking at sales for France. However, in additional code below, we will compare these results to sales from Germany. Further country comparisons would be interesting to investigate.

In [5]:
```
basket = (df[df['Country'] =="France"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))

print(basket.head(5))
```

```
Description  10 COLOUR SPACEBOY PEN  12 COLOURED PARTY BALLOONS  \
InvoiceNo
536370                          0.0                        0.0
536852                          0.0                        0.0
536974                          0.0                        0.0
537065                          0.0                        0.0
537463                          0.0                        0.0

Description  12 EGG HOUSE PAINTED WOOD  12 MESSAGE CARDS WITH ENVELOPES  \
InvoiceNo
536370                             0.0                              0.0
536852                             0.0                              0.0
536974                             0.0                              0.0
537065                             0.0                              0.0
537463                             0.0                              0.0

Description  12 PENCIL SMALL TUBE WOODLAND  \
InvoiceNo
536370                                 0.0
536852                                 0.0
536974                                 0.0
537065                                 0.0
537463                                 0.0

Description  12 PENCILS SMALL TUBE RED RETROSPOT  12 PENCILS SMALL TUBE SKULL  \
InvoiceNo
536370                                       0.0                          0.0
536852                                       0.0                          0.0
536974                                       0.0                          0.0
537065                                       0.0                          0.0
537463                                       0.0                          0.0

Description  12 PENCILS TALL TUBE POSY  12 PENCILS TALL TUBE RED RETROSPOT  \
InvoiceNo
536370                             0.0                                 0.0
536852                             0.0                                 0.0
536974                             0.0                                 0.0
537065                             0.0                                 0.0
537463                             0.0                                 0.0

Description  12 PENCILS TALL TUBE WOODLAND  ...  WRAP VINTAGE PETALS  DESIGN  \
InvoiceNo                                   ...
536370                                 0.0  ...                         0.0
536852                                 0.0  ...                         0.0
536974                                 0.0  ...                         0.0
537065                                 0.0  ...                         0.0
537463                                 0.0  ...                         0.0
```

```
Description  YELLOW COAT RACK PARIS FASHION  YELLOW GIANT GARDEN THERMOMETER  \
InvoiceNo
536370                                  0.0                              0.0
536852                                  0.0                              0.0
536974                                  0.0                              0.0
537065                                  0.0                              0.0
537463                                  0.0                              0.0

Description  YELLOW SHARK HELICOPTER  ZINC  STAR T-LIGHT HOLDER  \
InvoiceNo
536370                           0.0                         0.0
536852                           0.0                         0.0
536974                           0.0                         0.0
537065                           0.0                         0.0
537463                           0.0                         0.0

Description  ZINC FOLKART SLEIGH BELLS  ZINC HERB GARDEN CONTAINER  \
InvoiceNo
536370                             0.0                         0.0
536852                             0.0                         0.0
536974                             0.0                         0.0
537065                             0.0                         0.0
537463                             0.0                         0.0

Description  ZINC METAL HEART DECORATION  ZINC T-LIGHT HOLDER STAR LARGE  \
InvoiceNo
536370                               0.0                             0.0
536852                               0.0                             0.0
536974                               0.0                             0.0
537065                               0.0                             0.0
537463                               0.0                             0.0

Description  ZINC T-LIGHT HOLDER STARS SMALL
InvoiceNo
536370                                   0.0
536852                                   0.0
536974                                   0.0
537065                                   0.0
537463                                   0.0

[5 rows x 1563 columns]

C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

There are a lot of zeros in the data but we also need to make sure any positive values are converted to a 1 and anything less the 0 is set to 0. This step will complete the one hot encoding of the data and remove the postage column (since that charge is not one we wish to explore):

In [6]:
```
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)
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

In [7]:
```
print(basket_sets.head(5))
```

```
Description  10 COLOUR SPACEBOY PEN  12 COLOURED PARTY BALLOONS  \
InvoiceNo
536370                           0                          0
536852                           0                          0
536974                           0                          0
537065                           0                          0
537463                           0                          0

Description  12 EGG HOUSE PAINTED WOOD  12 MESSAGE CARDS WITH ENVELOPES  \
InvoiceNo
536370                              0                                0
536852                              0                                0
536974                              0                                0
537065                              0                                0
537463                              0                                0

Description  12 PENCIL SMALL TUBE WOODLAND  \
InvoiceNo
536370                                  0
536852                                  0
536974                                  0
537065                                  0
537463                                  0

Description  12 PENCILS SMALL TUBE RED RETROSPOT  12 PENCILS SMALL TUBE SKULL  \
InvoiceNo
536370                                        0                            0
536852                                        0                            0
536974                                        0                            0
537065                                        0                            0
537463                                        0                            0

Description  12 PENCILS TALL TUBE POSY  12 PENCILS TALL TUBE RED RETROSPOT  \
InvoiceNo
536370                              0                                    0
536852                              0                                    0
536974                              0                                    0
537065                              0                                    0
537463                              0                                    0

Description  12 PENCILS TALL TUBE WOODLAND  ...  WRAP VINTAGE PETALS  DESIGN  \
InvoiceNo                                   ...
536370                                  0  ...                           0
536852                                  0  ...                           0
536974                                  0  ...                           0
537065                                  0  ...                           0
537463                                  0  ...                           0
```

```
Description  YELLOW COAT RACK PARIS FASHION  YELLOW GIANT GARDEN THERMOMETER  \
InvoiceNo
536370                                    0                                0
536852                                    0                                0
536974                                    0                                0
537065                                    0                                0
537463                                    0                                0

Description  YELLOW SHARK HELICOPTER  ZINC  STAR T-LIGHT HOLDER  \
InvoiceNo
536370                             0                           0
536852                             0                           0
536974                             0                           0
537065                             0                           0
537463                             0                           0

Description  ZINC FOLKART SLEIGH BELLS  ZINC HERB GARDEN CONTAINER  \
InvoiceNo
536370                               0                           0
536852                               0                           0
536974                               0                           0
537065                               0                           0
537463                               0                           0

Description  ZINC METAL HEART DECORATION  ZINC T-LIGHT HOLDER STAR LARGE  \
InvoiceNo
536370                                 0                               0
536852                                 0                               0
536974                                 0                               0
537065                                 0                               0
537463                                 0                               0

Description  ZINC T-LIGHT HOLDER STARS SMALL
InvoiceNo
536370                                     0
536852                                     0
536974                                     0
537065                                     0
537463                                     0

[5 rows x 1562 columns]

C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Now that the data is structured properly, we can generate frequent item sets that have a support of at least 7% (this number was chosen so that we could get enough useful examples):

In [8]:
```
%%time

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

```
Wall time: 64 ms

C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
C:\Users\7327152\AppData\Roaming\Python\Python38\site-packages\mlxtend\frequent_patterns\fpcommon.py:111: DeprecationWarning: DataFrames wit
h non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with
bool type
  warnings.warn(
```

The final step is to generate the rules with their corresponding support, confidence and lift:

In [9]:
```python
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
#rules.head()
print(rules)
```

```
                                     antecedents  \
0                        (ALARM CLOCK BAKELIKE PINK)
1                       (ALARM CLOCK BAKELIKE GREEN)
2                         (ALARM CLOCK BAKELIKE RED)
3                       (ALARM CLOCK BAKELIKE GREEN)
4                        (ALARM CLOCK BAKELIKE PINK)
5                         (ALARM CLOCK BAKELIKE RED)
6                              (SPACEBOY LUNCH BOX)
7                             (DOLLY GIRL LUNCH BOX)
8                         (PLASTERS IN TIN SPACEBOY)
9                    (PLASTERS IN TIN CIRCUS PARADE)
10                (PLASTERS IN TIN WOODLAND ANIMALS)
11                   (PLASTERS IN TIN CIRCUS PARADE)
12                        (PLASTERS IN TIN SPACEBOY)
13                (PLASTERS IN TIN WOODLAND ANIMALS)
14              (SET/20 RED RETROSPOT PAPER NAPKINS)
15                     (SET/6 RED SPOTTY PAPER CUPS)
16                   (SET/6 RED SPOTTY PAPER PLATES)
17              (SET/20 RED RETROSPOT PAPER NAPKINS)
18                   (SET/6 RED SPOTTY PAPER PLATES)
19                     (SET/6 RED SPOTTY PAPER CUPS)
20  (SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...
21  (SET/6 RED SPOTTY PAPER PLATES, SET/6 RED SPOT...
22  (SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...
23                   (SET/6 RED SPOTTY PAPER PLATES)
24              (SET/20 RED RETROSPOT PAPER NAPKINS)
25                     (SET/6 RED SPOTTY PAPER CUPS)

                                     consequents  antecedent support  \
0                       (ALARM CLOCK BAKELIKE GREEN)            0.102041
1                        (ALARM CLOCK BAKELIKE PINK)            0.096939
2                       (ALARM CLOCK BAKELIKE GREEN)            0.094388
3                         (ALARM CLOCK BAKELIKE RED)            0.096939
4                         (ALARM CLOCK BAKELIKE RED)            0.102041
5                        (ALARM CLOCK BAKELIKE PINK)            0.094388
6                             (DOLLY GIRL LUNCH BOX)            0.125000
7                              (SPACEBOY LUNCH BOX)            0.099490
8                    (PLASTERS IN TIN CIRCUS PARADE)            0.137755
9                         (PLASTERS IN TIN SPACEBOY)            0.168367
10                   (PLASTERS IN TIN CIRCUS PARADE)            0.170918
11                (PLASTERS IN TIN WOODLAND ANIMALS)            0.168367
12                (PLASTERS IN TIN WOODLAND ANIMALS)            0.137755
13                        (PLASTERS IN TIN SPACEBOY)            0.170918
14                     (SET/6 RED SPOTTY PAPER CUPS)            0.132653
15              (SET/20 RED RETROSPOT PAPER NAPKINS)            0.137755
16              (SET/20 RED RETROSPOT PAPER NAPKINS)            0.127551
17                   (SET/6 RED SPOTTY PAPER PLATES)            0.132653
18                     (SET/6 RED SPOTTY PAPER CUPS)            0.127551
```

```
19                    (SET/6 RED SPOTTY PAPER PLATES)           0.137755
20                      (SET/6 RED SPOTTY PAPER CUPS)           0.102041
21               (SET/20 RED RETROSPOT PAPER NAPKINS)           0.122449
22                    (SET/6 RED SPOTTY PAPER PLATES)           0.102041
23  (SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...           0.127551
24  (SET/6 RED SPOTTY PAPER PLATES, SET/6 RED SPOT...           0.132653
25  (SET/6 RED SPOTTY PAPER PLATES, SET/20 RED RET...           0.137755

    consequent support   support  confidence      lift  leverage  conviction
0             0.096939  0.073980    0.725000  7.478947  0.064088    3.283859
1             0.102041  0.073980    0.763158  7.478947  0.064088    3.791383
2             0.096939  0.079082    0.837838  8.642959  0.069932    5.568878
3             0.094388  0.079082    0.815789  8.642959  0.069932    4.916181
4             0.094388  0.073980    0.725000  7.681081  0.064348    3.293135
5             0.102041  0.073980    0.783784  7.681081  0.064348    4.153061
6             0.099490  0.071429    0.571429  5.743590  0.058992    2.101190
7             0.125000  0.071429    0.717949  5.743590  0.058992    3.102273
8             0.168367  0.089286    0.648148  3.849607  0.066092    2.363588
9             0.137755  0.089286    0.530303  3.849607  0.066092    1.835747
10            0.168367  0.102041    0.597015  3.545907  0.073264    2.063681
11            0.170918  0.102041    0.606061  3.545907  0.073264    2.104592
12            0.170918  0.104592    0.759259  4.442233  0.081047    3.443878
13            0.137755  0.104592    0.611940  4.442233  0.081047    2.221939
14            0.137755  0.102041    0.769231  5.584046  0.083767    3.736395
15            0.132653  0.102041    0.740741  5.584046  0.083767    3.345481
16            0.132653  0.102041    0.800000  6.030769  0.085121    4.336735
17            0.127551  0.102041    0.769231  6.030769  0.085121    3.780612
18            0.137755  0.122449    0.960000  6.968889  0.104878   21.556122
19            0.127551  0.122449    0.888889  6.968889  0.104878    7.852041
20            0.137755  0.099490    0.975000  7.077778  0.085433   34.489796
21            0.132653  0.099490    0.812500  6.125000  0.083247    4.625850
22            0.127551  0.099490    0.975000  7.644000  0.086474   34.897959
23            0.102041  0.099490    0.780000  7.644000  0.086474    4.081633
24            0.122449  0.099490    0.750000  6.125000  0.083247    3.510204
25            0.102041  0.099490    0.722222  7.077778  0.085433    3.232653

C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

That's all there is to it! Build the frequent items using apriori then build the rules with association_rules .

Now, the tricky part is figuring out what this tells us. For instance, we can see that there are quite a few rules with a high lift value which means that it occurs more frequently than would be expected given the number of transaction and product combinations. We can also see several where the confidence is high as well. This part of the analysis is where the domain knowledge will come in handy. Since I do not have that, I'll just look for a couple of illustrative examples.

We can filter the dataframe using standard pandas code. In this case, look for a large lift (6) and high confidence (.8):

In [10]:
```
rules[ (rules['lift'] >= 6) &
       (rules['confidence'] >= 0.8) ]
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Out[10]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
2	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.094388	0.096939	0.079082	0.837838	8.642959	0.069932	5.568878
3	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.096939	0.094388	0.079082	0.815789	8.642959	0.069932	4.916181
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/20 RED RET...	(SET/6 RED SPOTTY PAPER CUPS)	0.102041	0.137755	0.099490	0.975000	7.077778	0.085433	34.489796
21	(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
22	(SET/20 RED RETROSPOT PAPER NAPKINS, SET/6 RED...	(SET/6 RED SPOTTY PAPER PLATES)	0.102041	0.127551	0.099490	0.975000	7.644000	0.086474	34.897959

In looking at the rules, it seems that the green and red alarm clocks are purchased together and the red paper cups, napkins and plates are purchased together in a manner that is higher than the overall probability would suggest.

At this point, you may want to look at how much opportunity there is to use the popularity of one product to drive sales of another. For instance, we can see that we sell 340 Green Alarm clocks but only 316 Red Alarm Clocks so maybe we can drive more Red Alarm Clock sales through recommendations?

In [11]:
```
basket['ALARM CLOCK BAKELIKE GREEN'].sum()
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Out[11]:
```
340.0
```

In [12]:
```
basket['ALARM CLOCK BAKELIKE RED'].sum()
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
```

Out[12]:
```
316.0
```

What is also interesting is to see how the combinations vary by country of purchase. Let's check out what some popular combinations might be in Germany:

In [13]:
```
basket2 = (df[df['Country'] =="Germany"]
          .groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))

basket_sets2 = basket2.applymap(encode_units)
basket_sets2.drop('POSTAGE', inplace=True, axis=1)
frequent_itemsets2 = apriori(basket_sets2, min_support=0.05, use_colnames=True)
rules2 = association_rules(frequent_itemsets2, metric="lift", min_threshold=1)

rules2[ (rules2['lift'] >= 4) &
        (rules2['confidence'] >= 0.5)]
```

```
C:\Users\7327152\Anaconda3\lib\site-packages\ipykernel\ipkernel.py:287: DeprecationWarning: `should_run_async` will not call `transform_cell
` automatically in the future. Please pass the result to `transformed_cell` argument and any exception that happen during thetransform in `p
reprocessing_exc_tuple` in IPython 7.17 and above.
  and should_run_async(code)
C:\Users\7327152\AppData\Roaming\Python\Python38\site-packages\mlxtend\frequent_patterns\fpcommon.py:111: DeprecationWarning: DataFrames wit
h non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with
bool type
  warnings.warn(
```

Out[13]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.115974	0.137856	0.067834	0.584906	4.242887	0.051846	2.076984
6	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.107221	0.137856	0.061269	0.571429	4.145125	0.046488	2.011670
10	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.070022	0.126915	0.059081	0.843750	6.648168	0.050194	5.587746

It seems that in addition to David Hasselhoff, Germans love Plasters in Tin Spaceboy and Woodland Animals.

In all seriousness, an analyst that has familiarity with the data would probably have a dozen different questions that this type of analysis could drive. I did not replicate this analysis for additional countries or customer combos but the overall process would be relatively simple given the basic pandas code shown above.

Conclusion

The really nice aspect of association analysis is that it is easy to run and relatively easy to interpret. If you did not have access to MLxtend and this association analysis, it would be exceedingly difficult to find these patterns using basic Excel analysis. With python and MLxtend, the analysis process is relatively straightforward and since you are in python, you have access to all the additional visualization techniques and data analysis tools in the python ecosystem.

Finally, I encourage you to check out the rest of the MLxtend library. If you are doing any work in sci-kit learn it is helpful to be familiar with MLxtend and how it could augment some of the existing tools in your data science toolkit.

In [ ]: