Análise de correlações

In [1]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import warnings

warnings.filterwarnings('ignore')
pd.options.display.float_format = "{:.2f}".format
pd.options.display.max_rows = 10
```

In [2]:
```python
df = pd.read_excel("Online Retail.xlsx")
```

Conhecendo o dataset

Colunas

In [3]:
```python
df.columns
```

Out[3]:
```
Index(['InvoiceNo', 'StockCode', 'Description', 'Quantity', 'InvoiceDate',
       'UnitPrice', 'CustomerID', 'Country'],
      dtype='object')
```

Tipos de dados das colunas

In [4]:
```python
df.dtypes
```

Out[4]:
```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
dtype: object
```

Dataset

In [5]:
```python
df
```

Out[5]:

	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	17850.00	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.00	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.00	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.00	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.00	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.00	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.00	France

541909 rows × 8 columns

Limpeza dos dados

Removendo linhas vazias

In [6]:
```python
df.dropna(axis='index', how='all', inplace=True)
```

Removendo linhas onde a coluna 'InvoiceNo' esteja vazia

In [7]:
```python
df.dropna(axis='index', subset=['InvoiceNo'], inplace=True)
```

Exibindo o resultado

In [8]:
```python
df
```

Out[8]:

	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	17850.00	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.00	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.00	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.00	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.00	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.00	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.00	France

541909 rows × 8 columns

Convertendo as colunas que contenham texto para string

In [9]:
```python
df['InvoiceNo'] = df['InvoiceNo'].astype('str')
df['StockCode'] = df['StockCode'].astype('str')
df['Description'] = df['Description'].astype('str')
df['Country'] = df['Country'].astype('str')
```

Removendo espaços errados (começo ou final) das strings

In [10]:
```python
df['InvoiceNo'] = df['InvoiceNo'].str.strip()
df['StockCode'] = df['StockCode'].str.strip()
df['Description'] = df['Description'].str.strip()
df['Country'] = df['Country'].str.strip()
```

Filtrando resultados com dados de 'Quantity' e 'UnitPrice' coerentes com a realidade

In [11]:
```python
df = df[(df['Quantity'] >= 0) & (df['UnitPrice'] >= 0)]
```

In [12]:
```python
df
```

Out[12]:

	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	17850.00	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.00	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.00	United Kingdom
...	...	...	...	...	...	...	...	...
541904	581587	22613	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.00	France
541905	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.00	France
541906	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.00	France
541907	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.00	France
541908	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.00	France

531283 rows × 8 columns

Escolhendo país para análise

Checando a quantidade de 'InvoiceNo' presente em cada país

In [13]:
```python
df.groupby(['Country'])['InvoiceNo'].count()
```

Out[13]:
```
Country
Australia                1185
Austria                   398
Bahrain                    18
Belgium                  2031
Brazil                     32
                        ...
Switzerland              1967
USA                       179
United Arab Emirates       68
United Kingdom         486284
Unspecified               446
Name: InvoiceNo, Length: 38, dtype: int64
```

Escolhendo a 'Germany' por possuir uma quantidade significativa de dados

In [14]:
```python
pais = 'Germany'
```

In [15]:
```python
df_germany = df[df['Country'] == pais]
```

Preparação dos dados para construir a correlação

Agrupando por 'InvoiceNo' (id da compra/transação) e 'Description' para separar o dataset em transações detalhadas (carrinhos de compras)

In [16]:
```python
df_grouped_germany = df_germany.groupby(['InvoiceNo', 'Description'])['Quantity'].sum()
```

Resultado do agrupamento

In [17]:
```python
df_grouped_germany
```

Out[17]:
```
InvoiceNo  Description
536527     3 HOOK HANGER MAGIC GARDEN             12
           5 HOOK HANGER MAGIC TOADSTOOL          12
           5 HOOK HANGER RED MAGIC TOADSTOOL      12
           ASSORTED COLOUR LIZARD SUCTION HOOK    24
           CHILDREN'S CIRCUS PARADE MUG           12
                                                  ..
581578     SPOTTY BUNTING                          9
           VINTAGE DONKEY TAIL GAME                6
           WRAP ALPHABET POSTER                   25
           WRAP CIRCUS PARADE                     25
           WRAP RED APPLES                        25
Name: Quantity, Length: 9015, dtype: int64
```

Usando função 'pandas.unstack' para selecionar uma coluna e transformar todas as suas linhas em coluna.

In [18]:
```python
carrinho_compras_germany = df_grouped_germany.unstack(level=1, fill_value=0)
```

Resultado da transformação, observe que agora todas as linhas são equivalentes as compras (carrinhos) e que cada coluna é um tipo de produto e o valor a quantidade comprada por aquele cliente.

In [19]:
```python
carrinho_compras_germany
```

Out[19]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 IVORY ROSE PEG PLACE SETTINGS	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	12 PENCILS TALL TUBE SKULLS	...	YULETIDE IMAGES GIFT WRAP SET	ZINC HEART T-LIGHT HOLDER	ZINC STAR T-LIGHT HOLDER	ZINC BOX SIGN HOME	ZINC FOLKART SLEIGH BELLS	ZINC HEART LATTICE T-LIGHT HOLDER
InvoiceNo																	
536527	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536840	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536861	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536967	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536983	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
581266	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581494	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581570	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581574	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581578	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

457 rows × 1695 columns

Checando se a coluna 'InvoiceNo' é o índice do dataframe

In [20]:
```python
carrinho_compras_germany.index
```

Out[20]:
```
Index(['536527', '536840', '536861', '536967', '536983', '537197', '537198',
       '537201', '537212', '537250',
       ...
       '580648', '581000', '581179', '581183', '581184', '581266', '581494',
       '581570', '581574', '581578'],
      dtype='object', name='InvoiceNo', length=457)
```

Caso não fosse, bastaria utilizar 'set_index'

In [21]:
```python
# carrinho_compras_germany.reset_index().set_index('InvoiceNo')
```

In [22]:
```python
# carrinho_compras_germany.index
```

Convertendo as quantidades dos itens comprados para booleano, pois para o algoritmo basta estar ou não no carrinho.

In [23]:
```python
carrinho_compras_germany = carrinho_compras_germany.applymap(lambda item: 1 if item > 0 else 0)
```

In [24]:
```python
carrinho_compras_germany
```

Out[24]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 IVORY ROSE PEG PLACE SETTINGS	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	12 PENCILS TALL TUBE SKULLS	...	YULETIDE IMAGES GIFT WRAP SET	ZINC HEART T-LIGHT HOLDER	ZINC STAR T-LIGHT HOLDER	ZINC BOX SIGN HOME	ZINC FOLKART SLEIGH BELLS	ZINC HEART LATTICE T-LIGHT HOLDER
InvoiceNo																	
536527	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536840	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536861	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536967	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536983	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
581266	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581494	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581570	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581574	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581578	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

457 rows × 1695 columns

Removendo a coluna 'POSTAGE' que representa envio e não um produto

In [25]:
```python
carrinho_compras_germany.drop(labels=['POSTAGE'], inplace=True, axis='columns')
```

In [26]:
```python
carrinho_compras_germany
```

Out[26]:

Description	10 COLOUR SPACEBOY PEN	12 COLOURED PARTY BALLOONS	12 IVORY ROSE PEG PLACE SETTINGS	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	12 PENCILS TALL TUBE SKULLS	...	YULETIDE IMAGES GIFT WRAP SET	ZINC HEART T-LIGHT HOLDER	ZINC STAR T-LIGHT HOLDER	ZINC BOX SIGN HOME	ZINC FOLKART SLEIGH BELLS	ZINC HEART LATTICE T-LIGHT HOLDER
InvoiceNo																	
536527	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536840	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536861	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536967	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
536983	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
581266	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581494	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581570	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581574	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0
581578	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0

457 rows × 1694 columns

Algoritmo APRIORI

In [27]:
```python
itemsets_mais_frequentes_germany = apriori(carrinho_compras_germany, min_support=0.05, use_colnames=True)
```

Suportes para cada produto

In [28]:
```python
itemsets_mais_frequentes_germany
```

Out[28]:

	support	itemsets
0	0.10	(6 RIBBONS RUSTIC CHARM)
1	0.07	(ALARM CLOCK BAKELIKE PINK)
2	0.07	(CHARLOTTE BAG APPLES DESIGN)
3	0.05	(CHILDRENS CUTLERY DOLLY GIRL)
4	0.06	(COFFEE MUG APPLES DESIGN)
...	...	...
54	0.07	(PLASTERS IN TIN WOODLAND ANIMALS, ROUND SNACK...
55	0.06	(WOODLAND CHARLOTTE BAG, RED RETROSPOT CHARLOT...
56	0.13	(ROUND SNACK BOXES SET OF 4 FRUITS, ROUND SNAC...
57	0.07	(SPACEBOY LUNCH BOX, ROUND SNACK BOXES SET OF4...
58	0.06	(WOODLAND CHARLOTTE BAG, ROUND SNACK BOXES SET...

59 rows × 2 columns

Ordenando de forma decrescente para pegar os itens com os maiores suportes

In [29]:
```python
itemsets_mais_frequentes_germany.sort_values(by='support', axis=0, ascending=False, inplace=True)
```

Dataframe ordenado com base nos maiores suportes

In [30]:
```python
itemsets_mais_frequentes_germany
```

Out[30]:

	support	itemsets
35	0.25	(ROUND SNACK BOXES SET OF4 WOODLAND)
34	0.16	(ROUND SNACK BOXES SET OF 4 FRUITS)
24	0.14	(PLASTERS IN TIN WOODLAND ANIMALS)
31	0.14	(REGENCY CAKESTAND 3 TIER)
56	0.13	(ROUND SNACK BOXES SET OF 4 FRUITS, ROUND SNAC...
...	...	...
17	0.05	(MINT KITCHEN SCALES)
25	0.05	(RABBIT NIGHT LIGHT)
51	0.05	(ROUND SNACK BOXES SET OF 4 FRUITS, PLASTERS I...
36	0.05	(SET 2 PANTRY DESIGN TEA TOWELS)
3	0.05	(CHILDRENS CUTLERY DOLLY GIRL)

59 rows × 2 columns

Top 10 itens com maiores suportes individuais

In [31]:
```python
itemsets_mais_frequentes_germany.iloc[0:10]
```

Out[31]:

	support	itemsets
35	0.25	(ROUND SNACK BOXES SET OF4 WOODLAND)
34	0.16	(ROUND SNACK BOXES SET OF 4 FRUITS)
24	0.14	(PLASTERS IN TIN WOODLAND ANIMALS)
31	0.14	(REGENCY CAKESTAND 3 TIER)
56	0.13	(ROUND SNACK BOXES SET OF 4 FRUITS, ROUND SNAC...
48	0.13	(WOODLAND CHARLOTTE BAG)
21	0.12	(PLASTERS IN TIN CIRCUS PARADE)
22	0.11	(PLASTERS IN TIN SPACEBOY)
0	0.10	(6 RIBBONS RUSTIC CHARM)
44	0.10	(SPACEBOY LUNCH BOX)

Regras de associação

In [32]:
```python
regras_associacao_germany = association_rules(itemsets_mais_frequentes_germany, metric='support', min_threshold=0.05)
```

In [33]:
```python
regras_associacao_germany
```

Out[33]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.16	0.25	0.13	0.83	3.40	0.09	4.53
1	(ROUND SNACK BOXES SET OF4 WOODLAND)	(ROUND SNACK BOXES SET OF 4 FRUITS)	0.25	0.16	0.13	0.54	3.40	0.09	1.81
2	(PLASTERS IN TIN WOODLAND ANIMALS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.14	0.25	0.07	0.54	2.20	0.04	1.64
3	(ROUND SNACK BOXES SET OF4 WOODLAND)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.25	0.14	0.07	0.30	2.20	0.04	1.24
4	(SPACEBOY LUNCH BOX)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.10	0.25	0.07	0.68	2.78	0.04	2.37
...	...	...	...	...	...	...	...	...	...
13	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.07	0.13	0.06	0.84	6.65	0.05	5.59
14	(ROUND SNACK BOXES SET OF4 WOODLAND)	(PLASTERS IN TIN CIRCUS PARADE)	0.25	0.12	0.06	0.23	2.00	0.03	1.15
15	(PLASTERS IN TIN CIRCUS PARADE)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.12	0.25	0.06	0.49	2.00	0.03	1.48
16	(ROUND SNACK BOXES SET OF 4 FRUITS)	(PLASTERS IN TIN CIRCUS PARADE)	0.16	0.12	0.05	0.32	2.75	0.03	1.30
17	(PLASTERS IN TIN CIRCUS PARADE)	(ROUND SNACK BOXES SET OF 4 FRUITS)	0.12	0.16	0.05	0.43	2.75	0.03	1.49

18 rows × 9 columns

Top 10 - Regras de associação com maiores 'suportes'

In [34]:
```python
regras_associacao_germany.iloc[0:10]
```

Out[34]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.16	0.25	0.13	0.83	3.40	0.09	4.53
1	(ROUND SNACK BOXES SET OF4 WOODLAND)	(ROUND SNACK BOXES SET OF 4 FRUITS)	0.25	0.16	0.13	0.54	3.40	0.09	1.81
2	(PLASTERS IN TIN WOODLAND ANIMALS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.14	0.25	0.07	0.54	2.20	0.04	1.64
3	(ROUND SNACK BOXES SET OF4 WOODLAND)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.25	0.14	0.07	0.30	2.20	0.04	1.24
4	(SPACEBOY LUNCH BOX)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.10	0.25	0.07	0.68	2.78	0.04	2.37
5	(ROUND SNACK BOXES SET OF4 WOODLAND)	(SPACEBOY LUNCH BOX)	0.25	0.10	0.07	0.29	2.78	0.04	1.26
6	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN CIRCUS PARADE)	0.14	0.12	0.07	0.49	4.24	0.05	1.74
7	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.12	0.14	0.07	0.58	4.24	0.05	2.08
8	(WOODLAND CHARLOTTE BAG)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.13	0.25	0.06	0.50	2.04	0.03	1.51
9	(ROUND SNACK BOXES SET OF4 WOODLAND)	(WOODLAND CHARLOTTE BAG)	0.25	0.13	0.06	0.26	2.04	0.03	1.18

Top 10 - Regras de associação com maiores 'confianças'

In [35]:
```python
regras_associacao_germany.sort_values(by='confidence', axis=0, ascending=False, inplace=True)
regras_associacao_germany.iloc[0:10]
```

Out[35]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
13	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.07	0.13	0.06	0.84	6.65	0.05	5.59
0	(ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.16	0.25	0.13	0.83	3.40	0.09	4.53
4	(SPACEBOY LUNCH BOX)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.10	0.25	0.07	0.68	2.78	0.04	2.37
7	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.12	0.14	0.07	0.58	4.24	0.05	2.08
11	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.11	0.14	0.06	0.57	4.15	0.05	2.01
2	(PLASTERS IN TIN WOODLAND ANIMALS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.14	0.25	0.07	0.54	2.20	0.04	1.64
1	(ROUND SNACK BOXES SET OF4 WOODLAND)	(ROUND SNACK BOXES SET OF 4 FRUITS)	0.25	0.16	0.13	0.54	3.40	0.09	1.81
8	(WOODLAND CHARLOTTE BAG)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.13	0.25	0.06	0.50	2.04	0.03	1.51
6	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN CIRCUS PARADE)	0.14	0.12	0.07	0.49	4.24	0.05	1.74
15	(PLASTERS IN TIN CIRCUS PARADE)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.12	0.25	0.06	0.49	2.00	0.03	1.48

Top 10 - Regras de associação com maiores 'lifts'

Lift significa o quão influente uma transação A é em uma transação B, ou seja, comprando A aumenta ou diminui a probabilidade de alguém comprar B.
(Lift >= 1 indica relação positiva, e Lift < 1 negativa)

In [36]:
```python
regras_associacao_germany.sort_values(by='lift', axis=0, ascending=False, inplace=True)
regras_associacao_germany.iloc[0:10]
```

Out[36]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
13	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.07	0.13	0.06	0.84	6.65	0.05	5.59
12	(WOODLAND CHARLOTTE BAG)	(RED RETROSPOT CHARLOTTE BAG)	0.13	0.07	0.06	0.47	6.65	0.05	1.74
7	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.12	0.14	0.07	0.58	4.24	0.05	2.08
6	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN CIRCUS PARADE)	0.14	0.12	0.07	0.49	4.24	0.05	1.74
11	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.11	0.14	0.06	0.57	4.15	0.05	2.01
10	(PLASTERS IN TIN WOODLAND ANIMALS)	(PLASTERS IN TIN SPACEBOY)	0.14	0.11	0.06	0.44	4.15	0.05	1.61
0	(ROUND SNACK BOXES SET OF 4 FRUITS)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.16	0.25	0.13	0.83	3.40	0.09	4.53
1	(ROUND SNACK BOXES SET OF4 WOODLAND)	(ROUND SNACK BOXES SET OF 4 FRUITS)	0.25	0.16	0.13	0.54	3.40	0.09	1.81
4	(SPACEBOY LUNCH BOX)	(ROUND SNACK BOXES SET OF4 WOODLAND)	0.10	0.25	0.07	0.68	2.78	0.04	2.37
5	(ROUND SNACK BOXES SET OF4 WOODLAND)	(SPACEBOY LUNCH BOX)	0.25	0.10	0.07	0.29	2.78	0.04	1.26

Conclusão

Com estas informações já seria possível tentar realocar alguns produtos no supermercado, deixando-os mais próximos, por exemplo. Ou então colocando um produto em específico à mostra (porta de entrada do supermercado), pois costuma ser o mais comprado.