

## Tiến hành import các thư viện cần thiết như: mlxtend, xlrd, openpyxl

```
In [7]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

```
In [8]: df = pd.read_excel('Online Retail.xlsx')
```

```
In [9]: df.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]: df.head(10)
```

Out[10]:

	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 [11]:

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

In [12]:

```
df.head(10)
```

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	17850.0	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 [13]:

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

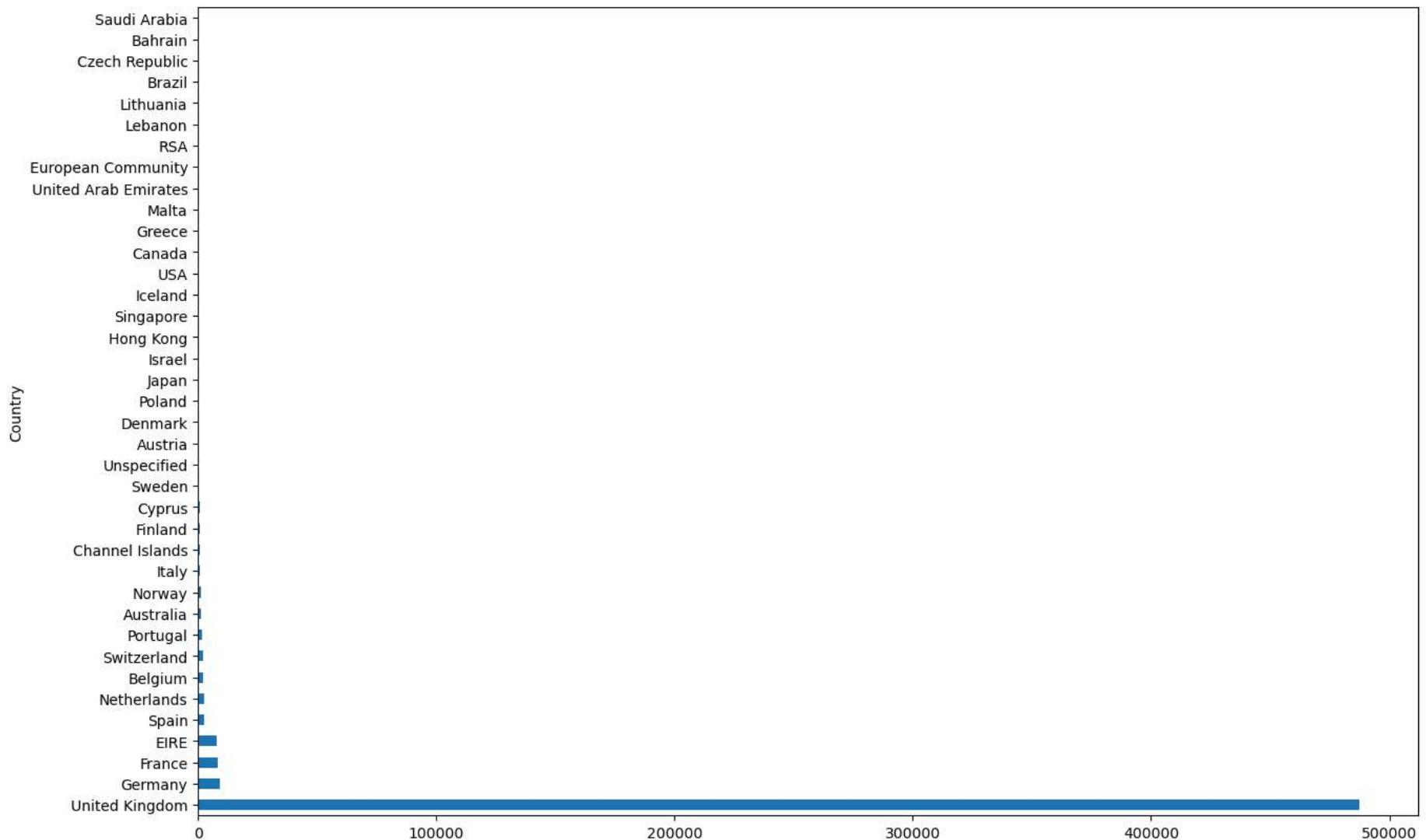
In [14]:

```
df=df[~df['InvoiceNo'].str.contains('C')]
```

In [15]:

```
df['Country'].value_counts().plot(kind='barh', figsize=(15,10))
```

```
Out[15]: <Axes: ylabel='Country'>
```



```
In [16]: basket = df[df['Country']=="Germany"].groupby(['InvoiceNo','Description'])['Quantity']
```

```
In [17]: basket = basket.sum().unstack().reset_index().fillna(0).set_index('InvoiceNo')
```

```
In [18]: basket.head(10)
```

Out[18]:

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	YU II ...
InvoiceNo											
536527	0.0	0.0	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	0.0	0.0	...
536861	0.0	0.0	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	0.0	0.0	...
536983	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537197	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537198	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537201	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537212	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
537250	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

10 rows × 1695 columns

In [19]:

```
def encode_data(datapoint):
    if datapoint <= 0:
        return 0
    if datapoint >= 1:
        return 1
```

In [20]:

```
basket = basket.applymap(encode_data)
```

```
C:\Users\tento\AppData\Local\Temp\ipykernel_19156\1901855995.py:1: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map instead.
```

```
basket = basket.applymap(encode_data)
```

```
In [21]: basket.drop('POSTAGE', inplace=True, axis=1)
```

```
In [22]: itemsets = apriori(basket.astype('bool'), min_support=0.05, use_colnames=True)
```

```
In [23]: itemsets.head(10)
```

```
Out[23]:
```

	<b>support</b>	<b>itemsets</b>
<b>0</b>	0.102845	(6 RIBBONS RUSTIC CHARM)
<b>1</b>	0.070022	(ALARM CLOCK BAKELIKE PINK)
<b>2</b>	0.065646	(CHARLOTTE BAG APPLES DESIGN)
<b>3</b>	0.050328	(CHILDRENS CUTLERY DOLLY GIRL)
<b>4</b>	0.061269	(COFFEE MUG APPLES DESIGN)
<b>5</b>	0.063457	(FAWN BLUE HOT WATER BOTTLE)
<b>6</b>	0.072210	(GUMBALL COAT RACK)
<b>7</b>	0.056893	(IVORY KITCHEN SCALES)
<b>8</b>	0.063457	(JAM JAR WITH PINK LID)
<b>9</b>	0.091904	(JAM MAKING SET PRINTED)

```
In [24]: rules = association_rules(itemsets, metric="confidence", min_threshold=0.5)
```

```
In [25]: rules.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   antecedents      8 non-null      object  
 1   consequents      8 non-null      object  
 2   antecedent support 8 non-null     float64 
 3   consequent support 8 non-null     float64 
 4   support          8 non-null      float64 
 5   confidence        8 non-null      float64 
 6   lift              8 non-null      float64 
 7   leverage          8 non-null      float64 
 8   conviction        8 non-null      float64 
 9   zhangs_metric     8 non-null      float64 
dtypes: float64(8), object(2)
memory usage: 772.0+ bytes
```

```
In [26]: rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x[0]).astype("unicode"))
rules["consequents"] = rules["consequents"].apply(lambda x: list(x[0]).astype("unicode"))
```

```
In [27]: for i in range(len(rules)):
    print(rules.loc[i, 'antecedents'], ' ==> ', rules.loc[i, 'consequents'],
          ' [', rules.loc[i, 'support'], ', ', rules.loc[i, 'confidence'], ']')
```

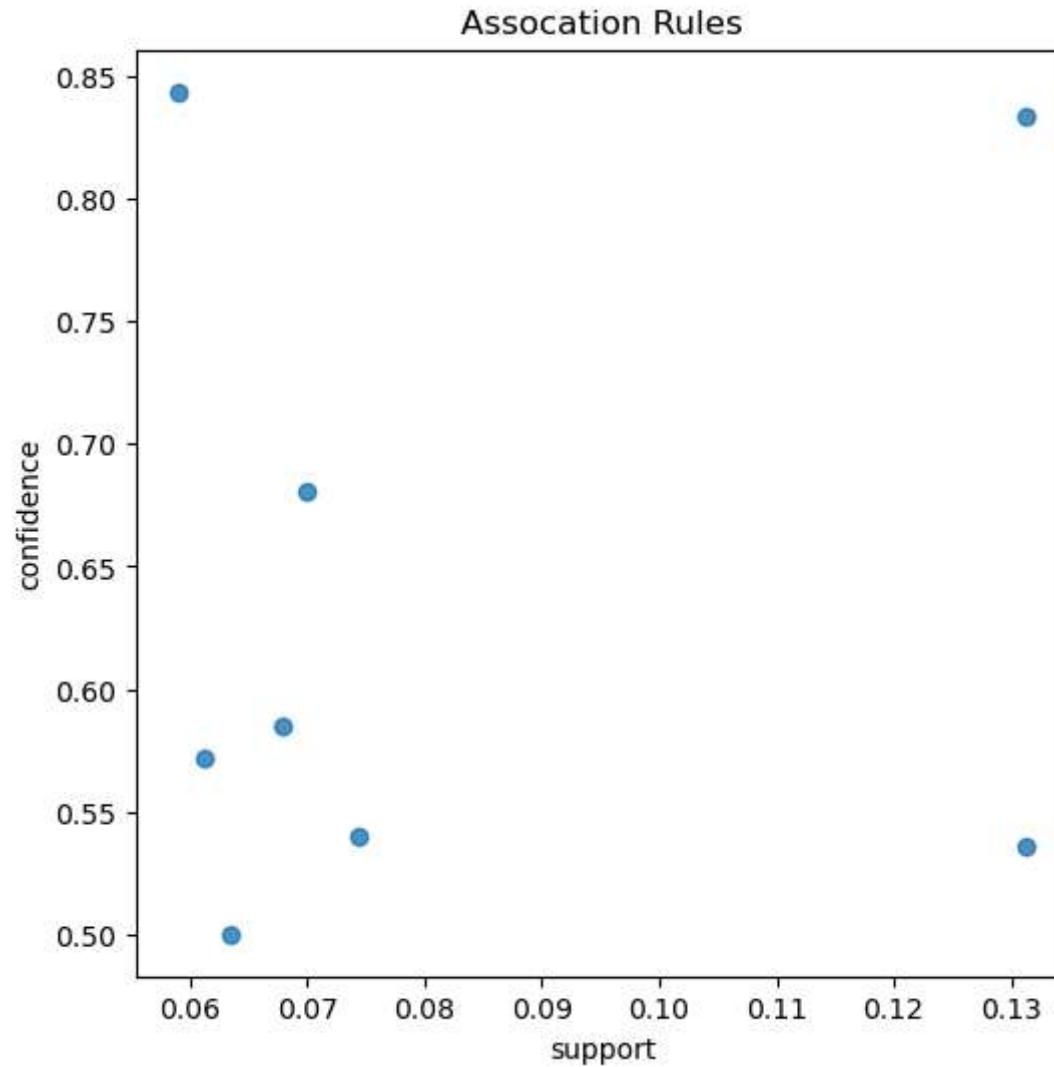
```
PLASTERS IN TIN CIRCUS PARADE ==> PLASTERS IN TIN WOODLAND ANIMALS [ 0.06783369803063458 , 0.5849056603773585 ]
PLASTERS IN TIN SPACEBOY ==> PLASTERS IN TIN WOODLAND ANIMALS [ 0.061269146608315096 , 0.5714285714285714 ]
PLASTERS IN TIN WOODLAND ANIMALS ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.07439824945295405 , 0.5396825396825397 ]
RED RETROSPOT CHARLOTTE BAG ==> WOODLAND CHARLOTTE BAG [ 0.05908096280087528 , 0.8437500000000001 ]
ROUND SNACK BOXES SET OF4 WOODLAND ==> ROUND SNACK BOXES SET OF 4 FRUITS [ 0.13129102844638948 , 0.5357142857142857 ]
ROUND SNACK BOXES SET OF 4 FRUITS ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.13129102844638948 , 0.8333333333333333 ]
SPACEBOY LUNCH BOX ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.0700218818380744 , 0.6808510638297872 ]
WOODLAND CHARLOTTE BAG ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.06345733041575492 , 0.5 ]
```

```
In [28]: support = rules['support'].values
confidence = rules['confidence'].values
```

```
In [29]: plt.figure(figsize=(6,6))
plt.title('Assocation Rules')
plt.xlabel('support')
```

```
plt.ylabel('confidence')
sns.regplot(x=support,y=confidence, fit_reg=False)
```

Out[29]: <Axes: title={'center': 'Assocation Rules'}, xlabel='support', ylabel='confidence'>



```
In [30]: from mlxtend.frequent_patterns import fpgrowth
itemsets = fpgrowth(basket.astype('bool'), min_support=0.05, use_colnames=True)
```

```
In [31]: rules = association_rules(itemsets, metric="confidence", min_threshold=0.5)
```

```
In [32]: rules.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   antecedents      8 non-null      object  
 1   consequents      8 non-null      object  
 2   antecedent support 8 non-null    float64 
 3   consequent support 8 non-null    float64 
 4   support          8 non-null      float64 
 5   confidence        8 non-null      float64 
 6   lift              8 non-null      float64 
 7   leverage          8 non-null      float64 
 8   conviction        8 non-null      float64 
 9   zhongs_metric     8 non-null      float64 
dtypes: float64(8), object(2)
memory usage: 772.0+ bytes
```

```
In [33]: rules["antecedents"] = rules["antecedents"].apply(lambda x: list(x)[0]).astype("unicode")
rules["consequents"] = rules["consequents"].apply(lambda x: list(x)[0]).astype("unicode")
```

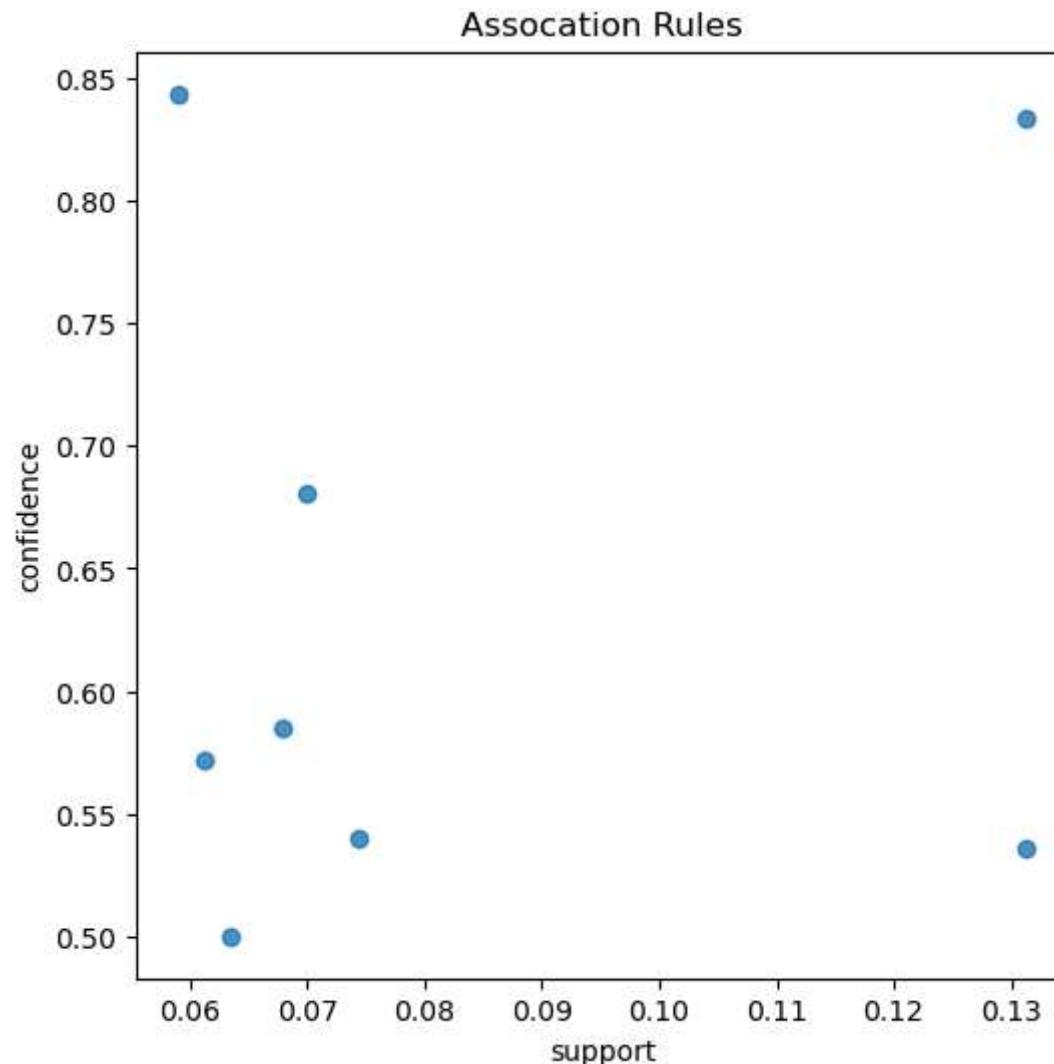
```
In [34]: for i in range(len(rules)):
    print(rules.loc[i, 'antecedents'], ' ==> ', rules.loc[i, 'consequents'],
          ' [', rules.loc[i, 'support'], ', ', rules.loc[i, 'confidence'], ']')
```

```
ROUND SNACK BOXES SET OF4 WOODLAND ==> ROUND SNACK BOXES SET OF 4 FRUITS [ 0.13129102844638948 , 0.5357142857142857 ]
ROUND SNACK BOXES SET OF 4 FRUITS ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.13129102844638948 , 0.8333333333333333 ]
WOODLAND CHARLOTTE BAG ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.06345733041575492 , 0.5 ]
PLASTERS IN TIN CIRCUS PARADE ==> PLASTERS IN TIN WOODLAND ANIMALS [ 0.06783369803063458 , 0.5849056603773585 ]
SPACEBOY LUNCH BOX ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.0700218818380744 , 0.6808510638297872 ]
RED RETROSPOT CHARLOTTE BAG ==> WOODLAND CHARLOTTE BAG [ 0.05908096280087528 , 0.8437500000000001 ]
PLASTERS IN TIN WOODLAND ANIMALS ==> ROUND SNACK BOXES SET OF4 WOODLAND [ 0.07439824945295405 , 0.5396825396825397 ]
PLASTERS IN TIN SPACEBOY ==> PLASTERS IN TIN WOODLAND ANIMALS [ 0.061269146608315096 , 0.5714285714285714 ]
```

```
In [35]: support = rules['support'].values
confidence = rules['confidence'].values
```

```
In [36]: plt.figure(figsize=(6,6))
plt.title('Assocation Rules')
plt.xlabel('support')
plt.ylabel('confidence')
sns.regplot(x=support,y=confidence, fit_reg=False)
```

```
Out[36]: <Axes: title={'center': 'Assocation Rules'}, xlabel='support', ylabel='confidence'>
```



**Kết luận hai thuật toán cho ra kết quả giống nhau**

In [ ]: