Assignment-09-Association Rules (Books)

In [1]:

```
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori,association_rules
from mlxtend.preprocessing import TransactionEncoder
```

In [2]:

```
# import dataset
book=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/book.csv")
book
```

Out[2]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas
0	0	1	0	1	0	0	1	0	0
1	1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	0	1	0	0
4	0	0	1	0	0	0	1	0	0
...	...	...	...	...	...	...	...	...	...
1995	0	0	1	0	0	1	1	1	0
1996	0	0	0	0	0	0	0	0	0
1997	0	0	0	0	0	0	0	0	0
1998	0	0	1	0	0	0	0	0	0
1999	0	0	0	0	0	0	0	0	0

2000 rows × 11 columns

In [3]:

```
# Data preprocessing not required as it already in transaction fromat
```

Apriori Algorithm

1. Association rules with 10% Support and 70% Confidence

In [4]:

```
# With 10% Support
frequent_itemsets = apriori(book,min_support=0.1,use_colnames=True)
frequent_itemsets
```

Out[4]:

	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
5	0.2410	(ArtBks)
6	0.2760	(GeogBks)
7	0.1135	(ItalCook)
8	0.1085	(Florence)
9	0.1650	(ChildBks, YouthBks)
10	0.2560	(ChildBks, CookBks)
11	0.1840	(ChildBks, DoItYBks)
12	0.1515	(ChildBks, RefBks)
13	0.1625	(ChildBks, ArtBks)
14	0.1950	(ChildBks, GeogBks)
15	0.1620	(YouthBks, CookBks)
16	0.1155	(YouthBks, DoItYBks)
17	0.1010	(YouthBks, ArtBks)
18	0.1205	(YouthBks, GeogBks)
19	0.1875	(DoItYBks, CookBks)
20	0.1525	(RefBks, CookBks)
21	0.1670	(ArtBks, CookBks)
22	0.1925	(GeogBks, CookBks)
23	0.1135	(ItalCook, CookBks)
24	0.1055	(RefBks, DoItYBks)
25	0.1235	(DoItYBks, ArtBks)
26	0.1325	(GeogBks, DoItYBks)
27	0.1105	(RefBks, GeogBks)
28	0.1275	(GeogBks, ArtBks)
29	0.1290	(ChildBks, YouthBks, CookBks)
30	0.1460	(ChildBks, DoItYBks, CookBks)
31	0.1225	(ChildBks, RefBks, CookBks)

	support	itemsets
32	0.1265	(ChildBks, ArtBks, CookBks)
33	0.1495	(ChildBks, GeogBks, CookBks)
34	0.1045	(ChildBks, DoItYBks, GeogBks)
35	0.1020	(ChildBks, GeogBks, ArtBks)
36	0.1015	(DoItYBks, ArtBks, CookBks)
37	0.1085	(GeogBks, DoItYBks, CookBks)
38	0.1035	(GeogBks, ArtBks, CookBks)

In [5]:

```
# With 70% confidence
rules=association_rules(frequent_itemsets,metric='lift',min_threshold=0.7)
rules
```

Out[5]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.06030
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.06030
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.07368
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.07368
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.06471
...	...	...	...	...	...	...	...	.
95	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.05710
96	(ArtBks, CookBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.05740
97	(GeogBks)	(ArtBks, CookBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.05740
98	(ArtBks)	(GeogBks, CookBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.05710
99	(CookBks)	(GeogBks, ArtBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.04854

100 rows × 9 columns

In [6]:

```
# A leverage value of 0 indicates independence. Range will be [-1,1]
# A high conviction value means that the consequent is highly depending on the antecedent a
```

In [7]:

```
rules.sort_values('lift',ascending=False)
```

Out[7]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
29	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.06458
28	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.06458
77	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.05715
80	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.05715
86	(ArtBks)	(DoItYBks, CookBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.05631
...	...	...	...	...	...	...	...	.
5	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.06471
13	(CookBks)	(YouthBks)	0.4310	0.2475	0.1620	0.375870	1.518667	0.05532
12	(YouthBks)	(CookBks)	0.2475	0.4310	0.1620	0.654545	1.518667	0.05532
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.07368
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.07368

100 rows × 9 columns

In [8]:

```
# Lift Ratio > 1 is a good influential rule in selecting the associated transaction
rules[rules.lift>1]
```

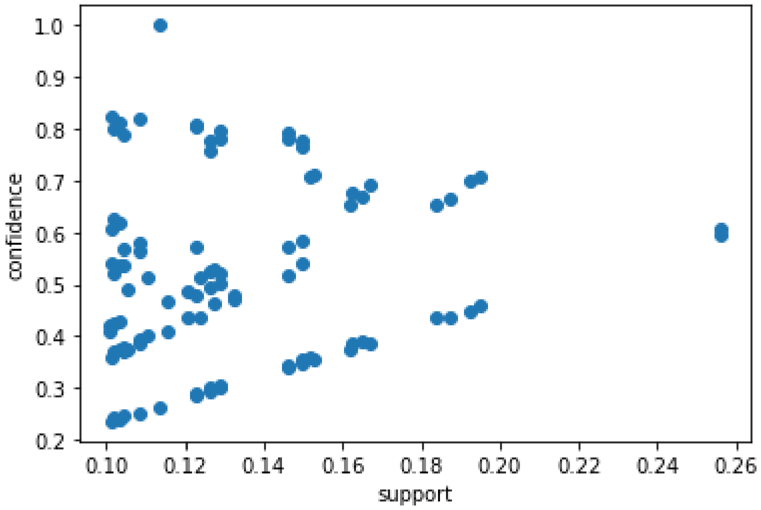
Out[8]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.06030
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.06030
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.07368
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.07368
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.06471
...	...	...	...	...	...	...	...	.
95	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.05710
96	(ArtBks, CookBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.05740
97	(GeogBks)	(ArtBks, CookBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.05740
98	(ArtBks)	(GeogBks, CookBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.05710
99	(CookBks)	(GeogBks, ArtBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.04854

100 rows × 9 columns

In [9]:

```
# Visualization of the obtained rule
plt.scatter(rules['support'],rules['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



2. Association rules with 20% Support and 60% confidence

In [10]:

```
# With 20% Support
frequent_itemsets2 = apriori(book,min_support=0.20,use_colnames=True)
frequent_itemsets2
```

Out[10]:

	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
5	0.2410	(ArtBks)
6	0.2760	(GeogBks)
7	0.2560	(ChildBks, CookBks)

In [11]:

```
# With 60% confidnce
rules2=association_rules(frequent_itemsets2,metric='lift',min_threshold=0.6)
rules2
```

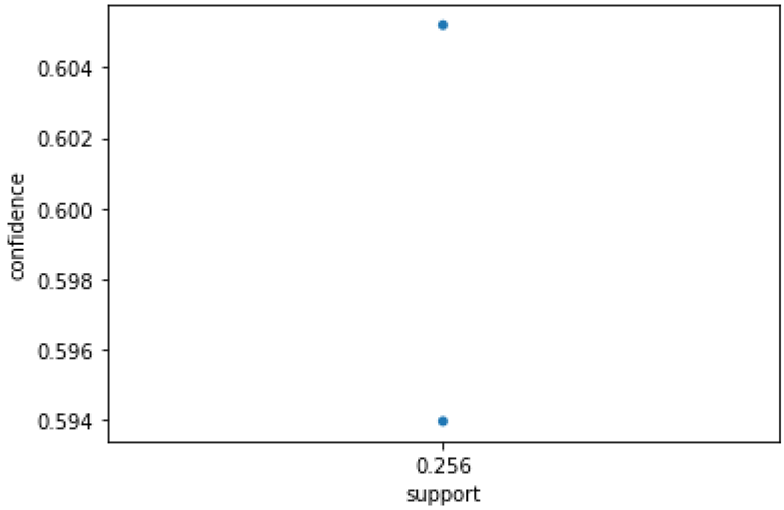
Out[11]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(ChildBks)	(CookBks)	0.423	0.431	0.256	0.605201	1.404179	0.073687
1	(CookBks)	(ChildBks)	0.431	0.423	0.256	0.593968	1.404179	0.073687

In [12]:

```
# Visualization of the obtained rule
sns.swarmplot(rules2['support'],rules2['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



3. Association rules with 5% Support and 80% confidence

In [13]:

```
# With 5% Support
frequent_itemsets3 = apriori(book,min_support=0.05,use_colnames=True)
frequent_itemsets3
```

Out[13]:

	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
...	...	...
95	0.0600	(YouthBks, DoItYBks, GeogBks, CookBks)
96	0.0560	(YouthBks, GeogBks, ArtBks, CookBks)
97	0.0650	(GeogBks, DoItYBks, ArtBks, CookBks)
98	0.0510	(YouthBks, CookBks, DoItYBks, GeogBks, ChildBks)
99	0.0535	(ArtBks, CookBks, DoItYBks, GeogBks, ChildBks)

100 rows × 2 columns

In [14]:

```
# With 80% confidence
rules3=association_rules(frequent_itemsets3,metric='lift',min_threshold=0.8)
rules3
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.0603
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.0603
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.0736
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.0736
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.0647
...	...	...	...	...	...	...	...	
657	(ArtBks)	(GeogBks, DoItYBks, ChildBks, CookBks)	0.2410	0.0890	0.0535	0.221992	2.494289	0.0320
658	(CookBks)	(GeogBks, DoItYBks, ArtBks, ChildBks)	0.4310	0.0595	0.0535	0.124130	2.086217	0.0278
659	(DoItYBks)	(GeogBks, ArtBks, ChildBks, CookBks)	0.2820	0.0835	0.0535	0.189716	2.272052	0.0299
660	(GeogBks)	(ChildBks, DoItYBks, ArtBks, CookBks)	0.2760	0.0820	0.0535	0.193841	2.363910	0.0308
661	(ChildBks)	(GeogBks, DoItYBks, ArtBks, CookBks)	0.4230	0.0650	0.0535	0.126478	1.945808	0.0260

662 rows × 9 columns

In [15]:

```
rules3[rules3.lift>1]
```

Out[15]:

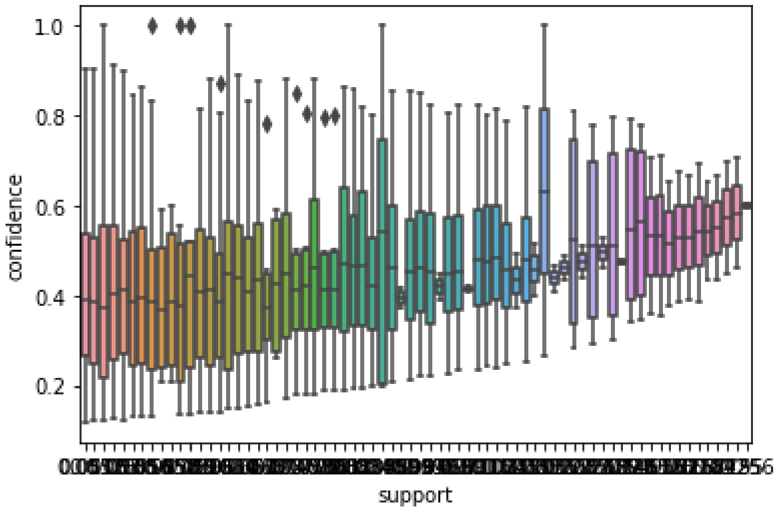
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.0603
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.0603
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.0736
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.0736
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.0647
...	...	...	...	...	...	...	...	
657	(ArtBks)	(GeogBks, DoItYBks, ChildBks, CookBks)	0.2410	0.0890	0.0535	0.221992	2.494289	0.0320
658	(CookBks)	(GeogBks, DoItYBks, ArtBks, ChildBks)	0.4310	0.0595	0.0535	0.124130	2.086217	0.0278
659	(DoItYBks)	(GeogBks, ArtBks, ChildBks, CookBks)	0.2820	0.0835	0.0535	0.189716	2.272052	0.0299
660	(GeogBks)	(ChildBks, DoItYBks, ArtBks, CookBks)	0.2760	0.0820	0.0535	0.193841	2.363910	0.0308
661	(ChildBks)	(GeogBks, DoItYBks, ArtBks, CookBks)	0.4230	0.0650	0.0535	0.126478	1.945808	0.0260

662 rows × 9 columns

In [16]:

```
# Visualization of the obtained rule
sns.boxplot(rules3['support'],rules3['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



In [ ]:

Assignment-09-Association Rules (my_movies)

In [17]:

```
# import dataset
movie=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/my_movies.csv")
movie
```

Out[17]:

	V1	V2	V3	V4	V5	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Patriot	LOT
0	Sixth Sense	LOTR1	Harry Potter1	Green Mile	LOTR2	1	0	1	1	0	
1	Gladiator	Patriot	Braveheart	NaN	NaN	0	1	0	0	1	
2	LOTR1	LOTR2	NaN	NaN	NaN	0	0	1	0	0	
3	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	
4	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	
5	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	
6	Harry Potter1	Harry Potter2	NaN	NaN	NaN	0	0	0	1	0	
7	Gladiator	Patriot	NaN	NaN	NaN	0	1	0	0	1	
8	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	
9	Sixth Sense	LOTR	Gladiator	Green Mile	NaN	1	1	0	0	0	

In [18]:

```
movie.shape
```

Out[18]:

```
(10, 15)
```

In [19]:

```
movie.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   V1             10 non-null     object
 1   V2             10 non-null     object
 2   V3             7 non-null      object
 3   V4             2 non-null      object
 4   V5             1 non-null      object
 5   Sixth Sense    10 non-null     int64 
 6   Gladiator      10 non-null     int64 
 7   LOTR1          10 non-null     int64 
 8   Harry Potter1  10 non-null     int64 
 9   Patriot        10 non-null     int64 
 10  LOTR2          10 non-null     int64 
 11  Harry Potter2  10 non-null     int64 
 12  LOTR           10 non-null     int64 
 13  Braveheart     10 non-null     int64 
 14  Green Mile     10 non-null     int64 
dtypes: int64(10), object(5)
memory usage: 1.3+ KB
```

In [20]:

```
movie2=movie.iloc[:,5:]
movie2
```

Out[20]:

	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Patriot	LOTR2	Harry Potter2	LOTR	Braveheart	Green Mile
0	1	0	1	1	0	1	0	0	0	1
1	0	1	0	0	1	0	0	0	1	0
2	0	0	1	0	0	1	0	0	0	0
3	1	1	0	0	1	0	0	0	0	0
4	1	1	0	0	1	0	0	0	0	0
5	1	1	0	0	1	0	0	0	0	0
6	0	0	0	1	0	0	1	0	0	0
7	0	1	0	0	1	0	0	0	0	0
8	1	1	0	0	1	0	0	0	0	0
9	1	1	0	0	0	0	0	1	0	1

Apriori Algorithm

1. Association rules with 10% Support and 70% confidence

In [21]:

```
# With 10% Support
frequent_itemsets5 = apriori(movie2,min_support=0.1,use_colnames=True)
frequent_itemsets5
```

Out[21]:

	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.2	(LOTR1)
3	0.2	(Harry Potter1)
4	0.6	(Patriot)
5	0.2	(LOTR2)
6	0.1	(Harry Potter2)
7	0.1	(LOTR)
8	0.1	(Braveheart)
9	0.2	(Green Mile)
10	0.5	(Gladiator, Sixth Sense)
11	0.1	(Sixth Sense, LOTR1)
12	0.1	(Sixth Sense, Harry Potter1)
13	0.4	(Patriot, Sixth Sense)
14	0.1	(Sixth Sense, LOTR2)
15	0.1	(LOTR, Sixth Sense)
16	0.2	(Green Mile, Sixth Sense)
17	0.6	(Gladiator, Patriot)
18	0.1	(Gladiator, LOTR)
19	0.1	(Gladiator, Braveheart)
20	0.1	(Gladiator, Green Mile)
21	0.1	(LOTR1, Harry Potter1)
22	0.2	(LOTR1, LOTR2)
23	0.1	(Green Mile, LOTR1)
24	0.1	(LOTR2, Harry Potter1)
25	0.1	(Harry Potter2, Harry Potter1)
26	0.1	(Green Mile, Harry Potter1)
27	0.1	(Patriot, Braveheart)
28	0.1	(Green Mile, LOTR2)
29	0.1	(Green Mile, LOTR)
30	0.4	(Gladiator, Patriot, Sixth Sense)
31	0.1	(Gladiator, LOTR, Sixth Sense)
32	0.1	(Gladiator, Green Mile, Sixth Sense)

	support	itemsets
33	0.1	(Sixth Sense, LOTR1, Harry Potter1)
34	0.1	(Sixth Sense, LOTR1, LOTR2)
35	0.1	(Green Mile, Sixth Sense, LOTR1)
36	0.1	(LOTR2, Sixth Sense, Harry Potter1)
37	0.1	(Green Mile, Sixth Sense, Harry Potter1)
38	0.1	(Green Mile, Sixth Sense, LOTR2)
39	0.1	(Green Mile, Sixth Sense, LOTR)
40	0.1	(Gladiator, Patriot, Braveheart)
41	0.1	(Gladiator, Green Mile, LOTR)
42	0.1	(LOTR2, LOTR1, Harry Potter1)
43	0.1	(Green Mile, LOTR1, Harry Potter1)
44	0.1	(Green Mile, LOTR1, LOTR2)
45	0.1	(Green Mile, Harry Potter1, LOTR2)
46	0.1	(Gladiator, Green Mile, Sixth Sense, LOTR)
47	0.1	(LOTR2, Sixth Sense, LOTR1, Harry Potter1)
48	0.1	(Green Mile, Sixth Sense, LOTR1, Harry Potter1)
49	0.1	(Green Mile, Sixth Sense, LOTR1, LOTR2)
50	0.1	(Green Mile, Sixth Sense, Harry Potter1, LOTR2)
51	0.1	(Green Mile, LOTR1, Harry Potter1, LOTR2)
52	0.1	(Green Mile, LOTR2, Harry Potter1, LOTR1, Sixt...

In [22]:

```
# With 70% confidence
rules5=association_rules(frequent_itemsets5,metric='lift',min_threshold=0.7)
rules5
```

Out[22]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.
2	(Sixth Sense)	(LOTR1)	0.6	0.2	0.1	0.166667	0.833333	-0.
3	(LOTR1)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.
4	(Sixth Sense)	(Harry Potter1)	0.6	0.2	0.1	0.166667	0.833333	-0.
...	...	...	...	...	...	...	...	
245	(Green Mile)	(Harry Potter1, LOTR1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
246	(LOTR2)	(Green Mile, Sixth Sense, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.
247	(Harry Potter1)	(Green Mile, Sixth Sense, LOTR1, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
248	(LOTR1)	(Green Mile, Harry Potter1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
249	(Sixth Sense)	(Green Mile, Harry Potter1, LOTR1, LOTR2)	0.6	0.1	0.1	0.166667	1.666667	0.

250 rows × 9 columns

In [23]:

```
# Lift Ratio > 1 is a good influential rule in selecting the associated transaction
rules5[rules5.lift>1]
```

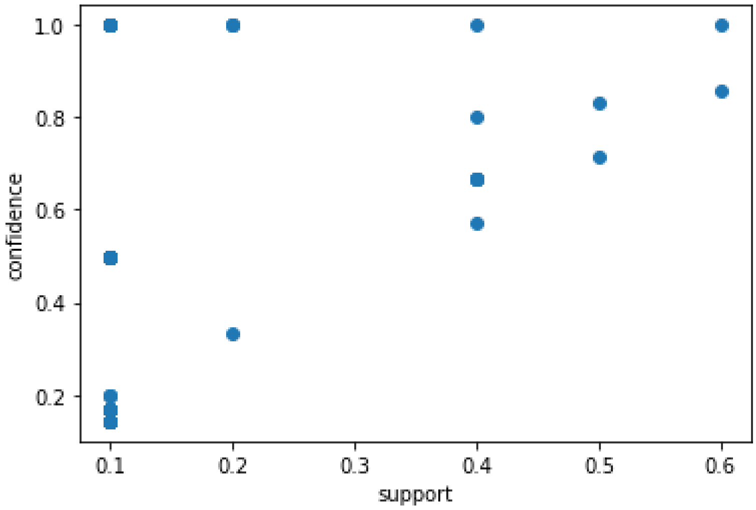
Out[23]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.
6	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.
7	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.
10	(LOTR)	(Sixth Sense)	0.1	0.6	0.1	1.000000	1.666667	0.
...	...	...	...	...	...	...	...	
245	(Green Mile)	(Harry Potter1, LOTR1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
246	(LOTR2)	(Green Mile, Sixth Sense, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.
247	(Harry Potter1)	(Green Mile, Sixth Sense, LOTR1, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
248	(LOTR1)	(Green Mile, Harry Potter1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
249	(Sixth Sense)	(Green Mile, Harry Potter1, LOTR1, LOTR2)	0.6	0.1	0.1	0.166667	1.666667	0.

236 rows × 9 columns

In [24]:

```
# Visualization of the obtained rule
plt.scatter(rules5['support'],rules5['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



2. Association rules with 5% Support and 90% cofidence

In [25]:

```
# with 5% support
frequent_itemsets6 = apriori(movie2,min_support=0.05,use_colnames=True)
frequent_itemsets6
```

Out[25]:

	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.2	(LOTR1)
3	0.2	(Harry Potter1)
4	0.6	(Patriot)
5	0.2	(LOTR2)
6	0.1	(Harry Potter2)
7	0.1	(LOTR)
8	0.1	(Braveheart)
9	0.2	(Green Mile)
10	0.5	(Gladiator, Sixth Sense)
11	0.1	(Sixth Sense, LOTR1)
12	0.1	(Sixth Sense, Harry Potter1)
13	0.4	(Patriot, Sixth Sense)
14	0.1	(Sixth Sense, LOTR2)
15	0.1	(LOTR, Sixth Sense)
16	0.2	(Green Mile, Sixth Sense)
17	0.6	(Gladiator, Patriot)
18	0.1	(Gladiator, LOTR)
19	0.1	(Gladiator, Braveheart)
20	0.1	(Gladiator, Green Mile)
21	0.1	(LOTR1, Harry Potter1)
22	0.2	(LOTR1, LOTR2)
23	0.1	(Green Mile, LOTR1)
24	0.1	(LOTR2, Harry Potter1)
25	0.1	(Harry Potter2, Harry Potter1)
26	0.1	(Green Mile, Harry Potter1)
27	0.1	(Patriot, Braveheart)
28	0.1	(Green Mile, LOTR2)
29	0.1	(Green Mile, LOTR)
30	0.4	(Gladiator, Patriot, Sixth Sense)
31	0.1	(Gladiator, LOTR, Sixth Sense)
32	0.1	(Gladiator, Green Mile, Sixth Sense)

	support	itemsets
33	0.1	(Sixth Sense, LOTR1, Harry Potter1)
34	0.1	(Sixth Sense, LOTR1, LOTR2)
35	0.1	(Green Mile, Sixth Sense, LOTR1)
36	0.1	(LOTR2, Sixth Sense, Harry Potter1)
37	0.1	(Green Mile, Sixth Sense, Harry Potter1)
38	0.1	(Green Mile, Sixth Sense, LOTR2)
39	0.1	(Green Mile, Sixth Sense, LOTR)
40	0.1	(Gladiator, Patriot, Braveheart)
41	0.1	(Gladiator, Green Mile, LOTR)
42	0.1	(LOTR2, LOTR1, Harry Potter1)
43	0.1	(Green Mile, LOTR1, Harry Potter1)
44	0.1	(Green Mile, LOTR1, LOTR2)
45	0.1	(Green Mile, Harry Potter1, LOTR2)
46	0.1	(Gladiator, Green Mile, Sixth Sense, LOTR)
47	0.1	(LOTR2, Sixth Sense, LOTR1, Harry Potter1)
48	0.1	(Green Mile, Sixth Sense, LOTR1, Harry Potter1)
49	0.1	(Green Mile, Sixth Sense, LOTR1, LOTR2)
50	0.1	(Green Mile, Sixth Sense, Harry Potter1, LOTR2)
51	0.1	(Green Mile, LOTR1, Harry Potter1, LOTR2)
52	0.1	(Green Mile, LOTR2, Harry Potter1, LOTR1, Sixt...

In [26]:

```
# 90% confidence
rules6=association_rules(frequent_itemsets6,metric='lift',min_threshold=0.9)
rules6
```

Out[26]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.
4	(LOTR)	(Sixth Sense)	0.1	0.6	0.1	1.000000	1.666667	0.
...	...	...	...	...	...	...	...	
233	(Green Mile)	(Harry Potter1, LOTR1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
234	(LOTR2)	(Green Mile, Sixth Sense, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.
235	(Harry Potter1)	(Green Mile, Sixth Sense, LOTR1, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
236	(LOTR1)	(Green Mile, Harry Potter1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
237	(Sixth Sense)	(Green Mile, Harry Potter1, LOTR1, LOTR2)	0.6	0.1	0.1	0.166667	1.666667	0.

238 rows × 9 columns

In [27]:

```
# Lift Ratio > 1 is a good influential rule in selecting the associated transaction
rules6[rules6.lift>1]
```

Out[27]:

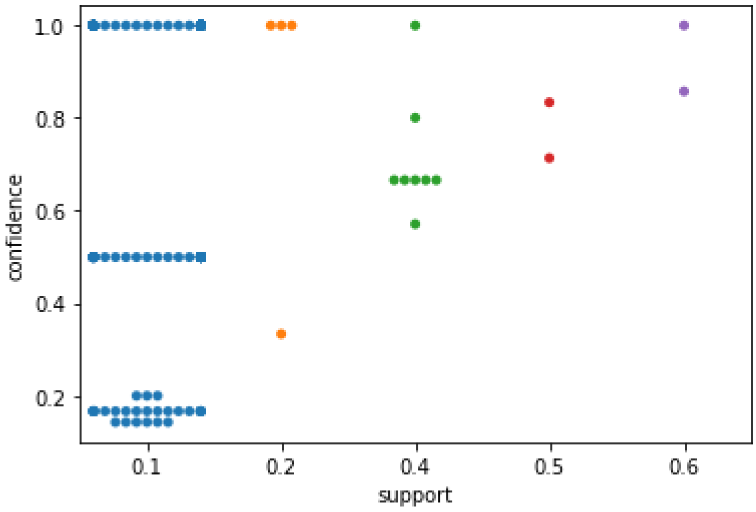
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.
4	(LOTR)	(Sixth Sense)	0.1	0.6	0.1	1.000000	1.666667	0.
...	...	...	...	...	...	...	...	
233	(Green Mile)	(Harry Potter1, LOTR1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
234	(LOTR2)	(Green Mile, Sixth Sense, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.
235	(Harry Potter1)	(Green Mile, Sixth Sense, LOTR1, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
236	(LOTR1)	(Green Mile, Harry Potter1, Sixth Sense, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.
237	(Sixth Sense)	(Green Mile, Harry Potter1, LOTR1, LOTR2)	0.6	0.1	0.1	0.166667	1.666667	0.

236 rows × 9 columns

In [28]:

```
# Visualization of the obtained rule
sns.swarmplot(rules6['support'],rules6['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\categorical.py:1296: Use
rWarning: 81.1% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



In [ ]: