In [1]:
```
!pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\nishi\anaconda3\lib\site-pac
kages (0.19.0)
Requirement already satisfied: scipy>=1.2.1 in c:\users\nishi\anaconda3\lib\sit
e-packages (from mlxtend) (1.6.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\nishi\anaconda3\li
b\site-packages (from mlxtend) (3.3.4)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\nishi\anaconda3
\lib\site-packages (from mlxtend) (0.24.1)
Requirement already satisfied: joblib>=0.13.2 in c:\users\nishi\anaconda3\lib\s
ite-packages (from mlxtend) (1.0.1)
Requirement already satisfied: pandas>=0.24.2 in c:\users\nishi\anaconda3\lib\s
ite-packages (from mlxtend) (1.2.4)
Requirement already satisfied: setuptools in c:\users\nishi\anaconda3\lib\site-
packages (from mlxtend) (52.0.0.post20210125)
Requirement already satisfied: numpy>=1.16.2 in c:\users\nishi\anaconda3\lib\si
te-packages (from mlxtend) (1.20.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\nishi\anaconda3
\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\nishi\anaconda3\li
b\site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\nishi\anaconda3\lib\si
te-packages (from matplotlib>=3.0.0->mlxtend) (8.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\u
sers\nishi\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.4.
7)
Requirement already satisfied: cycler>=0.10 in c:\users\nishi\anaconda3\lib\sit
e-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: six in c:\users\nishi\anaconda3\lib\site-package
s (from cycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\nishi\anaconda3\lib\sit
e-packages (from pandas>=0.24.2->mlxtend) (2021.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\nishi\anaconda3
\lib\site-packages (from scikit-learn>=0.20.3->mlxtend) (2.1.0)
```

Book

In [2]:
```
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 [7]:
```
Book=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Association_rules\\book
Book
```

Out[7]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	Ital/
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 [8]:
```
Book
```

Out[8]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	Ital/
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 [9]:
```
Book.head()
```

Out[9]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	Ital
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	

Preprocessing

In [10]:
```
# Lets get the data in the transaction format
df=pd.get_dummies(Book)
df.head()
```

Out[10]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	ItalArt
0	0	1	0	1	0	0	1	0	0	0
1	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	1	0	1	0	0	0
4	0	0	1	0	0	0	1	0	0	0

Apriori Algorithm

In [12]:
```python
freq_itemsets = apriori(df, min_support=0.1, use_colnames=True)
freq_itemsets
```

Out[12]:

	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	(CookBks, YouthBks)
16	0.1155	(DoItYBks, YouthBks)
17	0.1010	(ArtBks, YouthBks)
18	0.1205	(GeogBks, YouthBks)
19	0.1875	(DoItYBks, CookBks)
20	0.1525	(CookBks, RefBks)
21	0.1670	(ArtBks, CookBks)
22	0.1925	(GeogBks, CookBks)
23	0.1135	(ItalCook, CookBks)
24	0.1055	(DoItYBks, RefBks)
25	0.1235	(DoItYBks, ArtBks)
26	0.1325	(DoItYBks, GeogBks)
27	0.1105	(GeogBks, RefBks)
28	0.1275	(GeogBks, ArtBks)
29	0.1290	(ChildBks, CookBks, YouthBks)
30	0.1460	(ChildBks, DoItYBks, CookBks)
31	0.1225	(ChildBks, CookBks, RefBks)
32	0.1265	(ChildBks, ArtBks, CookBks)

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

In [13]:
```
rules = association_rules(freq_itemsets, metric="lift", min_threshold=0.7)
rules
```

Out[13]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714
...	...	...	...	...	...	...	...	...
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408
96	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107
97	(ArtBks)	(GeogBks, CookBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408
99	(CookBks)	(GeogBks, ArtBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547

100 rows × 9 columns

In [14]:
```
rules.sort_values('lift',ascending = False).head(15)
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
29	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064582
28	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582
77	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150
80	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150
85	(ArtBks)	(DoItYBks, CookBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.056313
84	(DoItYBks, CookBks)	(ArtBks)	0.1875	0.2410	0.1015	0.541333	2.246196	0.056313
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408
97	(ArtBks)	(GeogBks, CookBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107
96	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107
52	(ChildBks, CookBks)	(RefBks)	0.2560	0.2145	0.1225	0.478516	2.230842	0.067588
57	(RefBks)	(ChildBks, CookBks)	0.2145	0.2560	0.1225	0.571096	2.230842	0.067588
76	(ChildBks, GeogBks)	(ArtBks)	0.1950	0.2410	0.1020	0.523077	2.170444	0.055005
81	(ArtBks)	(ChildBks, GeogBks)	0.2410	0.1950	0.1020	0.423237	2.170444	0.055005
86	(DoItYBks)	(CookBks, ArtBks)	0.2820	0.1670	0.1015	0.359929	2.155264	0.054406

#An 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 and range [0 inf]

In [15]:
```
rules.sort_values('lift',ascending = False)[0:20]
```

Out[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
29	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064582
28	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582
77	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150
80	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150
85	(ArtBks)	(DoItYBks, CookBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.056313
84	(DoItYBks, CookBks)	(ArtBks)	0.1875	0.2410	0.1015	0.541333	2.246196	0.056313
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408
97	(ArtBks)	(GeogBks, CookBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107
96	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107
52	(ChildBks, CookBks)	(RefBks)	0.2560	0.2145	0.1225	0.478516	2.230842	0.067588
57	(RefBks)	(ChildBks, CookBks)	0.2145	0.2560	0.1225	0.571096	2.230842	0.067588
76	(ChildBks, GeogBks)	(ArtBks)	0.1950	0.2410	0.1020	0.523077	2.170444	0.055005
81	(ArtBks)	(ChildBks, GeogBks)	0.2410	0.1950	0.1020	0.423237	2.170444	0.055005
86	(DoItYBks)	(CookBks, ArtBks)	0.2820	0.1670	0.1015	0.359929	2.155264	0.054406
83	(CookBks, ArtBks)	(DoItYBks)	0.1670	0.2820	0.1015	0.607784	2.155264	0.054406
65	(ChildBks, CookBks)	(GeogBks)	0.2560	0.2760	0.1495	0.583984	2.115885	0.078844
68	(GeogBks)	(ChildBks, CookBks)	0.2760	0.2560	0.1495	0.541667	2.115885	0.078844
89	(DoItYBks, CookBks)	(GeogBks)	0.1875	0.2760	0.1085	0.578667	2.096618	0.056750
92	(GeogBks)	(DoItYBks, CookBks)	0.2760	0.1875	0.1085	0.393116	2.096618	0.056750

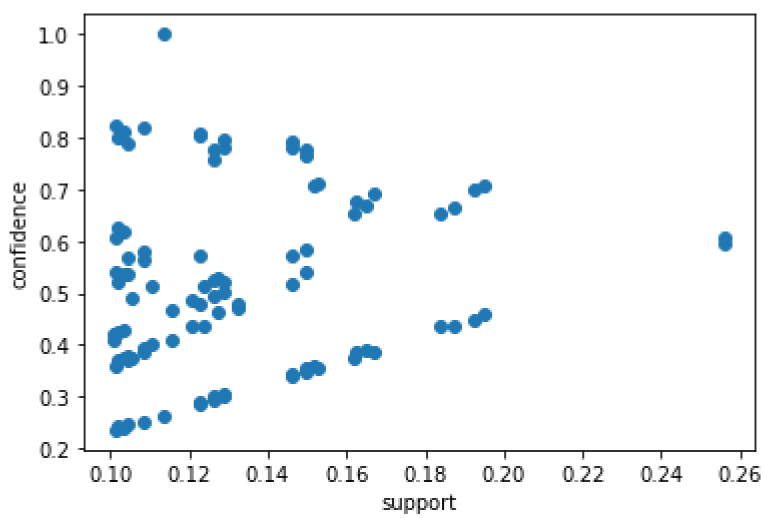
In [16]:
```
rules[rules.lift>1]
```

Out[16]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714
...	...	...	...	...	...	...	...	...
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408
96	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107
97	(ArtBks)	(GeogBks, CookBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408
99	(CookBks)	(GeogBks, ArtBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547

100 rows × 9 columns

In [17]:
```
plt.scatter(rules['support'],rules['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



In [18]:
```
frequent_itemsets = apriori(df, min_support=0.05, use_colnames=True)
frequent_itemsets
```

Out[18]:

	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	(DoItYBks, GeogBks, CookBks, YouthBks)
96	0.0560	(ArtBks, GeogBks, CookBks, YouthBks)
97	0.0650	(ArtBks, DoItYBks, GeogBks, CookBks)
98	0.0510	(DoItYBks, GeogBks, YouthBks, CookBks, ChildBks)
99	0.0535	(DoItYBks, GeogBks, ArtBks, CookBks, ChildBks)

100 rows × 2 columns

In [19]:
```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=0.7)
rules
```

Out[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714
...	...	...	...	...	...	...	...	...
657	(DoItYBks)	(CookBks, ChildBks, GeogBks, ArtBks)	0.2820	0.0835	0.0535	0.189716	2.272052	0.029953
658	(GeogBks)	(CookBks, DoItYBks, ArtBks, ChildBks)	0.2760	0.0820	0.0535	0.193841	2.363910	0.030868
659	(ArtBks)	(ChildBks, DoItYBks, GeogBks, CookBks)	0.2410	0.0890	0.0535	0.221992	2.494289	0.032051
660	(CookBks)	(ChildBks, DoItYBks, GeogBks, ArtBks)	0.4310	0.0595	0.0535	0.124130	2.086217	0.027856
661	(ChildBks)	(CookBks, DoItYBks, GeogBks, ArtBks)	0.4230	0.0650	0.0535	0.126478	1.945808	0.026005

662 rows × 9 columns

In [20]:
```
rules.sort_values('lift',ascending = False).head(15)
```

Out[20]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
183	(ItalCook)	(CookBks, YouthBks)	0.1135	0.1620	0.0590	0.519824	3.208789	0.040613
182	(CookBks, YouthBks)	(ItalCook)	0.1620	0.1135	0.0590	0.364198	3.208789	0.040613
513	(GeogBks, CookBks)	(ChildBks, ItalCook)	0.1925	0.0850	0.0525	0.272727	3.208556	0.036137
508	(ChildBks, ItalCook)	(GeogBks, CookBks)	0.0850	0.1925	0.0525	0.617647	3.208556	0.036137
511	(GeogBks, ItalCook)	(ChildBks, CookBks)	0.0640	0.2560	0.0525	0.820312	3.204346	0.036116
510	(ChildBks, CookBks)	(GeogBks, ItalCook)	0.2560	0.0640	0.0525	0.205078	3.204346	0.036116
646	(CookBks, ChildBks, ArtBks)	(DoItYBks, GeogBks)	0.1265	0.1325	0.0535	0.422925	3.191886	0.036739
647	(DoItYBks, GeogBks)	(CookBks, ChildBks, ArtBks)	0.1325	0.1265	0.0535	0.403774	3.191886	0.036739
515	(ItalCook)	(ChildBks, GeogBks, CookBks)	0.1135	0.1495	0.0525	0.462555	3.094014	0.035532
506	(ChildBks, GeogBks, CookBks)	(ItalCook)	0.1495	0.1135	0.0525	0.351171	3.094014	0.035532
639	(ChildBks, DoItYBks, GeogBks)	(CookBks, ArtBks)	0.1045	0.1670	0.0535	0.511962	3.065639	0.036048
654	(CookBks, ArtBks)	(ChildBks, DoItYBks, GeogBks)	0.1670	0.1045	0.0535	0.320359	3.065639	0.036048
355	(ChildBks, YouthBks)	(DoItYBks, RefBks)	0.1650	0.1055	0.0530	0.321212	3.044665	0.035592
358	(DoItYBks, RefBks)	(ChildBks, YouthBks)	0.1055	0.1650	0.0530	0.502370	3.044665	0.035592
655	(ChildBks, ArtBks)	(DoItYBks, GeogBks, CookBks)	0.1625	0.1085	0.0535	0.329231	3.034385	0.035869

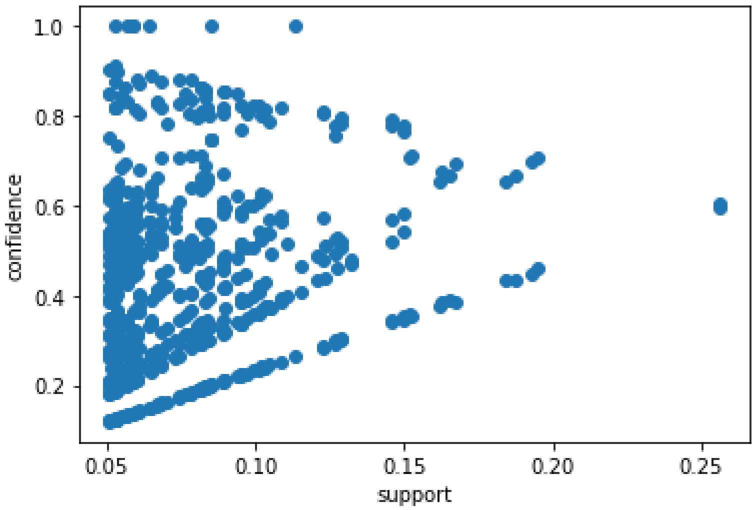
In [21]:
```
rules[rules.lift>1]
```

Out[21]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714
...	...	...	...	...	...	...	...	...
657	(DoItYBks)	(CookBks, ChildBks, GeogBks, ArtBks)	0.2820	0.0835	0.0535	0.189716	2.272052	0.029953
658	(GeogBks)	(CookBks, DoItYBks, ArtBks, ChildBks)	0.2760	0.0820	0.0535	0.193841	2.363910	0.030868
659	(ArtBks)	(ChildBks, DoItYBks, GeogBks, CookBks)	0.2410	0.0890	0.0535	0.221992	2.494289	0.032051
660	(CookBks)	(ChildBks, DoItYBks, GeogBks, ArtBks)	0.4310	0.0595	0.0535	0.124130	2.086217	0.027856
661	(ChildBks)	(CookBks, DoItYBks, GeogBks, ArtBks)	0.4230	0.0650	0.0535	0.126478	1.945808	0.026005

662 rows × 9 columns

In [22]:
```
plt.scatter(rules['support'],rules['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



Movies

In [58]:
```
movies=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Association_rules\\my
```

In [59]:
```
movies
```

Out[59]:

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

In [60]:
```
df1=pd.get_dummies(movies)
df1.head()
```

Out[60]:

	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Patriot	LOTR2	Harry Potter2	LOTR	Braveheart	Green Mile	...	V2_L
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 rows × 25 columns

In [61]:
```
freq_itemsets = apriori(df1, min_support=0.1, use_colnames=True)
freq_itemsets
```

Out[61]:

	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.2	(LOTR1)
3	0.2	(Harry Potter1)
4	0.6	(Patriot)
...	...	...
1392	0.1	(V1_Sixth Sense, LOTR1, V5_LOTR2, V2_LOTR1, V3...
1393	0.1	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...
1394	0.1	(V1_Sixth Sense, LOTR2, V5_LOTR2, V2_LOTR1, V3...
1395	0.1	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...
1396	0.1	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...

1397 rows × 2 columns

In [63]:
```python
Rules = association_rules(freq_itemsets, metric="lift", min_threshold=0.9)
Rules
```

Out[63]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.0
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.0
2	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.0
3	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.0
4	(Sixth Sense)	(LOTR)	0.6	0.1	0.1	0.166667	1.666667	0.0
...	...	...	...	...	...	...	...	
64211	(V3_Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.1	0.1	0.1	1.000000	10.000000	0.0
64212	(Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.0
64213	(Sixth Sense)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.6	0.1	0.1	0.166667	1.666667	0.0
64214	(V4_Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.0
64215	(Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.0

64216 rows × 9 columns

```
In [64]: Rules.sort_values('lift',ascending = False).head(15)
```

Out[64]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
32108	(Sixth Sense, V3_Harry Potter1, Harry Potter1,...	(V5_LOTR2, LOTR1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33435	(LOTR1, LOTR2, V5_LOTR2, Harry Potter1, Sixth ...	(V3_Harry Potter1)	0.1	0.1	0.1	1.0	10.0	0.09
33413	(Sixth Sense, LOTR2)	(LOTR1, V5_LOTR2, V2_LOTR1, Harry Potter1, V4_...	0.1	0.1	0.1	1.0	10.0	0.09
33414	(V4_Green Mile, LOTR2)	(LOTR1, V5_LOTR2, V2_LOTR1, Harry Potter1, Six...	0.1	0.1	0.1	1.0	10.0	0.09
33415	(V5_LOTR2, V2_LOTR1)	(LOTR1, LOTR2, Harry Potter1, Sixth Sense, V4_...	0.1	0.1	0.1	1.0	10.0	0.09
33416	(V5_LOTR2, Harry Potter1)	(LOTR1, LOTR2, V2_LOTR1, Sixth Sense, V4_Green...	0.1	0.1	0.1	1.0	10.0	0.09
33417	(V5_LOTR2, Sixth Sense)	(LOTR1, LOTR2, V2_LOTR1, Harry Potter1, V4_Gre...	0.1	0.1	0.1	1.0	10.0	0.09
33418	(V5_LOTR2, V4_Green Mile)	(LOTR1, LOTR2, V2_LOTR1, Harry Potter1, Sixth ...	0.1	0.1	0.1	1.0	10.0	0.09
33419	(V2_LOTR1, Harry Potter1)	(LOTR1, LOTR2, V5_LOTR2, Sixth Sense, V4_Green...	0.1	0.1	0.1	1.0	10.0	0.09
33420	(Sixth Sense, V2_LOTR1)	(LOTR1, LOTR2, V5_LOTR2, Harry Potter1, V4_Gre...	0.1	0.1	0.1	1.0	10.0	0.09

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
33421	(V4_Green Mile, V2_LOTR1)	(LOTR1, LOTR2, V5_LOTR2, Harry Potter1, Sixth ...	0.1	0.1	0.1	1.0	10.0	0.09
33422	(Sixth Sense, Harry Potter1)	(LOTR1, LOTR2, V5_LOTR2, V2_LOTR1, V4_Green Mile)	0.1	0.1	0.1	1.0	10.0	0.09
33423	(V4_Green Mile, Harry Potter1)	(LOTR1, LOTR2, V5_LOTR2, V2_LOTR1, Sixth Sense)	0.1	0.1	0.1	1.0	10.0	0.09
52644	(V5_LOTR2, V2_LOTR1, V4_Green Mile, LOTR1)	(Sixth Sense, V1_Sixth Sense, Green Mile, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
52643	(V5_LOTR2, Sixth Sense, V2_LOTR1, LOTR1)	(V4_Green Mile, V1_Sixth Sense, Green Mile, LO...	0.1	0.1	0.1	1.0	10.0	0.09

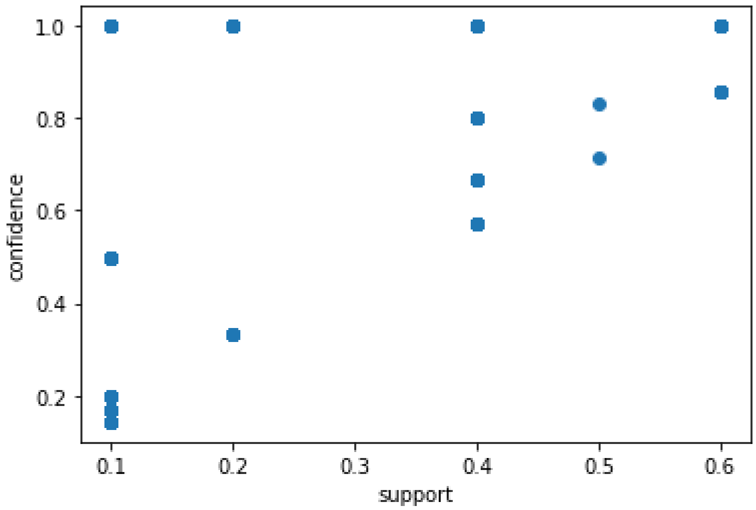
In [47]:
```
Rules[Rules.lift>1]
```

Out[47]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.(
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.(
2	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.(
3	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.(
4	(Sixth Sense)	(LOTR)	0.6	0.1	0.1	0.166667	1.666667	0.(
...	...	...	...	...	...	...	...	
64211	(V3_Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.1	0.1	0.1	1.000000	10.000000	0.(
64212	(Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.(
64213	(Sixth Sense)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.6	0.1	0.1	0.166667	1.666667	0.(
64214	(V4_Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.(
64215	(Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.(

64202 rows × 9 columns

In [65]:
```python
plt.scatter(Rules['support'],Rules['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



In [66]:
```python
freq_itemsets = apriori(df1, min_support=0.05, use_colnames=True)
freq_itemsets
```

Out[66]:

	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.2	(LOTR1)
3	0.2	(Harry Potter1)
4	0.6	(Patriot)
...	...	...
1392	0.1	(V1_Sixth Sense, LOTR1, V5_LOTR2, V2_LOTR1, V3...
1393	0.1	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...
1394	0.1	(V1_Sixth Sense, LOTR2, V5_LOTR2, V2_LOTR1, V3...
1395	0.1	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...
1396	0.1	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...

1397 rows × 2 columns

In [67]:
```
Rules = association_rules(freq_itemsets, metric="lift", min_threshold=0.7)
Rules
```

Out[67]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.0
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.0
2	(Sixth Sense)	(LOTR1)	0.6	0.2	0.1	0.166667	0.833333	-0.0
3	(LOTR1)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.0
4	(Sixth Sense)	(Harry Potter1)	0.6	0.2	0.1	0.166667	0.833333	-0.0
...	...	...	...	...	...	...	...	
64247	(V3_Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.1	0.1	0.1	1.000000	10.000000	0.0
64248	(Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.0
64249	(Sixth Sense)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.6	0.1	0.1	0.166667	1.666667	0.0
64250	(V4_Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.0
64251	(Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.0

64252 rows × 9 columns

In [69]:
```
Rules.sort_values('lift',ascending = False).head(15)
```

Out[69]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
32126	(Sixth Sense, V3_Harry Potter1, LOTR1, Harry P...	(V5_LOTR2, Green Mile, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33437	(V4_Green Mile, V2_LOTR1, Harry Potter1)	(V5_LOTR2, Sixth Sense, LOTR1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33415	(Sixth Sense, V2_LOTR1, LOTR1)	(V5_LOTR2, V4_Green Mile, Harry Potter1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33416	(V2_LOTR1, V4_Green Mile, LOTR1)	(V5_LOTR2, Sixth Sense, Harry Potter1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33417	(Sixth Sense, LOTR1, Harry Potter1)	(V5_LOTR2, V4_Green Mile, V2_LOTR1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33418	(V4_Green Mile, LOTR1, Harry Potter1)	(V5_LOTR2, Sixth Sense, V2_LOTR1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33419	(Sixth Sense, V4_Green Mile, LOTR1)	(V5_LOTR2, V2_LOTR1, Harry Potter1, LOTR2)	0.1	0.1	0.1	1.0	10.0	0.09
33420	(V5_LOTR2, V2_LOTR1, LOTR2)	(Sixth Sense, V4_Green Mile, LOTR1, Harry Pott...	0.1	0.1	0.1	1.0	10.0	0.09
33421	(V5_LOTR2, Harry Potter1, LOTR2)	(Sixth Sense, V2_LOTR1, V4_Green Mile, LOTR1)	0.1	0.1	0.1	1.0	10.0	0.09
33422	(V5_LOTR2, Sixth Sense, LOTR2)	(V2_LOTR1, V4_Green Mile, LOTR1, Harry Potter1)	0.1	0.1	0.1	1.0	10.0	0.09
33423	(V5_LOTR2, V4_Green Mile, LOTR2)	(Sixth Sense, V2_LOTR1, LOTR1, Harry Potter1)	0.1	0.1	0.1	1.0	10.0	0.09

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
33424	(V2_LOTR1, Harry Potter1, LOTR2)	(V5_LOTR2, Sixth Sense, V4_Green Mile, LOTR1)	0.1	0.1	0.1	1.0	10.0	0.09
33425	(Sixth Sense, V2_LOTR1, LOTR2)	(V5_LOTR2, V4_Green Mile, LOTR1, Harry Potter1)	0.1	0.1	0.1	1.0	10.0	0.09
33426	(V4_Green Mile, V2_LOTR1, LOTR2)	(V5_LOTR2, Sixth Sense, LOTR1, Harry Potter1)	0.1	0.1	0.1	1.0	10.0	0.09
33427	(Sixth Sense, Harry Potter1, LOTR2)	(V5_LOTR2, V2_LOTR1, V4_Green Mile, LOTR1)	0.1	0.1	0.1	1.0	10.0	0.09

In [70]:
```
Rules[Rules.lift>1]
```

Out[70]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.(
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.(
6	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.(
7	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.(
10	(Sixth Sense)	(LOTR)	0.6	0.1	0.1	0.166667	1.666667	0.(
...	...	...	...	...	...	...	...	
64247	(V3_Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.1	0.1	0.1	1.000000	10.000000	0.(
64248	(Harry Potter1)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.(
64249	(Sixth Sense)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.6	0.1	0.1	0.166667	1.666667	0.(
64250	(V4_Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.(
64251	(Green Mile)	(V1_Sixth Sense, LOTR1, LOTR2, V5_LOTR2, V2_LO...	0.2	0.1	0.1	0.500000	5.000000	0.(

64202 rows × 9 columns

In [71]:
```
plt.scatter(Rules['support'],Rules['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```