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

```
Collecting mlxtend
  Downloading mlxtend-0.19.0-py2.py3-none-any.whl (1.3 MB)
Requirement already satisfied: joblib>=0.13.2 in c:\users\hp\anaconda3\lib\site
-packages (from mlxtend) (1.0.1)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\hp\anaconda3\li
b\site-packages (from mlxtend) (0.23.1)
Requirement already satisfied: scipy>=1.2.1 in c:\users\hp\anaconda3\lib\site-p
ackages (from mlxtend) (1.6.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\hp\anaconda3\lib\s
ite-packages (from mlxtend) (3.3.4)
Requirement already satisfied: numpy>=1.16.2 in c:\users\hp\anaconda3\lib\site-
packages (from mlxtend) (1.19.2)
Requirement already satisfied: pandas>=0.24.2 in c:\users\hp\anaconda3\lib\site
-packages (from mlxtend) (1.2.4)
Requirement already satisfied: setuptools in c:\users\hp\anaconda3\lib\site-pac
kages (from mlxtend) (52.0.0.post20210125)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\u
sers\hp\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.4.7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-
packages (from matplotlib>=3.0.0->mlxtend) (8.2.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\s
ite-packages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\hp\anaconda3\li
b\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-p
ackages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages
(from cycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\hp\anaconda3\lib\site-p
ackages (from pandas>=0.24.2->mlxtend) (2021.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\hp\anaconda3\li
b\site-packages (from scikit-learn>=0.20.3->mlxtend) (2.1.0)
Installing collected packages: mlxtend
Successfully installed mlxtend-0.19.0
```

In [26]:
```
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 [27]:
```
movie=pd.read_csv('my_movies.csv')
movie
```

Out[27]:

	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 [28]:
```
movie.shape
```

Out[28]: (10, 15)

In [29]:
```
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: 1.0+ KB
```

In [30]:
```
movie1 = movie.iloc[:,:5]
movie1
```

Out[30]:

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

In [31]:
```
movie2=movie.iloc[:,5:]
movie2
```

Out[31]:

	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

In [50]:
```
movie2.isna().sum()
```

Out[50]:
```
Sixth Sense      0
Gladiator        0
LOTR1            0
Harry Potter1    0
Patriot          0
LOTR2            0
Harry Potter2    0
LOTR             0
Braveheart       0
Green Mile       0
dtype: int64
```

1 value of Support

In [51]:
```
# with 10% support
freq_itemsets = apriori(movie2, min_support = 0.1, use_colnames=True)
freq_itemsets
```

Out[51]:

	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	(Sixth Sense, Gladiator)
11	0.1	(Sixth Sense, LOTR1)
12	0.1	(Sixth Sense, Harry Potter1)
13	0.4	(Sixth Sense, Patriot)
14	0.1	(LOTR2, Sixth Sense)
15	0.1	(Sixth Sense, LOTR)
16	0.2	(Sixth Sense, Green Mile)
17	0.6	(Patriot, Gladiator)
18	0.1	(Gladiator, LOTR)
19	0.1	(Gladiator, Braveheart)
20	0.1	(Green Mile, Gladiator)
21	0.1	(LOTR1, Harry Potter1)
22	0.2	(LOTR2, LOTR1)
23	0.1	(Green Mile, LOTR1)
24	0.1	(LOTR2, Harry Potter1)
25	0.1	(Harry Potter1, Harry Potter2)
26	0.1	(Green Mile, Harry Potter1)
27	0.1	(Patriot, Braveheart)
28	0.1	(LOTR2, Green Mile)
29	0.1	(Green Mile, LOTR)
30	0.4	(Patriot, Sixth Sense, Gladiator)
31	0.1	(Sixth Sense, Gladiator, LOTR)
32	0.1	(Sixth Sense, Green Mile, Gladiator)

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

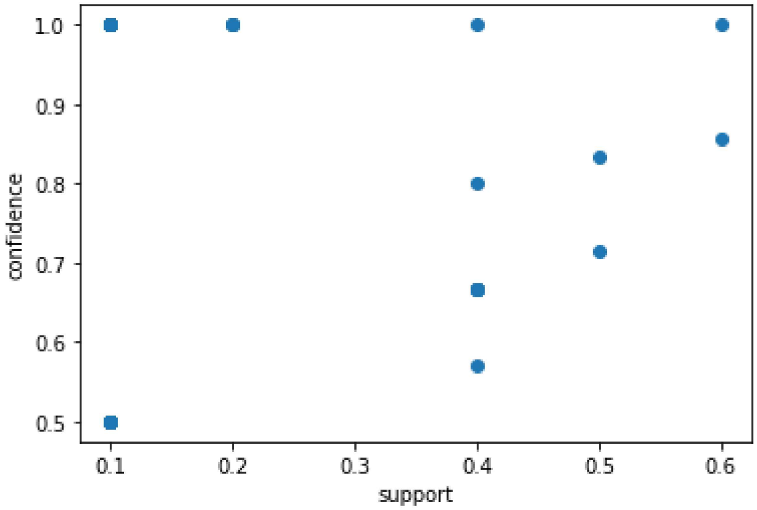
In [52]:
```
# 50% confidence
AR1 = association_rules(freq_itemsets, metric = 'confidence', min_threshold=0.5)
AR1
```

Out[52]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08
2	(LOTR1)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.02
3	(Harry Potter1)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.02
4	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04
...	...	...	...	...	...	...	...	...
211	(LOTR2, LOTR1)	(Sixth Sense, Green Mile, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08
212	(Harry Potter1)	(LOTR2, Sixth Sense, Green Mile, LOTR1)	0.2	0.1	0.1	0.500000	5.000000	0.08
213	(Green Mile)	(LOTR2, Sixth Sense, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08
214	(LOTR2)	(LOTR1, Sixth Sense, Green Mile, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08
215	(LOTR1)	(LOTR2, Sixth Sense, Green Mile, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08

216 rows × 9 columns

In [53]:
```
plt.scatter(AR1['support'], AR1['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```

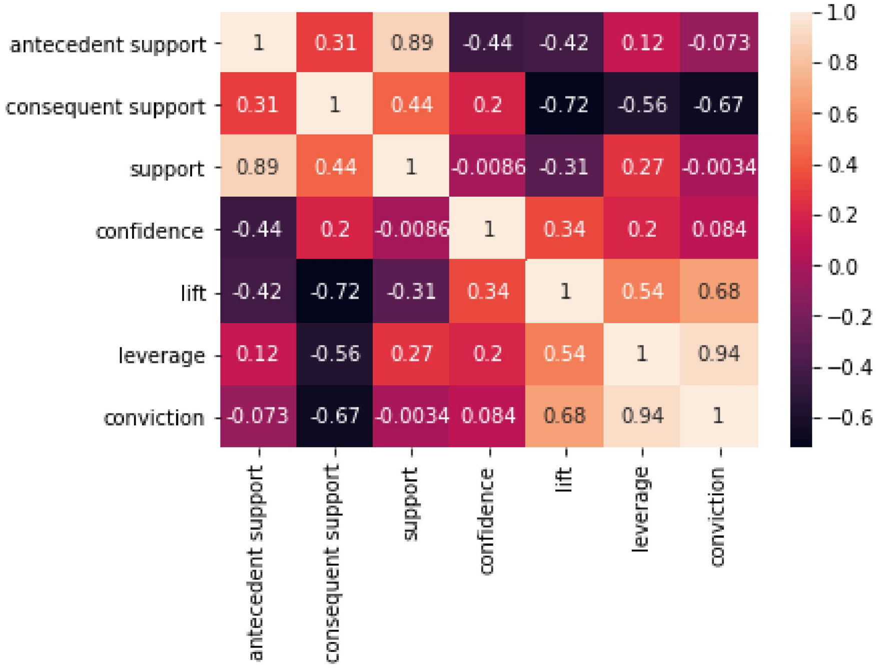


In [54]:
```
corr_AR1 = AR1.corr()
corr_AR1
```

Out[54]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	0.306857	0.889789	-0.439667	-0.420128	0.119624	-0.072571
consequent support	0.306857	1.000000	0.438872	0.195239	-0.720211	-0.562317	-0.665919
support	0.889789	0.438872	1.000000	-0.008629	-0.307204	0.272735	-0.003399
confidence	-0.439667	0.195239	-0.008629	1.000000	0.342850	0.201372	0.084379
lift	-0.420128	-0.720211	-0.307204	0.342850	1.000000	0.538233	0.676383
leverage	0.119624	-0.562317	0.272735	0.201372	0.538233	1.000000	0.936802
conviction	-0.072571	-0.665919	-0.003399	0.084379	0.676383	0.936802	1.000000

In [55]:
```
sns.heatmap(data=corr_AR1, annot=True)
plt.show()
```



2. Value of support

In [56]:
```
# with 15% support
freq_itemsets_2 = apriori(movie2, min_support = 0.15, use_colnames=True)
freq_itemsets_2
```

Out[56]:

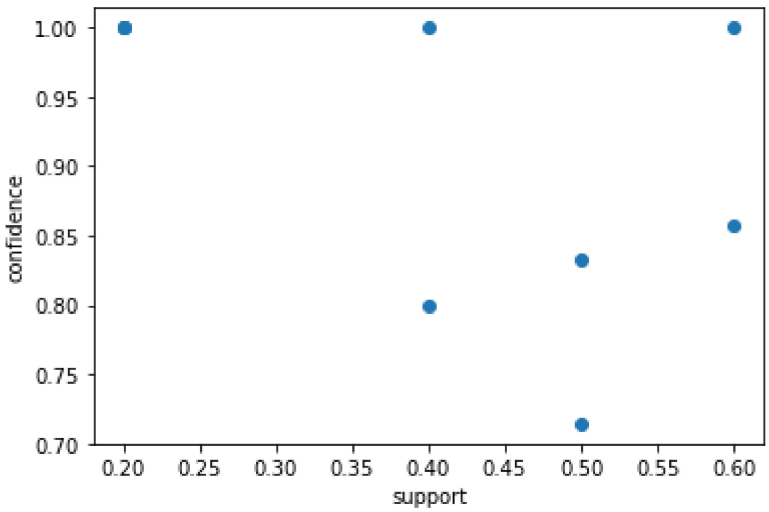
	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.2	(LOTR1)
3	0.2	(Harry Potter1)
4	0.6	(Patriot)
5	0.2	(LOTR2)
6	0.2	(Green Mile)
7	0.5	(Sixth Sense, Gladiator)
8	0.4	(Sixth Sense, Patriot)
9	0.2	(Sixth Sense, Green Mile)
10	0.6	(Patriot, Gladiator)
11	0.2	(LOTR2, LOTR1)
12	0.4	(Patriot, Sixth Sense, Gladiator)

In [57]:
```
# 70% confidence
AR2 = association_rules(freq_itemsets_2, metric = 'confidence', min_threshold=0.7
AR2
```

Out[57]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	co
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	
2	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.08	
3	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	
4	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	
5	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.16	
6	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.16	
7	(Sixth Sense, Patriot)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	
8	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	

In [58]:
```
plt.scatter(AR2['support'], AR2['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```

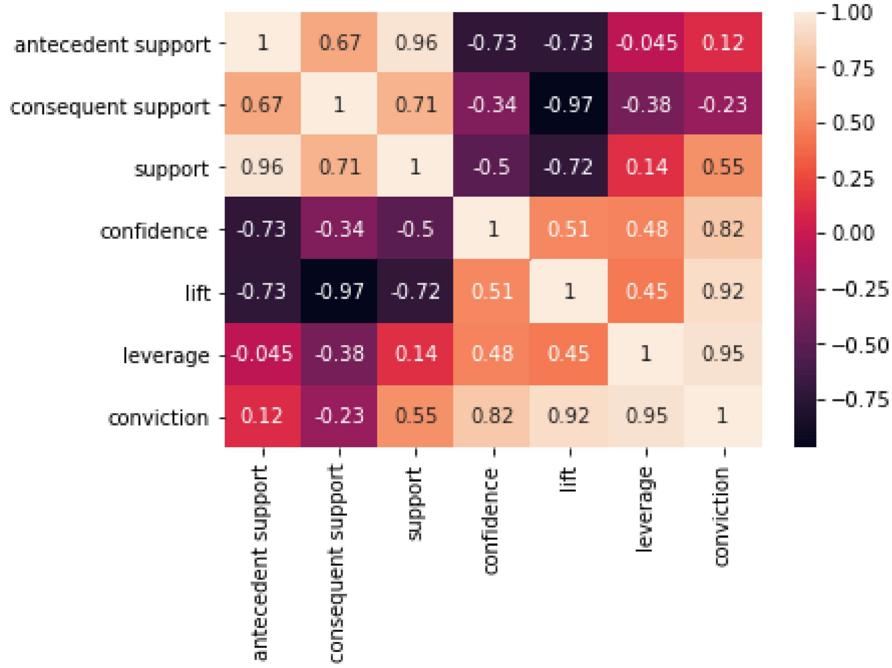


In [59]:
```
corr_AR2 = AR2.corr()
corr_AR2
```

Out[59]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	0.666724	0.956456	-0.725272	-0.727076	-0.044923	0.118262
consequent support	0.666724	1.000000	0.713616	-0.339730	-0.972862	-0.381039	-0.226455
support	0.956456	0.713616	1.000000	-0.498743	-0.716884	0.138343	0.554700
confidence	-0.725272	-0.339730	-0.498743	1.000000	0.509452	0.483948	0.819590
lift	-0.727076	-0.972862	-0.716884	0.509452	1.000000	0.450579	0.924500
leverage	-0.044923	-0.381039	0.138343	0.483948	0.450579	1.000000	0.951303
conviction	0.118262	-0.226455	0.554700	0.819590	0.924500	0.951303	1.000000

In [60]:
```
sns.heatmap(data=corr_AR2, annot = True)
plt.show()
```



3. Value of support

In [61]:
```
# with 20% support
freq_itemsets_3 = apriori(movie2, min_support = 0.20, use_colnames=True)
freq_itemsets_3
```

Out[61]:

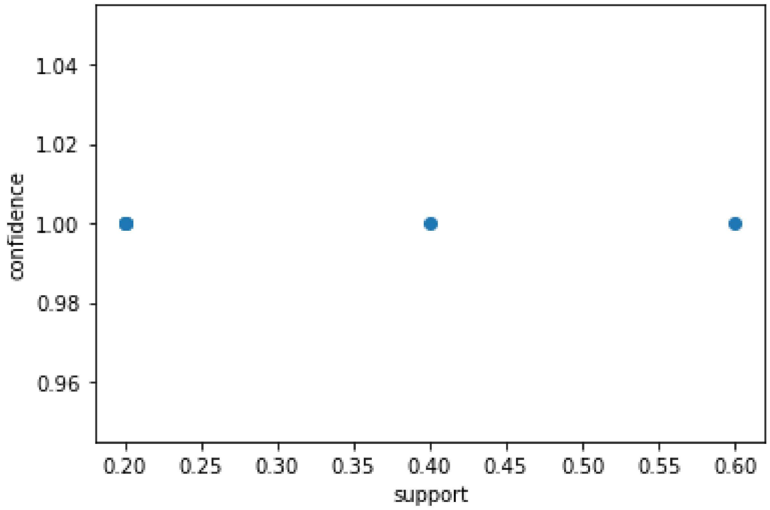
	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.2	(LOTR1)
3	0.2	(Harry Potter1)
4	0.6	(Patriot)
5	0.2	(LOTR2)
6	0.2	(Green Mile)
7	0.5	(Sixth Sense, Gladiator)
8	0.4	(Sixth Sense, Patriot)
9	0.2	(Sixth Sense, Green Mile)
10	0.6	(Patriot, Gladiator)
11	0.2	(LOTR2, LOTR1)
12	0.4	(Patriot, Sixth Sense, Gladiator)

In [62]:
```
# 90% confidence
AR3 = association_rules(freq_itemsets_3, metric = 'confidence', min_threshold=0.9
AR3
```

Out[62]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	cc
0	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.0	1.666667	0.08	
1	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.0	1.428571	0.18	
2	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.0	5.000000	0.16	
3	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.0	5.000000	0.16	
4	(Sixth Sense, Patriot)	(Gladiator)	0.4	0.7	0.4	1.0	1.428571	0.12	

In [63]:
```
plt.scatter(AR3['support'], AR3['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```

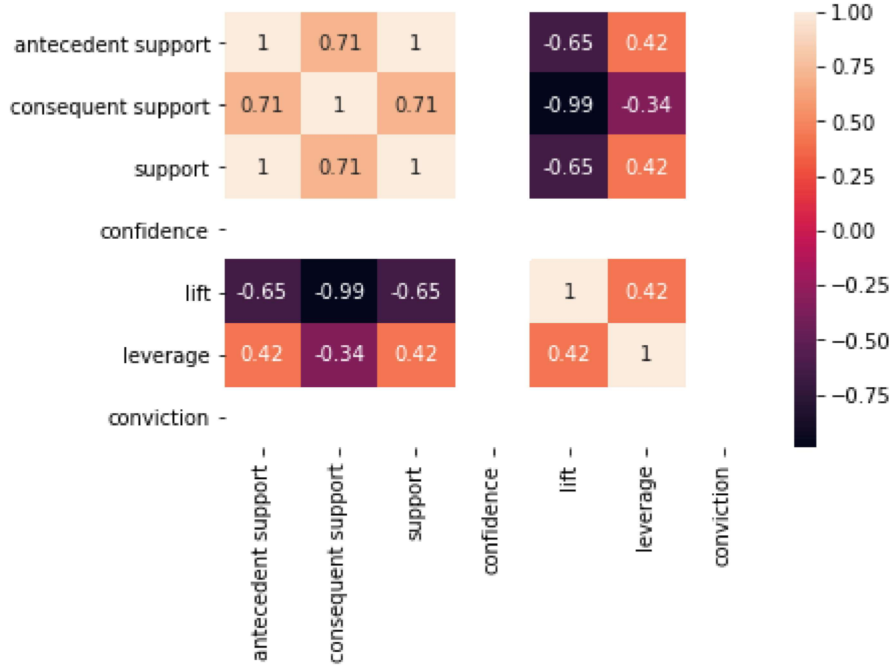


In [64]:
```
corr_AR3 = AR3.corr()
corr_AR3
```

Out[64]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	0.712691	1.000000	NaN	-0.646332	0.419263	NaN
consequent support	0.712691	1.000000	0.712691	NaN	-0.994216	-0.338042	NaN
support	1.000000	0.712691	1.000000	NaN	-0.646332	0.419263	NaN
confidence	NaN	NaN	NaN	NaN	NaN	NaN	NaN
lift	-0.646332	-0.994216	-0.646332	NaN	1.000000	0.419587	NaN
leverage	0.419263	-0.338042	0.419263	NaN	0.419587	1.000000	NaN
conviction	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [49]:
```
sns.heatmap(data=corr_AR3, annot=True)
plt.show()
```



In [ ]: