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

```
Requirement already satisfied: mlxtend in c:\users\nandini\anaconda3\lib\site
-packages (0.19.0)Note: you may need to restart the kernel to use updated pac
kages.

Requirement already satisfied: joblib>=0.13.2 in c:\users\nandini\anaconda3\l
ib\site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\nandini\anaco
nda3\lib\site-packages (from mlxtend) (0.24.1)
Requirement already satisfied: setuptools in c:\users\nandini\anaconda3\lib\s
ite-packages (from mlxtend) (52.0.0.post20210125)
Requirement already satisfied: numpy>=1.16.2 in c:\users\nandini\anaconda3\li
b\site-packages (from mlxtend) (1.20.1)
Requirement already satisfied: scipy>=1.2.1 in c:\users\nandini\anaconda3\lib
\site-packages (from mlxtend) (1.6.2)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\nandini\anaconda
3\lib\site-packages (from mlxtend) (3.3.4)
Requirement already satisfied: pandas>=0.24.2 in c:\users\nandini\anaconda3\l
ib\site-packages (from mlxtend) (1.2.4)
Requirement already satisfied: cycler>=0.10 in c:\users\nandini\anaconda3\lib
```

Problem Statement:-

Prepare rules for the all the data sets 1) Try different values of support and confidence. Observe the change in number of rules for different support,confidence values 2) Change the minimum length in apriori algorithm 3) Visulize the obtained rules using different plots

1. Import Neccesary Libraries

In [21]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from mlxtend.preprocessing import transactionencoder
from mlxtend.frequent_patterns import apriori, association_rules
```

2. Import Data

In [3]:
```
My_Movies = pd.read_csv('my_movies.csv')
My_Movies
```

Out[3]:

	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

3. Data Understanding

In [4]:
```
My_Movies.shape
```

Out[4]:
```
(10, 15)
```

In [5]:
```
My_Movies.isna().sum()
```

Out[5]:
```
V1              0
V2              0
V3              3
V4              8
V5              9
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
```

In [6]:
```

My_Movies.dtypes
```

Out[6]:
```
V1              object
V2              object
V3              object
V4              object
V5              object
Sixth Sense      int64
Gladiator        int64
LOTR1            int64
Harry Potter1    int64
Patriot          int64
LOTR2            int64
Harry Potter2    int64
LOTR             int64
Braveheart       int64
Green Mile       int64
dtype: object
```

In [7]:
```
My_Movies.head(10)
```

Out[7]:

	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 [8]:
```
My_Movies.describe(include='all')
```

Out[8]:

	V1	V2	V3	V4	V5	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Pa
count	10	10	7	2	1	10.000000	10.000000	10.000000	10.000000	10.000
unique	4	5	4	1	1	NaN	NaN	NaN	NaN	
top	Gladiator	Patriot	Sixth Sense	Green Mile	LOTR2	NaN	NaN	NaN	NaN	
freq	6	6	4	2	1	NaN	NaN	NaN	NaN	
mean	NaN	NaN	NaN	NaN	NaN	0.600000	0.700000	0.200000	0.200000	0.600
std	NaN	NaN	NaN	NaN	NaN	0.516398	0.483046	0.421637	0.421637	0.516
min	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	0.000000	0.000000	0.000
25%	NaN	NaN	NaN	NaN	NaN	0.000000	0.250000	0.000000	0.000000	0.000
50%	NaN	NaN	NaN	NaN	NaN	1.000000	1.000000	0.000000	0.000000	1.000
75%	NaN	NaN	NaN	NaN	NaN	1.000000	1.000000	0.000000	0.000000	1.000
max	NaN	NaN	NaN	NaN	NaN	1.000000	1.000000	1.000000	1.000000	1.000

4. Data Preparation

In [9]:
```
My_Movies=My_Movies.iloc[:,5:]
My_Movies
```

Out[9]:

	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 [10]:
```
My_Movies
```

Out[10]:

	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

5. Choosing different values of Support & Confidence

5.a value of support '5%'

In [12]:
```
frequent_items = apriori(df = My_Movies, min_support=0.05, use_colnames=True, ma
frequent_items
```

Out[12]:

	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	(LOTR1, Sixth Sense)
12	0.1	(Sixth Sense, Harry Potter1)
13	0.4	(Patriot, Sixth Sense)
14	0.1	(LOTR2, Sixth Sense)
15	0.1	(LOTR, Sixth Sense)
16	0.2	(Green Mile, Sixth Sense)
17	0.6	(Patriot, Gladiator)
18	0.1	(LOTR, Gladiator)
19	0.1	(Braveheart, Gladiator)
20	0.1	(Green Mile, Gladiator)
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)

5.b With confidence of '30%'

In [15]:
```
Association_Rules_1 = association_rules(df = frequent_items, metric='confidence',
Association_Rules_1
```

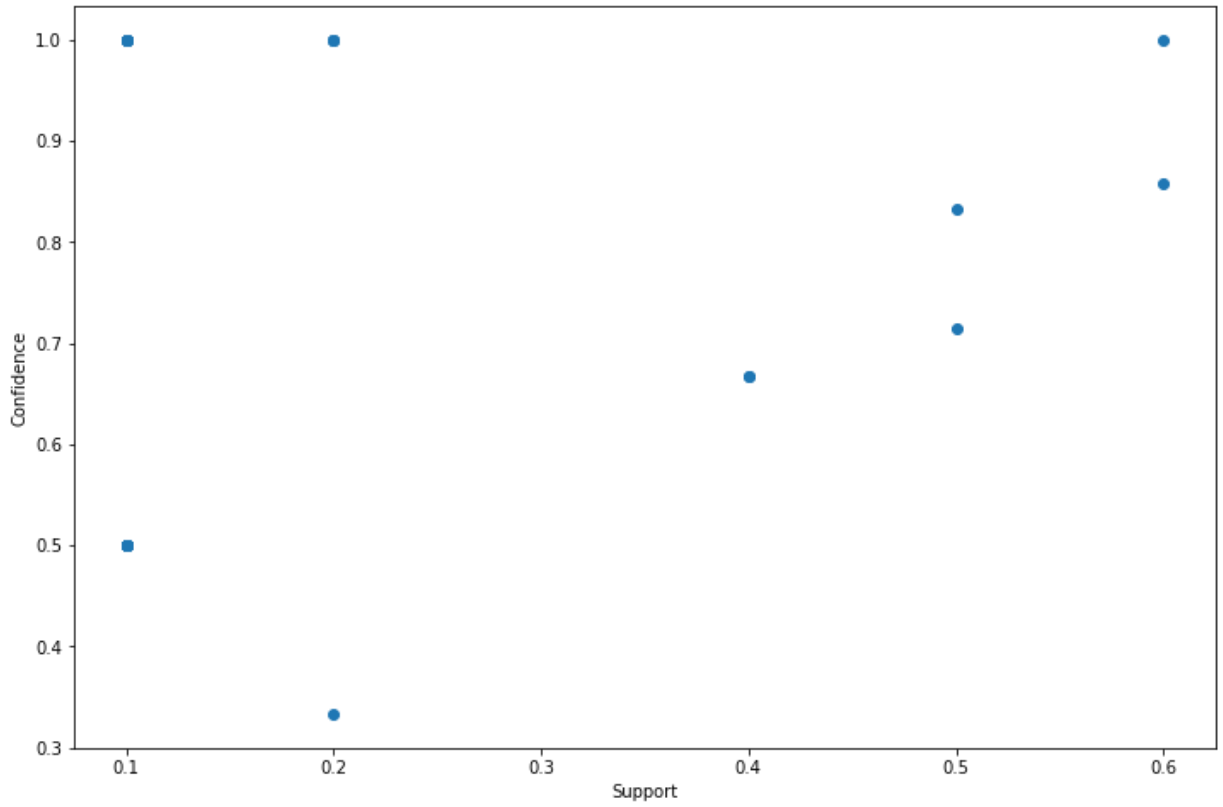
Out[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	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	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04
5	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04
6	(LOTR2)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.02
7	(LOTR)	(Sixth Sense)	0.1	0.6	0.1	1.000000	1.666667	0.04
8	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.08
9	(Sixth Sense)	(Green Mile)	0.6	0.2	0.2	0.333333	1.666667	0.08
10	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18
11	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18
12	(LOTR)	(Gladiator)	0.1	0.7	0.1	1.000000	1.428571	0.03
13	(Braveheart)	(Gladiator)	0.1	0.7	0.1	1.000000	1.428571	0.03
14	(Green Mile)	(Gladiator)	0.2	0.7	0.1	0.500000	0.714286	-0.04
15	(LOTR1)	(Harry Potter1)	0.2	0.2	0.1	0.500000	2.500000	0.06
16	(Harry Potter1)	(LOTR1)	0.2	0.2	0.1	0.500000	2.500000	0.06
17	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.16
18	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.16
19	(Green Mile)	(LOTR1)	0.2	0.2	0.1	0.500000	2.500000	0.06
20	(LOTR1)	(Green Mile)	0.2	0.2	0.1	0.500000	2.500000	0.06
21	(LOTR2)	(Harry Potter1)	0.2	0.2	0.1	0.500000	2.500000	0.06
22	(Harry Potter1)	(LOTR2)	0.2	0.2	0.1	0.500000	2.500000	0.06
23	(Harry Potter2)	(Harry Potter1)	0.1	0.2	0.1	1.000000	5.000000	0.08
24	(Harry Potter1)	(Harry Potter2)	0.2	0.1	0.1	0.500000	5.000000	0.08
25	(Green Mile)	(Harry Potter1)	0.2	0.2	0.1	0.500000	2.500000	0.06

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
26	(Harry Potter1)	(Green Mile)	0.2	0.2	0.1	0.500000	2.500000	0.06
27	(Braveheart)	(Patriot)	0.1	0.6	0.1	1.000000	1.666667	0.04
28	(Green Mile)	(LOTR2)	0.2	0.2	0.1	0.500000	2.500000	0.06
29	(LOTR2)	(Green Mile)	0.2	0.2	0.1	0.500000	2.500000	0.06
30	(Green Mile)	(LOTR)	0.2	0.1	0.1	0.500000	5.000000	0.08
31	(LOTR)	(Green Mile)	0.1	0.2	0.1	1.000000	5.000000	0.08

5.c Visualization on Scatter plot

In [20]:
```
plt.figure(figsize=(12,8))
plt.scatter(Association_Rules_1['support'], Association_Rules_1['confidence'])
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.show()
```



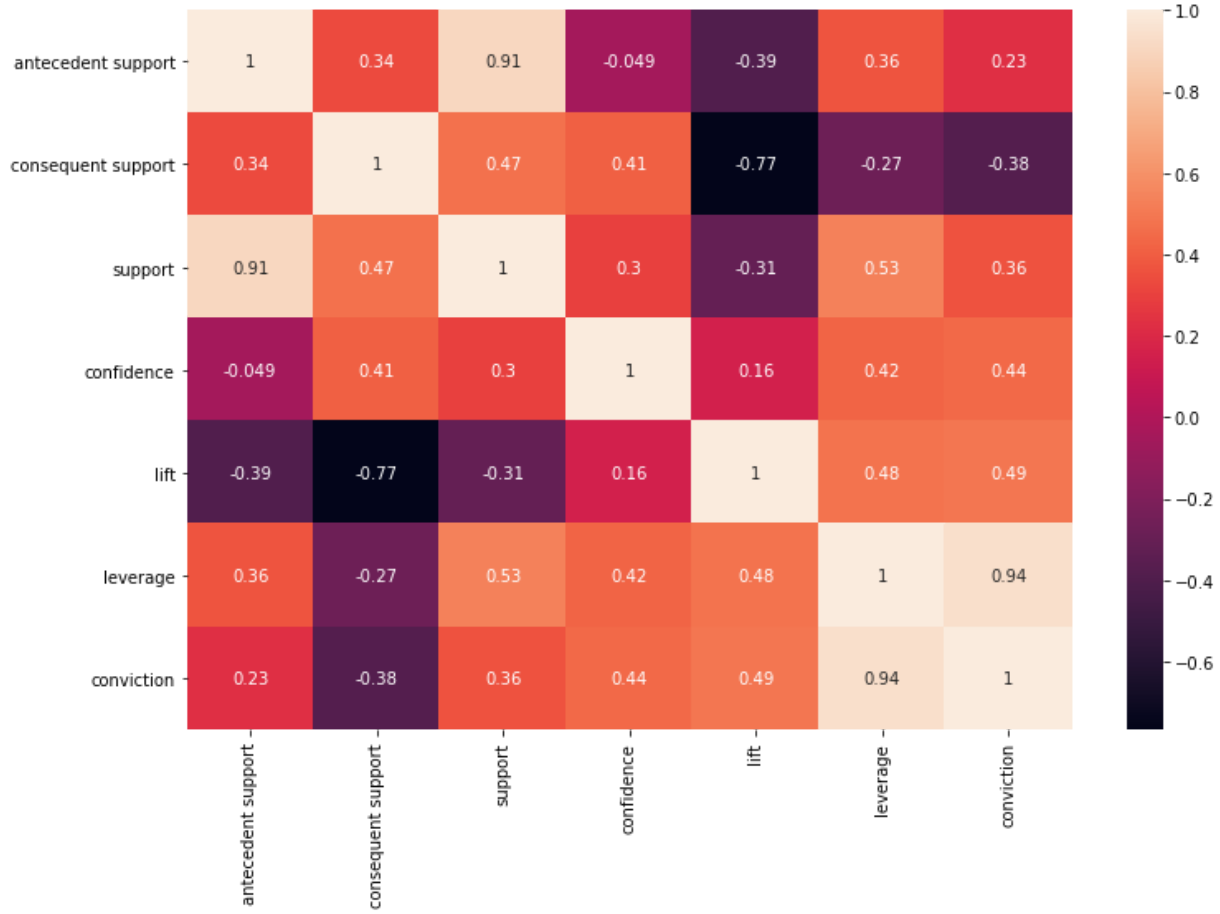
In [23]:
```
corr_Association_rule_1 = Association_Rules_1.corr()
corr_Association_rule_1
```

Out[23]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	0.337123	0.908569	-0.048942	-0.388227	0.364947	0.232475
consequent support	0.337123	1.000000	0.473316	0.408581	-0.767369	-0.274424	-0.377459
support	0.908569	0.473316	1.000000	0.299406	-0.310810	0.531609	0.363809
confidence	-0.048942	0.408581	0.299406	1.000000	0.156823	0.423796	0.441726
lift	-0.388227	-0.767369	-0.310810	0.156823	1.000000	0.480895	0.491092
leverage	0.364947	-0.274424	0.531609	0.423796	0.480895	1.000000	0.943971
conviction	0.232475	-0.377459	0.363809	0.441726	0.491092	0.943971	1.000000

5.d Visualization on Heatmap

In [26]:
```
plt.figure(figsize=(12,8))
sns.heatmap(corr_Association_rule_1, annot = True)
plt.show()
```



6. Value of support '10%'

In [28]:
```
frequent_items_1 = apriori(df = My_Movies, min_support=0.10, use_colnames=True,ma
frequent_items_1
```

Out[28]:

	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	(LOTR1, Sixth Sense)
12	0.1	(Sixth Sense, Harry Potter1)
13	0.4	(Patriot, Sixth Sense)
14	0.1	(LOTR2, Sixth Sense)
15	0.1	(LOTR, Sixth Sense)
16	0.2	(Green Mile, Sixth Sense)
17	0.6	(Patriot, Gladiator)
18	0.1	(LOTR, Gladiator)
19	0.1	(Braveheart, Gladiator)
20	0.1	(Green Mile, Gladiator)
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	(Patriot, Gladiator, Sixth Sense)
31	0.1	(LOTR, Gladiator, Sixth Sense)
32	0.1	(Green Mile, Gladiator, Sixth Sense)

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

6.a Confidence '50%'

In [30]:
```
Association_Rules_2 = association_rules(df = frequent_items_1, metric='confidence
Association_Rules_2
```

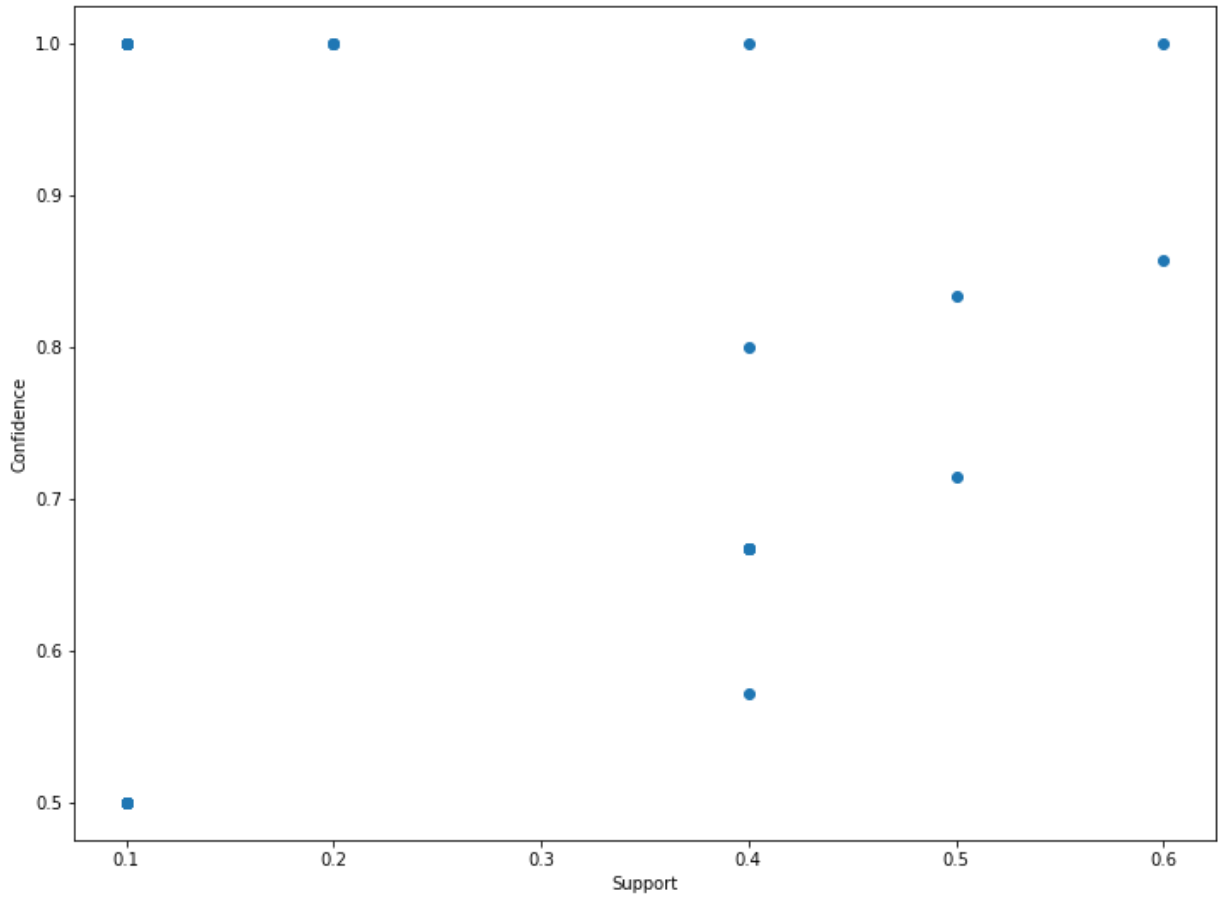
Out[30]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	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	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04
...	...	...	...	...	...	...	...	...
182	(LOTR2, Harry Potter1)	(Green Mile, LOTR1)	0.1	0.1	0.1	1.000000	10.000000	0.09
183	(Green Mile)	(LOTR1, LOTR2, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08
184	(LOTR1)	(Green Mile, LOTR2, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08
185	(LOTR2)	(Green Mile, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08
186	(Harry Potter1)	(Green Mile, LOTR1, LOTR2)	0.2	0.1	0.1	0.500000	5.000000	0.08

187 rows × 9 columns

6.b Visualization on Scatter plot

In [39]:
```
plt.figure(figsize=(12,9))
plt.scatter(Association_Rules_2['support'], Association_Rules_2['confidence'])
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.show()
```



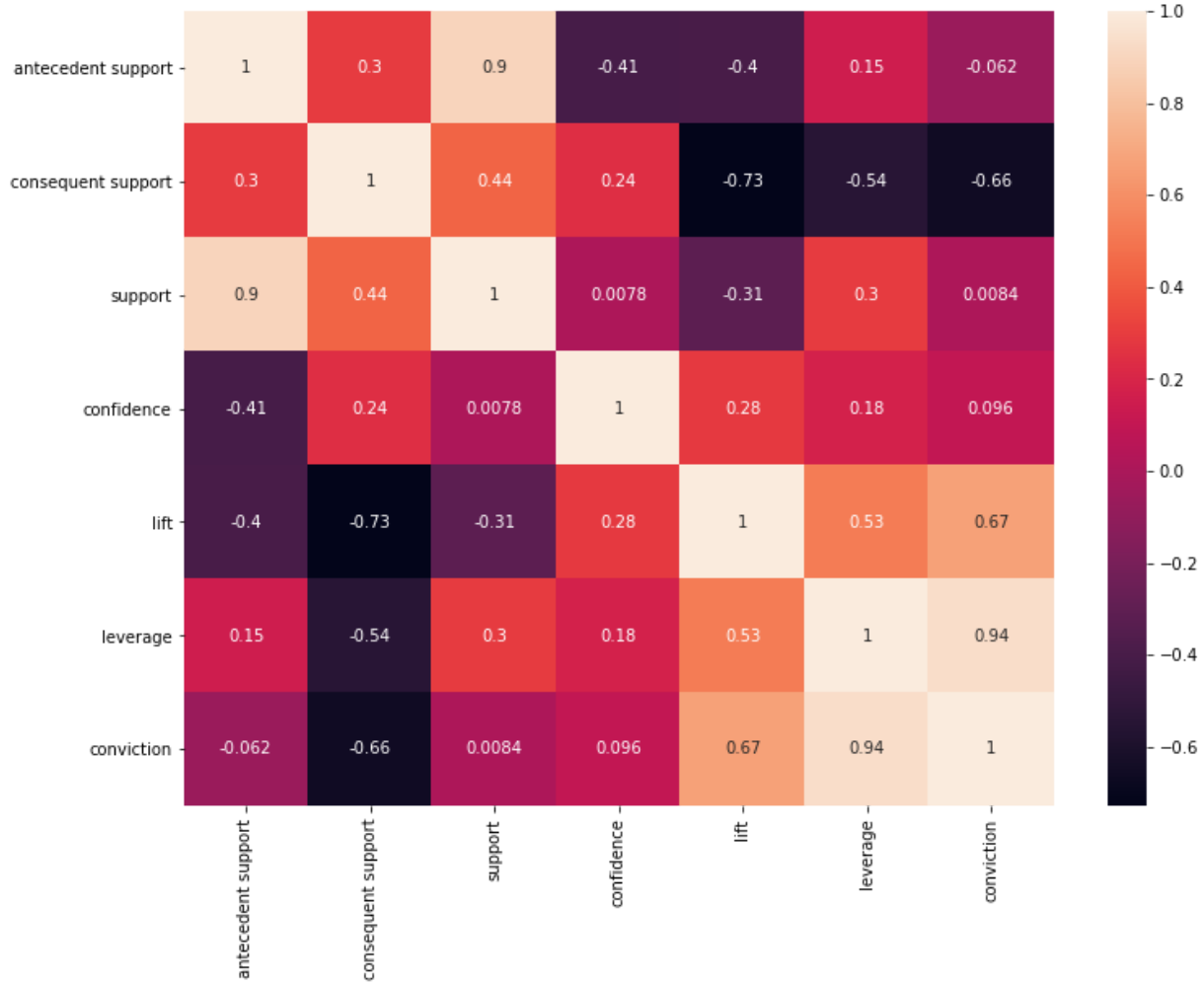
In [40]:
```
corr_Association_rule_2 = Association_Rules_2.corr()
corr_Association_rule_2
```

Out[40]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	0.296027	0.896898	-0.409066	-0.402957	0.153684	-0.061535
consequent support	0.296027	1.000000	0.436982	0.241788	-0.728485	-0.540219	-0.659205
support	0.896898	0.436982	1.000000	0.007757	-0.314632	0.295879	0.008394
confidence	-0.409066	0.241788	0.007757	1.000000	0.284649	0.182584	0.096485
lift	-0.402957	-0.728485	-0.314632	0.284649	1.000000	0.525149	0.670031
leverage	0.153684	-0.540219	0.295879	0.182584	0.525149	1.000000	0.936143
conviction	-0.061535	-0.659205	0.008394	0.096485	0.670031	0.936143	1.000000

6.c Visualization Heatmap

In [43]:
```
plt.figure(figsize=(12,9))
sns.heatmap(corr_Association_rule_2, annot = True)
plt.show()
```



7. Value of support '15%'

In [45]:
```
frequent_items_2 = apriori(df = My_Movies, min_support=0.15, use_colnames=True, m
frequent_items_2
```

Out[45]:

	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	(Gladiator, Sixth Sense)
8	0.4	(Patriot, Sixth Sense)
9	0.2	(Green Mile, Sixth Sense)
10	0.6	(Patriot, Gladiator)
11	0.2	(LOTR1, LOTR2)
12	0.4	(Patriot, Gladiator, Sixth Sense)

7.a Confidence '70%'

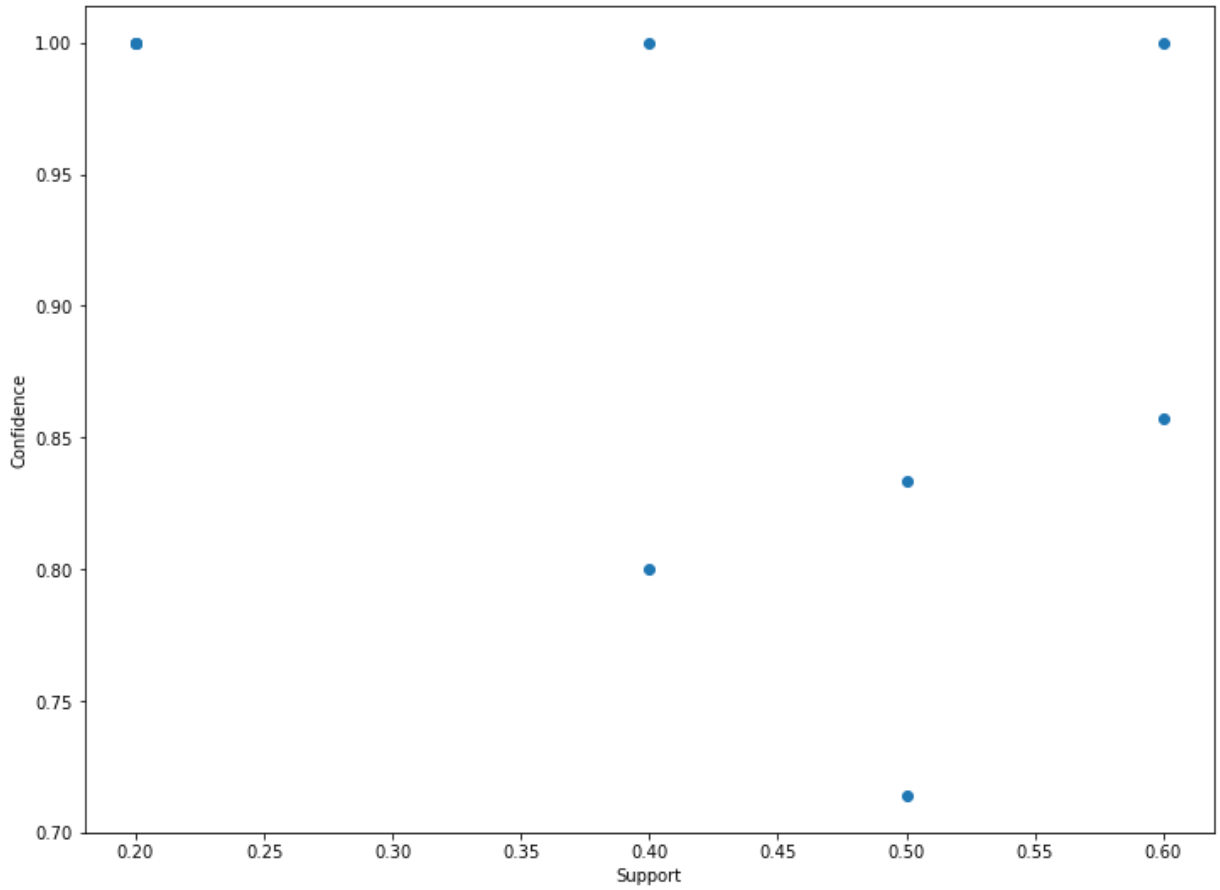
In [47]:
```
Association_Rules_3 = association_rules(df = frequent_items_2, metric='confidence
Association_Rules_3
```

Out[47]:

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

7.b Visualization on Scatter plot

In [50]:
```
plt.figure(figsize=(12,9))
plt.scatter(Association_Rules_3['support'], Association_Rules_3['confidence'])
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.show()
```



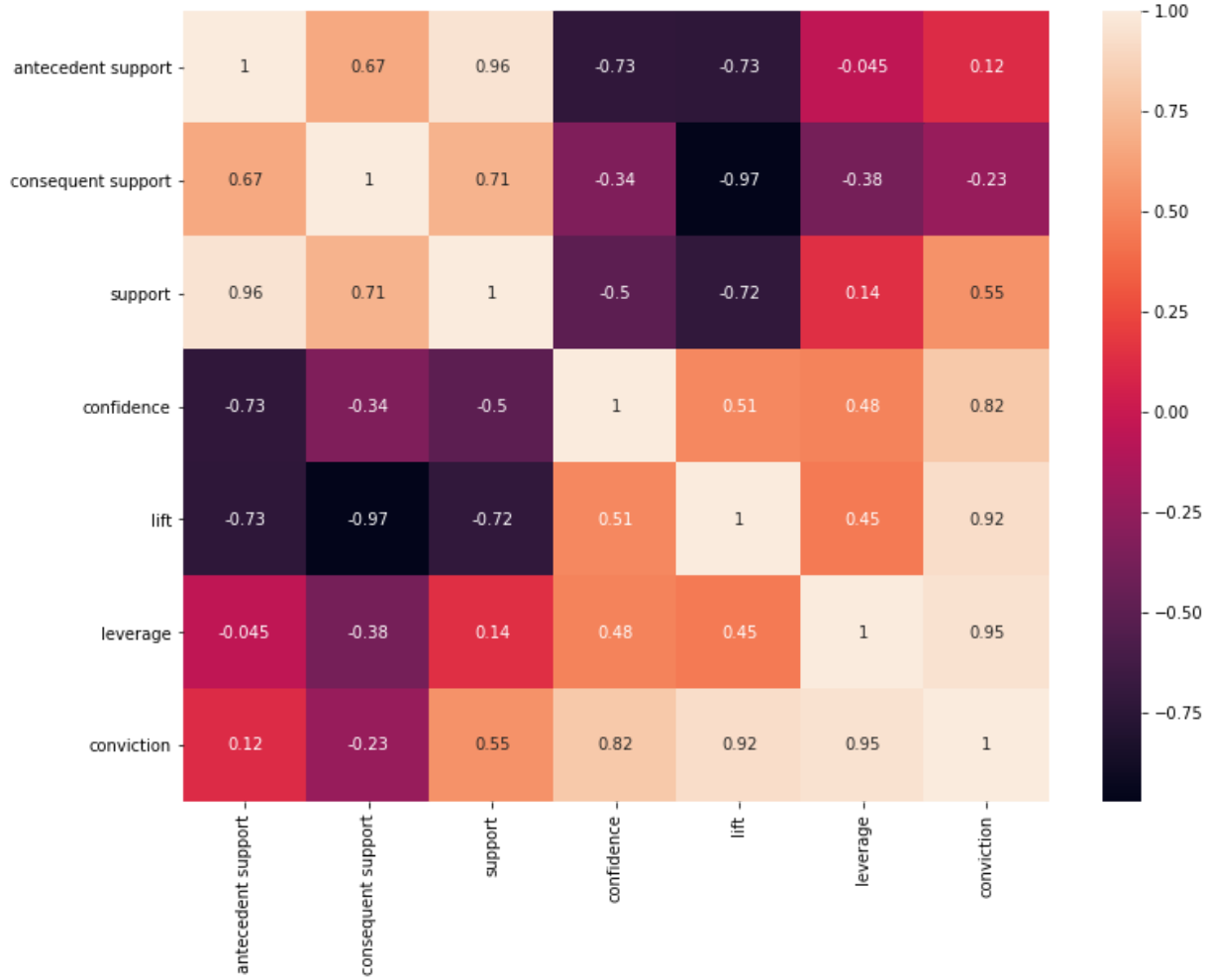
In [51]:
```
corr_Association_rule_3 = Association_Rules_3.corr()
corr_Association_rule_3
```

Out[51]:

	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

7.c Visualization on Heatmap

In [54]:
```
plt.figure(figsize=(12,9))
sns.heatmap(corr_Association_rule_3, annot = True)
plt.show()
```



Conclusion : -

Different values of support and confidences are choosen & Visualized on Scatter plot & Heat map