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

```
ite-packages (from mlxtend) (52.0.0.post20210125)
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: 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: joblib>=0.13.2 in c:\users\nandini\anaconda3\l
ib\site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\nandini\anaco
nda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\nandini\anaconda3\li
b\site-packages (from matplotlib>=3.0.0->mlxtend) (8.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\nandini\anaconda3\lib
\site-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
c:\users\nandini\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxten
```

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 [2]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import transactionencoder
```

2. Import Data

In [3]:
```
Book_Data  = pd.read_csv('book.csv')
Book_Data
```

Out[3]:

	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

3. Data Understanding

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

Out[4]:
```
(2000, 11)
```

In [5]:
```
Book_Data.dtypes
```

Out[5]:
```
ChildBks     int64
YouthBks     int64
CookBks      int64
DoItYBks     int64
RefBks       int64
ArtBks       int64
GeogBks      int64
ItalCook     int64
ItalAtlas    int64
ItalArt      int64
Florence     int64
dtype: object
```

In [6]:
```
Book_Data.isna().sum()
```

Out[6]:
```
ChildBks     0
YouthBks     0
CookBks      0
DoItYBks     0
RefBks       0
ArtBks       0
GeogBks      0
ItalCook     0
ItalAtlas    0
ItalArt      0
Florence     0
dtype: int64
```

In [7]:
```
Book_Data.describe(include='all').nunique()
```

Out[7]:
```
ChildBks     5
YouthBks     5
CookBks      5
DoItYBks     5
RefBks       5
ArtBks       5
GeogBks      5
ItalCook     5
ItalAtlas    5
ItalArt      5
Florence     5
dtype: int64
```

In [8]:
```
Book_Data.head(20)
```

6	0	1	0	0	0	0	0	0	0
7	0	1	0	0	1	0	0	0	0
8	1	0	0	1	0	0	0	0	0
9	1	1	1	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0
11	0	0	1	0	0	0	1	0	0
12	1	0	0	0	0	1	0	0	0
13	1	1	0	1	1	1	0	0	1
14	1	1	1	0	0	0	0	0	0
15	1	1	1	0	0	0	1	0	0
16	0	0	1	0	0	0	0	0	0

In [9]:
```
Book_Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ChildBks   2000 non-null   int64
 1   YouthBks   2000 non-null   int64
 2   CookBks    2000 non-null   int64
 3   DoItYBks   2000 non-null   int64
 4   RefBks     2000 non-null   int64
 5   ArtBks     2000 non-null   int64
 6   GeogBks    2000 non-null   int64
 7   ItalCook   2000 non-null   int64
 8   ItalAtlas  2000 non-null   int64
 9   ItalArt    2000 non-null   int64
 10  Florence   2000 non-null   int64
dtypes: int64(11)
memory usage: 172.0 KB
```

1. Value of support 5%

In [57]:
```
frequent_Items = apriori(df = Book_Data, min_support=0.05, use_colnames=True, max
frequent_Items
```

18	0.0965	(YouthBks, RefBks)
19	0.1010	(YouthBks, ArtBks)
20	0.1205	(YouthBks, GeogBks)
21	0.0590	(YouthBks, ItalCook)
22	0.1875	(DoItYBks, CookBks)
23	0.1525	(RefBks, CookBks)
24	0.1670	(CookBks, ArtBks)
25	0.1925	(GeogBks, CookBks)
26	0.1135	(ItalCook, CookBks)
27	0.1055	(RefBks, DoItYBks)
28	0.1235	(DoItYBks, ArtBks)
29	0.1325	(DoItYBks, GeogBks)
30	0.0585	(ItalCook, DoItYBks)

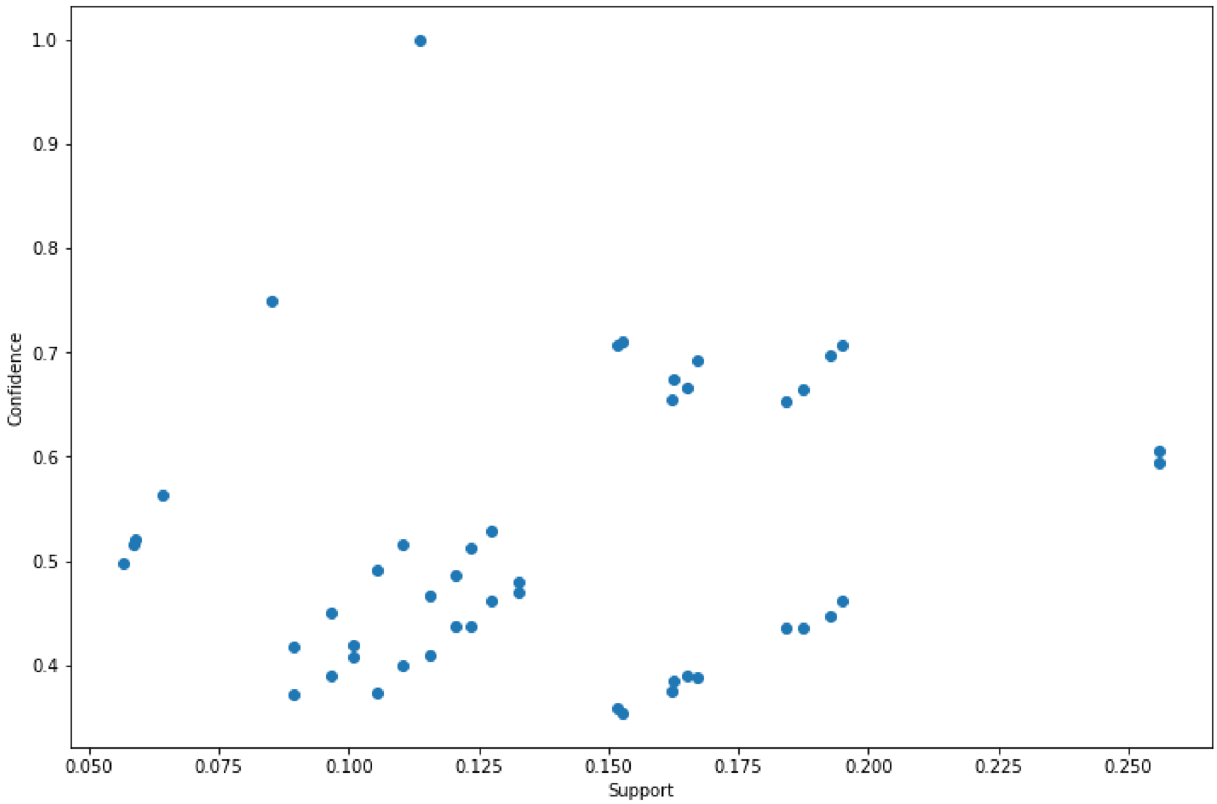
1.a For 30% Confidence

In [59]:
```
Association_Rule_1 = association_rules(df = frequent_Items, metric='confidence',
Association_Rule_1
```

28	(CookBks)	(ArtBks)	0.4310	0.2410	0.1670	0.387471	1.607763	0.063129
29	(ArtBks)	(CookBks)	0.2410	0.4310	0.1670	0.692946	1.607763	0.063129
30	(GeogBks)	(CookBks)	0.2760	0.4310	0.1925	0.697464	1.618245	0.073544
31	(CookBks)	(GeogBks)	0.4310	0.2760	0.1925	0.446636	1.618245	0.073544
32	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582
33	(RefBks)	(DoItYBks)	0.2145	0.2820	0.1055	0.491841	1.744119	0.045011
34	(DoItYBks)	(RefBks)	0.2820	0.2145	0.1055	0.374113	1.744119	0.045011
35	(DoItYBks)	(ArtBks)	0.2820	0.2410	0.1235	0.437943	1.817192	0.055538
36	(ArtBks)	(DoItYBks)	0.2410	0.2820	0.1235	0.512448	1.817192	0.055538
37	(DoItYBks)	(GeogBks)	0.2820	0.2760	0.1325	0.469858	1.702385	0.054668
38	(GeogBks)	(DoItYBks)	0.2760	0.2820	0.1325	0.480072	1.702385	0.054668

1.b Visualization on Scatter plot

In [60]:
```
plt.figure(figsize=(12,8))
plt.scatter(Association_Rule_1['support'], Association_Rule_1['confidence'])
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.show()
```



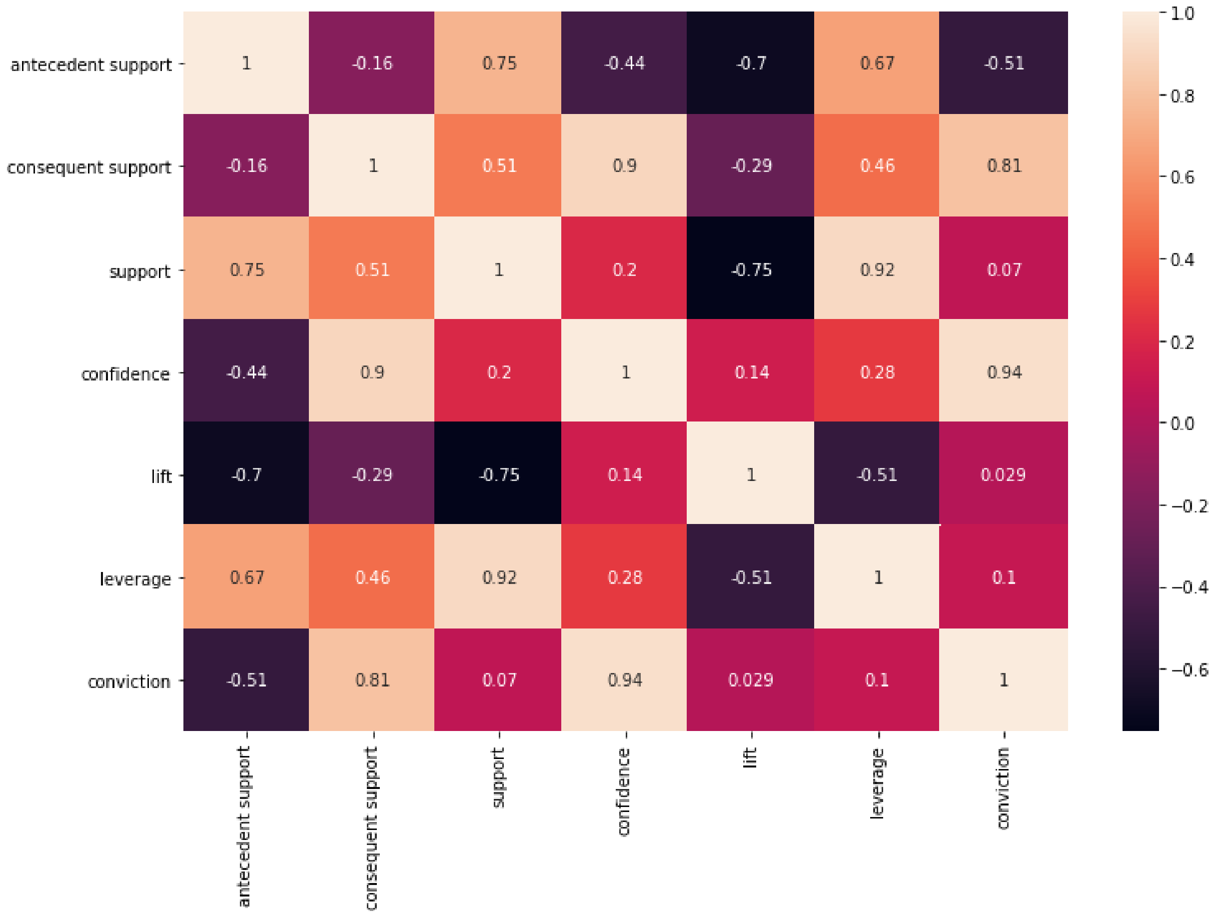
In [61]:
```
corr_Association_Rule_1 = Association_Rule_1.corr()
corr_Association_Rule_1
```

Out[61]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	-0.158455	0.747537	-0.444987	-0.697670	0.666032	-0.512398
consequent support	-0.158455	1.000000	0.509299	0.900647	-0.291025	0.464148	0.811079
support	0.747537	0.509299	1.000000	0.202833	-0.753486	0.916341	0.069865
confidence	-0.444987	0.900647	0.202833	1.000000	0.140217	0.279391	0.942153
lift	-0.697670	-0.291025	-0.753486	0.140217	1.000000	-0.510409	0.028533
leverage	0.666032	0.464148	0.916341	0.279391	-0.510409	1.000000	0.100157
conviction	-0.512398	0.811079	0.069865	0.942153	0.028533	0.100157	1.000000

1.c Visualization On Heatmap

In [62]:
```
plt.figure(figsize=(12,8))
sns.heatmap(data = corr_Association_Rule_1,annot=True )
plt.show()
```



2. Value of Support '10%'

In [63]:
```
frequent_Items_1 = apriori(df = Book_Data, min_support=0.10, use_colnames=True, r
frequent_Items_1
```

Out[63]:

	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	(CookBks, ArtBks)
22	0.1925	(GeogBks, CookBks)
23	0.1135	(ItalCook, CookBks)
24	0.1055	(RefBks, DoItYBks)
25	0.1235	(DoItYBks, ArtBks)
26	0.1325	(DoItYBks, GeogBks)
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)
32	0.1265	(ChildBks, CookBks, ArtBks)

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

2.a Confidence with min threshold 50%

In [64]:
```
Association_Rule_2 = association_rules(df = frequent_Items_1, metric='confidence'
Association_Rule_2
```

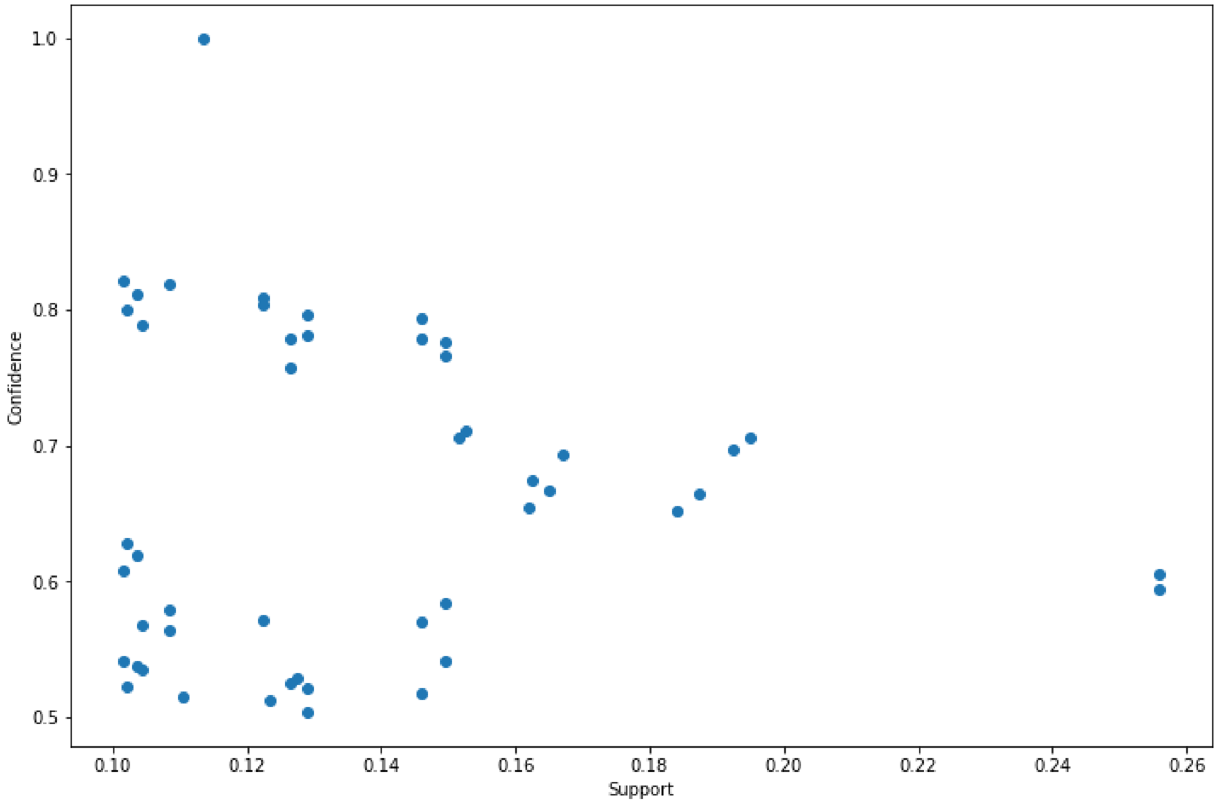
Out[64]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
0	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308
1	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687
2	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687
3	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.064714
4	(RefBks)	(ChildBks)	0.2145	0.4230	0.1515	0.706294	1.669725	0.060767
5	(ArtBks)	(ChildBks)	0.2410	0.4230	0.1625	0.674274	1.594028	0.060557
6	(GeogBks)	(ChildBks)	0.2760	0.4230	0.1950	0.706522	1.670264	0.078252
7	(YouthBks)	(CookBks)	0.2475	0.4310	0.1620	0.654545	1.518667	0.055328
8	(DoItYBks)	(CookBks)	0.2820	0.4310	0.1875	0.664894	1.542677	0.065958
9	(RefBks)	(CookBks)	0.2145	0.4310	0.1525	0.710956	1.649549	0.060050
10	(ArtBks)	(CookBks)	0.2410	0.4310	0.1670	0.692946	1.607763	0.063129
11	(GeogBks)	(CookBks)	0.2760	0.4310	0.1925	0.697464	1.618245	0.073544
12	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582
13	(ArtBks)	(DoItYBks)	0.2410	0.2820	0.1235	0.512448	1.817192	0.055538
14	(RefBks)	(GeogBks)	0.2145	0.2760	0.1105	0.515152	1.866491	0.051298
15	(ArtBks)	(GeogBks)	0.2410	0.2760	0.1275	0.529046	1.916832	0.060984
16	(ChildBks, YouthBks)	(CookBks)	0.1650	0.4310	0.1290	0.781818	1.813963	0.057885
17	(ChildBks, CookBks)	(YouthBks)	0.2560	0.2475	0.1290	0.503906	2.035985	0.065640
18	(YouthBks, CookBks)	(ChildBks)	0.1620	0.4230	0.1290	0.796296	1.882497	0.060474
19	(YouthBks)	(ChildBks, CookBks)	0.2475	0.2560	0.1290	0.521212	2.035985	0.065640
20	(ChildBks, DoItYBks)	(CookBks)	0.1840	0.4310	0.1460	0.793478	1.841017	0.066696
21	(ChildBks, CookBks)	(DoItYBks)	0.2560	0.2820	0.1460	0.570312	2.022385	0.073808
22	(DoItYBks, CookBks)	(ChildBks)	0.1875	0.4230	0.1460	0.778667	1.840820	0.066687
23	(DoItYBks)	(ChildBks, CookBks)	0.2820	0.2560	0.1460	0.517730	2.022385	0.073808
24	(ChildBks, RefBks)	(CookBks)	0.1515	0.4310	0.1225	0.808581	1.876058	0.057204
25	(RefBks, CookBks)	(ChildBks)	0.1525	0.4230	0.1225	0.803279	1.899004	0.057993
26	(RefBks)	(ChildBks, CookBks)	0.2145	0.2560	0.1225	0.571096	2.230842	0.067588

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
27	(ChildBks, ArtBks)	(CookBks)	0.1625	0.4310	0.1265	0.778462	1.806175	0.056462
28	(CookBks, ArtBks)	(ChildBks)	0.1670	0.4230	0.1265	0.757485	1.790745	0.055859
29	(ArtBks)	(ChildBks, CookBks)	0.2410	0.2560	0.1265	0.524896	2.050376	0.064804
30	(ChildBks, GeogBks)	(CookBks)	0.1950	0.4310	0.1495	0.766667	1.778809	0.065455
31	(ChildBks, CookBks)	(GeogBks)	0.2560	0.2760	0.1495	0.583984	2.115885	0.078844
32	(GeogBks, CookBks)	(ChildBks)	0.1925	0.4230	0.1495	0.776623	1.835989	0.068072
33	(GeogBks)	(ChildBks, CookBks)	0.2760	0.2560	0.1495	0.541667	2.115885	0.078844
34	(ChildBks, DoItYBks)	(GeogBks)	0.1840	0.2760	0.1045	0.567935	2.057735	0.053716
35	(ChildBks, GeogBks)	(DoItYBks)	0.1950	0.2820	0.1045	0.535897	1.900346	0.049510
36	(DoItYBks, GeogBks)	(ChildBks)	0.1325	0.4230	0.1045	0.788679	1.864490	0.048452
37	(ChildBks, GeogBks)	(ArtBks)	0.1950	0.2410	0.1020	0.523077	2.170444	0.055005
38	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150
39	(GeogBks, ArtBks)	(ChildBks)	0.1275	0.4230	0.1020	0.800000	1.891253	0.048067
40	(DoItYBks, CookBks)	(ArtBks)	0.1875	0.2410	0.1015	0.541333	2.246196	0.056313
41	(DoItYBks, ArtBks)	(CookBks)	0.1235	0.4310	0.1015	0.821862	1.906873	0.048272
42	(CookBks, ArtBks)	(DoItYBks)	0.1670	0.2820	0.1015	0.607784	2.155264	0.054406
43	(DoItYBks, GeogBks)	(CookBks)	0.1325	0.4310	0.1085	0.818868	1.899926	0.051392
44	(GeogBks, CookBks)	(DoItYBks)	0.1925	0.2820	0.1085	0.563636	1.998711	0.054215
45	(DoItYBks, CookBks)	(GeogBks)	0.1875	0.2760	0.1085	0.578667	2.096618	0.056750
46	(GeogBks, CookBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107
47	(GeogBks, ArtBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547
48	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057408

2.b Visualization on scatter plot

In [65]:
```
plt.figure(figsize=(12,8))
plt.scatter(Association_Rule_2['support'], Association_Rule_2['confidence'] )
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.show()
```



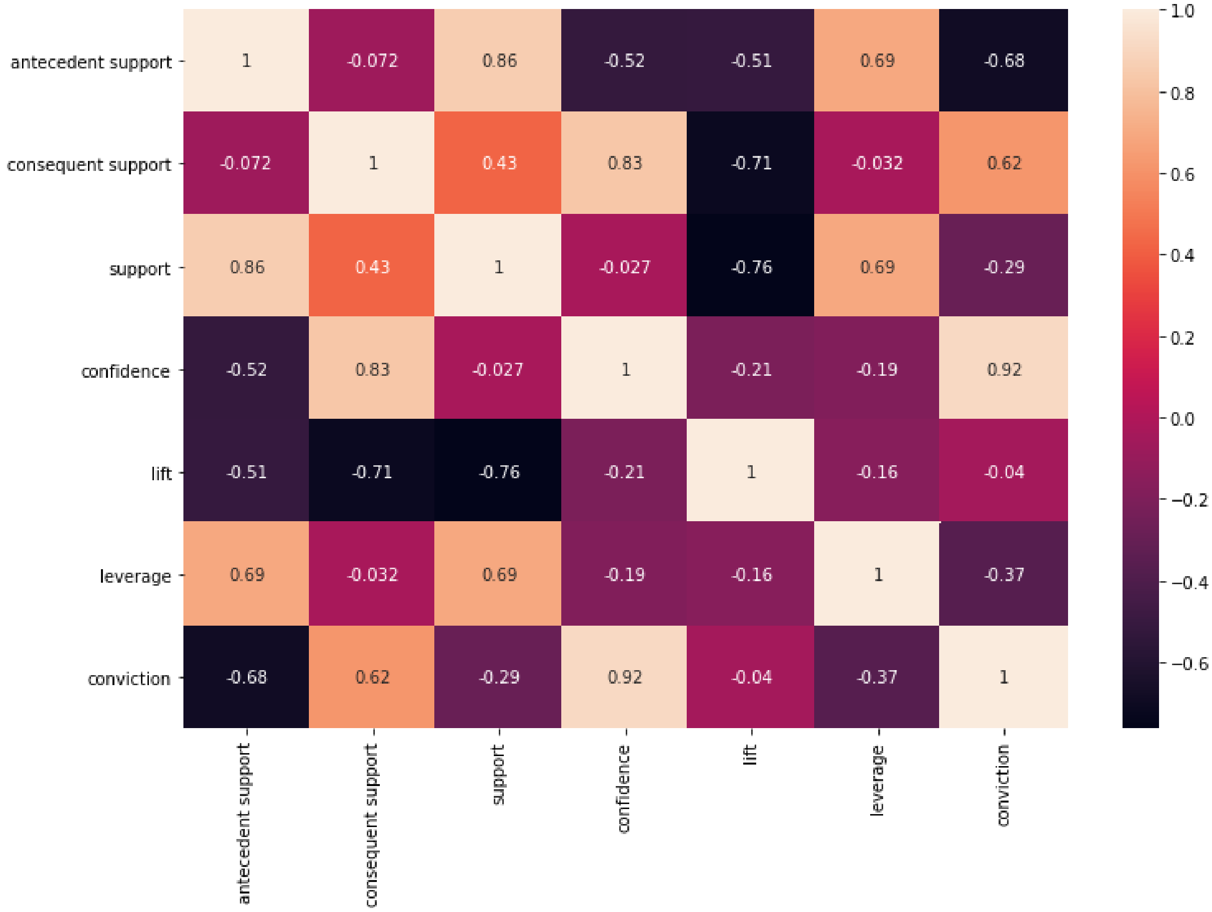
In [66]:
```
corr_Association_Rule_2=Association_Rule_2.corr()
corr_Association_Rule_2
```

Out[66]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	-0.071635	0.858437	-0.519173	-0.512713	0.690605	-0.683617
consequent support	-0.071635	1.000000	0.426340	0.831627	-0.711482	-0.031692	0.621042
support	0.858437	0.426340	1.000000	-0.027358	-0.762058	0.692109	-0.291754
confidence	-0.519173	0.831627	-0.027358	1.000000	-0.208019	-0.185236	0.917344
lift	-0.512713	-0.711482	-0.762058	-0.208019	1.000000	-0.157036	-0.040188
leverage	0.690605	-0.031692	0.692109	-0.185236	-0.157036	1.000000	-0.365942
conviction	-0.683617	0.621042	-0.291754	0.917344	-0.040188	-0.365942	1.000000

2.c Visualization on heatmap

In [67]:
```
plt.figure(figsize=(12,8))
sns.heatmap(corr_Association_Rule_2, annot=True)
plt.show()
```



3. Value of support '15%'

In [70]:
```
frequent_Items_2 = apriori(df = Book_Data, min_support=0.15, use_colnames=True, r
frequent_Items_2
```

Out[70]:

	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	0.1650	(ChildBks, YouthBks)
8	0.2560	(ChildBks, CookBks)
9	0.1840	(ChildBks, DoItYBks)
10	0.1515	(ChildBks, RefBks)
11	0.1625	(ChildBks, ArtBks)
12	0.1950	(ChildBks, GeogBks)
13	0.1620	(YouthBks, CookBks)
14	0.1875	(DoItYBks, CookBks)
15	0.1525	(RefBks, CookBks)
16	0.1670	(CookBks, ArtBks)
17	0.1925	(GeogBks, CookBks)

3.a Confidence of '70%'

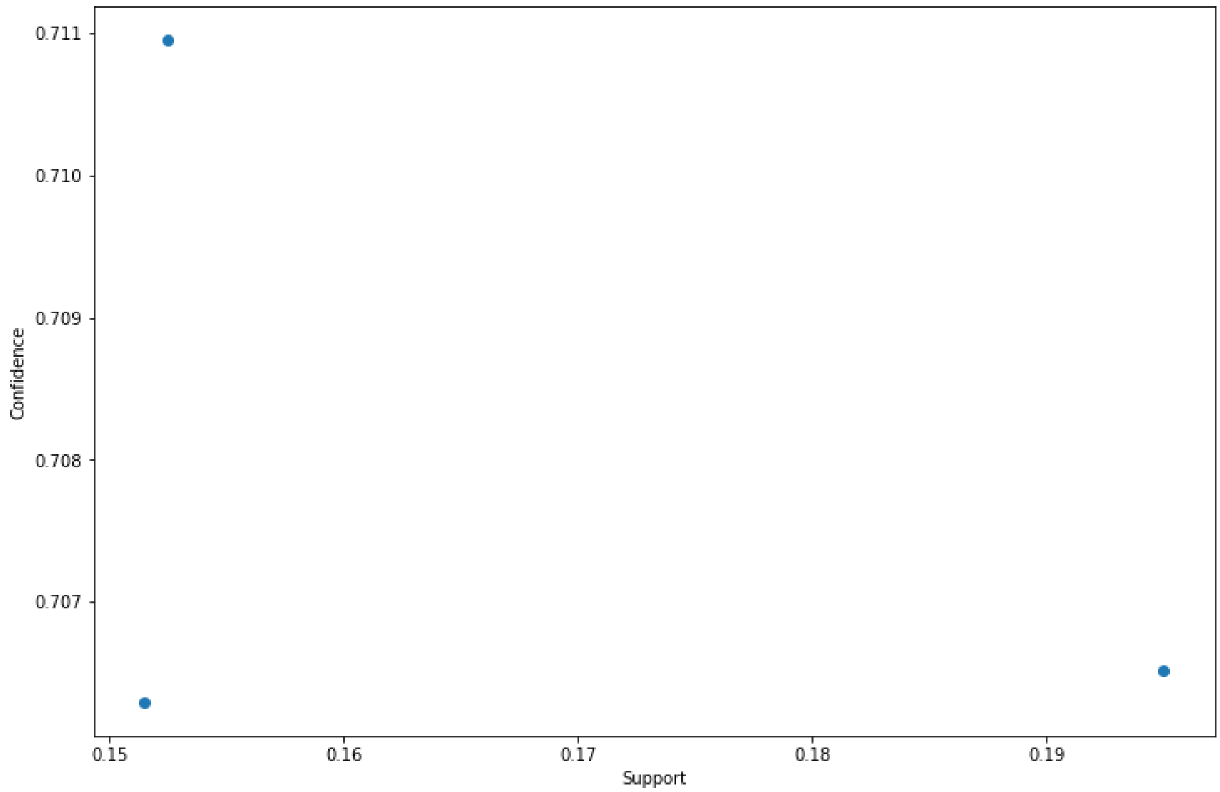
In [71]:
```
Association_Rule_3 = association_rules(df = frequent_Items_2, metric='confidence'
Association_Rule_3
```

Out[71]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	c
0	(RefBks)	(ChildBks)	0.2145	0.423	0.1515	0.706294	1.669725	0.060767	
1	(GeogBks)	(ChildBks)	0.2760	0.423	0.1950	0.706522	1.670264	0.078252	
2	(RefBks)	(CookBks)	0.2145	0.431	0.1525	0.710956	1.649549	0.060050	

3.b Visualization on Scatter plot

In [72]:
```
plt.figure(figsize=(12,8))
plt.scatter(Association_Rule_3['support'], Association_Rule_3['confidence'])
plt.xlabel('Support')
plt.ylabel('Confidence')
plt.show()
```



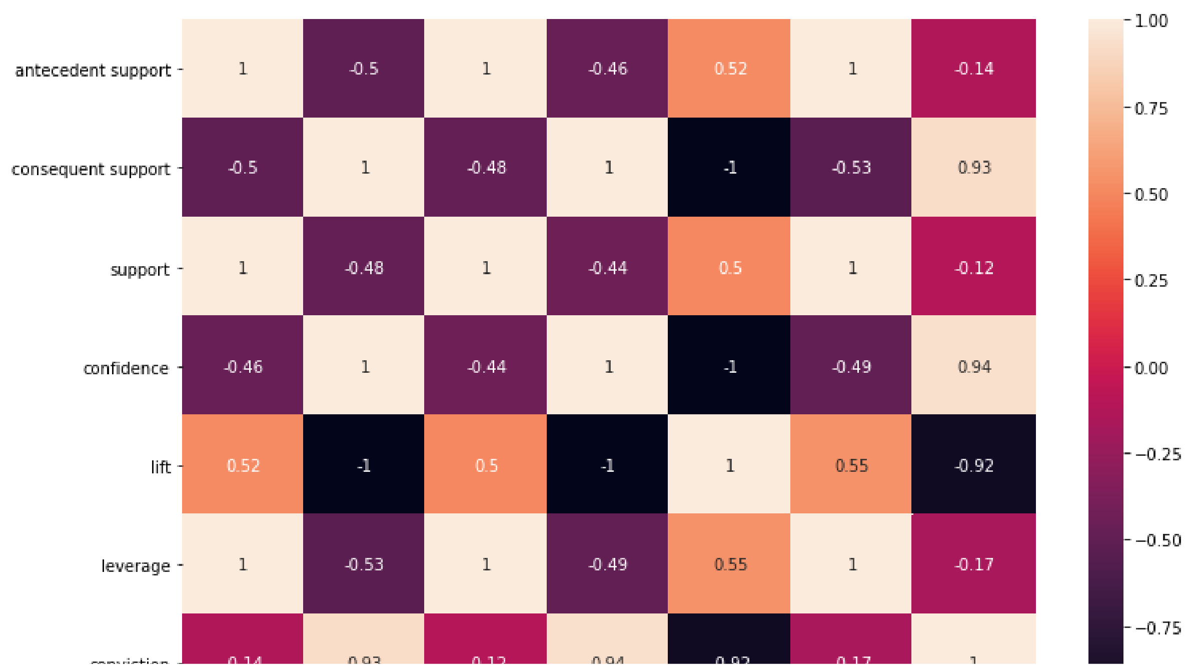
In [73]:
```
corr_Association_Rule_3 = Association_Rule_3.corr()
corr_Association_Rule_3
```

Out[73]:

	antecedent support	consequent support	support	confidence	lift	leverage	conviction
antecedent support	1.000000	-0.500000	0.999797	-0.461960	0.519640	0.999397	-0.136385
consequent support	-0.500000	1.000000	-0.482460	0.999059	-0.999739	-0.529775	0.926126
support	0.999797	-0.482460	1.000000	-0.444008	0.502331	0.998495	-0.116410
confidence	-0.461960	0.999059	-0.444008	1.000000	-0.997808	-0.492483	0.941618
lift	0.519640	-0.999739	0.502331	-0.997808	1.000000	0.548999	-0.917273
leverage	0.999397	-0.529775	0.998495	-0.492483	0.548999	1.000000	-0.170708
conviction	-0.136385	0.926126	-0.116410	0.941618	-0.917273	-0.170708	1.000000

3.c Visualization on Heatmap

In [74]:
```
plt.figure(figsize=(12,8))
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