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

Association Rules:

What goes with what..

If(antecedents) -Then(concequents)

In [1]:

```
%pip install networkx
```

```
Requirement already satisfied: networkx in c:\users\rajesh\anaconda3\lib\site-packages (2.6.3)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:

```
pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\rajesh\anaconda3\lib\site-packages (0.19.0)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend)
(0.24.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend) (1.19.
5)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend) (3
.4.3)
Requirement already satisfied: setuptools in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend) (58.0.4)
Requirement already satisfied: scipy>=1.2.1 in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend) (1.7.1)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pandas>=0.24.2 in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend) (1.3.
4)
Requirement already satisfied: joblib>=0.13.2 in c:\users\rajesh\anaconda3\lib\site-packages (from mlxtend) (1.0.
1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\rajesh\anaconda3\lib\site-packages (from matplotlib>=
3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\rajesh\anaconda3\lib\site-packages (from matplotlib>
=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\rajesh\anaconda3\lib\site-packages (from matplotlib>=3.0.
0->mlxtend) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\rajesh\anaconda3\lib\site-packages (from matplotlib>=3.0
.0->mlxtend) (8.4.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\rajesh\anaconda3\lib\site-packages (from matplotl
ib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: six in c:\users\rajesh\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\rajesh\anaconda3\lib\site-packages (from pandas>=0.24.2->
mlxtend) (2021.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\rajesh\anaconda3\lib\site-packages (from scikit-l
earn>=0.20.3->mlxtend) (2.2.0)
```

In [3]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.frequent_patterns import apriori,association_rules
```

In [4]:

```
books_data=pd.read_csv('book (1).csv')
books_data
```

Out[4]:

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

...	...	...	...	...	...	...	...	...	...	...	...
1995	0	0	1	0	0	1	1	1	0	1	1
1996	0	0	0	0	0	0	0	0	0	0	0
1997	0	0	0	0	0	0	0	0	0	0	0
1998	0	0	1	0	0	0	0	0	0	0	0
1999	0	0	0	0	0	0	0	0	0	0	0

2000 rows × 11 columns

In [5]:

```
#Let us consider The one who buy a book='1' , The one who didn't buy a book ='0'
```

In [6]:

```
books_data.shape
```

Out[6]:

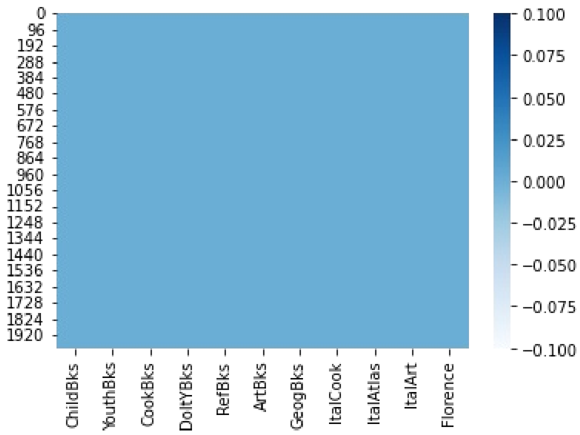
```
(2000, 11)
```

In [7]:

```
sns.heatmap(books_data.isnull(),cmap='Blues')
```

Out[7]:

```
<AxesSubplot:>
```



In [8]:

```
books_data.dtypes
```

Out[8]:

```
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 [9]:

```
books_data.describe().round(2).style.background_gradient(cmap = 'Oranges')
```

Out[9]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	ItalArt	
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	200
mean	0.420000	0.250000	0.430000	0.280000	0.210000	0.240000	0.280000	0.110000	0.040000	0.050000	
std	0.490000	0.430000	0.500000	0.450000	0.410000	0.430000	0.450000	0.320000	0.190000	0.210000	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	1.000000	0.000000	1.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

NOTE:

. Lift- High Lift value indicates both the items are associates strongly

. Leverage- A leverage value of 0 indicates independence. Range will be [-1 1]

. Conviction- A high conviction value means that the consequent is highly depending on the antecedent and range [0 inf]

In [10]:
```
freq_items1=apriori(df=books_data, min_support=0.20,use_colnames= True)
freq_items1
```

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)

confidence = 60%

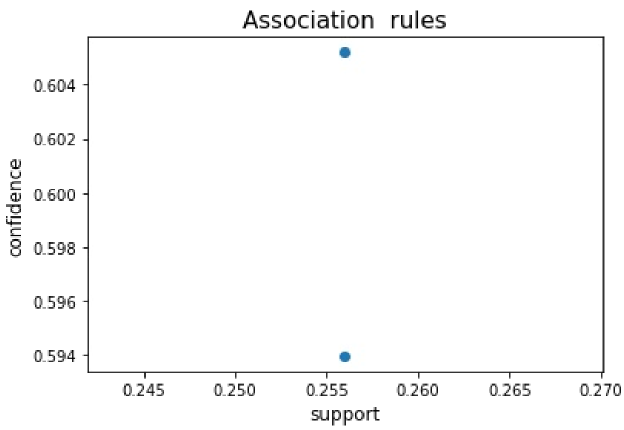
In [11]:
```
rules1=association_rules( df=freq_items1,metric='lift',min_threshold=0.6)
rules1
```

Out[11]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ChildBks)	(CookBks)	0.423	0.431	0.256	0.605201	1.404179	0.073687	1.441240
1	(CookBks)	(ChildBks)	0.431	0.423	0.256	0.593968	1.404179	0.073687	1.421069

. Greater the lift value means stronger the association between the items

In [12]:
```
#visualization of obtained rule
plt.scatter(x='support',y='confidence',data=rules1 )
plt.xlabel('support', size= 12)
plt.ylabel('confidence', size=12)
plt.title('Association  rules',size=15)
plt.show()
```



For Min Support = 0.10

In [13]:
```
freq_items2=apriori(df=books_data, min_support=0.10,use_colnames= True)
```

```
freq_items2
```

Out[13]:

	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	(YouthBks, ChildBks)
10	0.2560	(ChildBks, CookBks)
11	0.1840	(ChildBks, DoItYBks)
12	0.1515	(RefBks, ChildBks)
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	(CookBks, DoItYBks)
20	0.1525	(RefBks, CookBks)
21	0.1670	(ArtBks, CookBks)
22	0.1925	(CookBks, GeogBks)
23	0.1135	(ItalCook, CookBks)
24	0.1055	(RefBks, DoItYBks)
25	0.1235	(ArtBks, DoItYBks)
26	0.1325	(DoItYBks, GeogBks)
27	0.1105	(RefBks, GeogBks)
28	0.1275	(ArtBks, GeogBks)
29	0.1290	(YouthBks, ChildBks, CookBks)
30	0.1460	(ChildBks, CookBks, DoItYBks)
31	0.1225	(RefBks, ChildBks, CookBks)
32	0.1265	(ArtBks, ChildBks, CookBks)
33	0.1495	(ChildBks, CookBks, GeogBks)
34	0.1045	(ChildBks, DoItYBks, GeogBks)
35	0.1020	(ChildBks, ArtBks, GeogBks)
36	0.1015	(ArtBks, CookBks, DoItYBks)
37	0.1085	(GeogBks, CookBks, DoItYBks)
38	0.1035	(ArtBks, CookBks, GeogBks)

In [14]:

```
freq_items2.shape
```

Out[14]:

```
(39, 2)
```

70% confidence

In [15]:

```
rules2=association_rules( df=freq_items2,metric='lift',min_threshold=0.70)
rules2
```

Out[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308	1.731000
1	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308	1.233750

2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714	1.270770
...	...	...	...	...	...	...	...	...	...
95	(ArtBks, GeogBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547	3.022812
96	(CookBks, GeogBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107	1.641657
97	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107	1.415327
98	(CookBks)	(ArtBks, GeogBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547	1.148237
99	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408	1.332800

100 rows × 9 columns

In [16]:

```
rules2.sort_values('lift',ascending=False)
```

Out[16]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
29	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064582	1.203406
28	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582	inf
76	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150	1.944628
81	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150	1.328448
85	(ArtBks)	(CookBks, DoItYBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.056313	1.403674
...	...	...	...	...	...	...	...	...	...
5	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.064714	1.660347
12	(YouthBks)	(CookBks)	0.2475	0.4310	0.1620	0.654545	1.518667	0.055328	1.647105
13	(CookBks)	(YouthBks)	0.4310	0.2475	0.1620	0.375870	1.518667	0.055328	1.205678
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240

100 rows × 9 columns

In [17]:

```
rules2[rules2.lift>1]
```

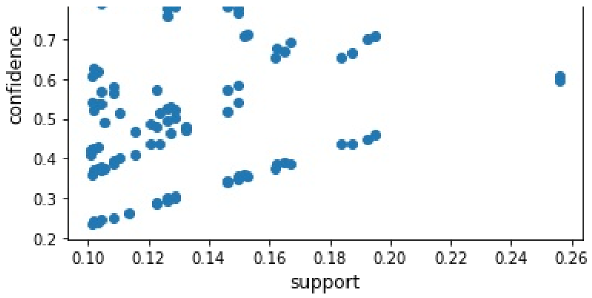
Out[17]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308	1.731000
1	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308	1.233750
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714	1.270770
...	...	...	...	...	...	...	...	...	...
95	(ArtBks, GeogBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547	3.022812
96	(CookBks, GeogBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107	1.641657
97	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107	1.415327
98	(CookBks)	(ArtBks, GeogBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547	1.148237
99	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408	1.332800

100 rows × 9 columns

In [18]:

```
plt.scatter(x='support',y='confidence', data=rules2)
plt.xlabel('support', size= 12)
plt.ylabel('confidence', size=12)
plt.title('Association  rules',size=15)
plt.show()
```



For Min Support = 0.02

In [19]:
```
freq_items3=apriori(df=books_data, min_support=0.02,use_colnames= True)
freq_items3
```

Out[19]:

	support	itemsets
0	0.4230	(ChildBks)
1	0.2475	(YouthBks)
2	0.4310	(CookBks)
3	0.2820	(DoItYBks)
4	0.2145	(RefBks)
...	...	...
271	0.0210	(ArtBks, RefBks, CookBks, DoItYBks, YouthBks, ...
272	0.0270	(RefBks, CookBks, DoItYBks, YouthBks, ChildBks...
273	0.0310	(ArtBks, CookBks, DoItYBks, YouthBks, ChildBks...
274	0.0225	(ArtBks, RefBks, CookBks, YouthBks, ChildBks, ...
275	0.0240	(ArtBks, RefBks, CookBks, DoItYBks, ChildBks, ...

276 rows × 2 columns

confidence=80%

In [20]:
```
rules3=association_rules( df=freq_items2,metric='lift',min_threshold=0.08)
rules3
```

Out[20]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308	1.731000
1	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308	1.233750
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714	1.270770
...	...	...	...	...	...	...	...	...	...
95	(ArtBks, GeogBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547	3.022812
96	(CookBks, GeogBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107	1.641657
97	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107	1.415327
98	(CookBks)	(ArtBks, GeogBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547	1.148237
99	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408	1.332800

100 rows × 9 columns

In [21]:
```
## 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 and range [0 inf]
```

In [22]:
```
rules3.sort_values('lift',ascending=False)
```

Out[22]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
29	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064582	1.203406
28	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064582	inf
76	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150	1.944628
81	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150	1.328448
85	(ArtBks)	(CookBks, DoItYBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.056313	1.403674
...	...	...	...	...	...	...	...	...	...
5	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.064714	1.660347
12	(YouthBks)	(CookBks)	0.2475	0.4310	0.1620	0.654545	1.518667	0.055328	1.647105
13	(CookBks)	(YouthBks)	0.4310	0.2475	0.1620	0.375870	1.518667	0.055328	1.205678
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240

100 rows × 9 columns

In [23]:

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

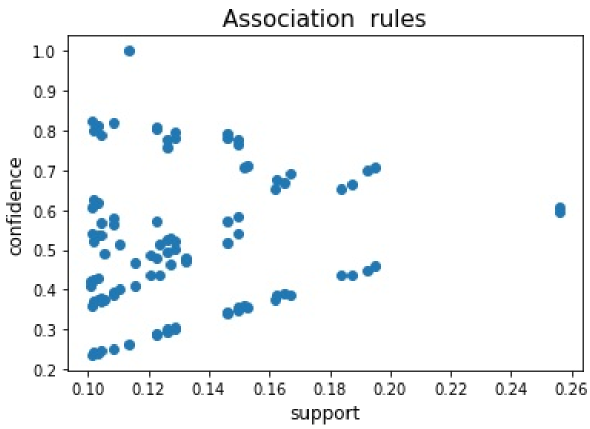
Out[23]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.060308	1.731000
1	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.060308	1.233750
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.064714	1.270770
...	...	...	...	...	...	...	...	...	...
95	(ArtBks, GeogBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547	3.022812
96	(CookBks, GeogBks)	(ArtBks)	0.1925	0.2410	0.1035	0.537662	2.230964	0.057107	1.641657
97	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057107	1.415327
98	(CookBks)	(ArtBks, GeogBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547	1.148237
99	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057408	1.332800

100 rows × 9 columns

In [24]:

```
plt.scatter( x='support',y='confidence',data=rules3)
plt.xlabel('support', size= 12)
plt.ylabel('confidence', size=12)
plt.title('Association  rules',size=15)
plt.show()
```



THE END!

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js