Import Dataset

In [9]:

```
!pip install mlxtend
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mlxtend.frequent_patterns import apriori,association_rules
from mlxtend.preprocessing import TransactionEncoder
```

```
Requirement already satisfied: mlxtend in c:\users\asus\anaconda3\lib\site-p
ackages (0.19.0)
Requirement already satisfied: setuptools in c:\users\asus\anaconda3\lib\sit
e-packages (from mlxtend) (52.0.0.post20210125)
Requirement already satisfied: scipy>=1.2.1 in c:\users\asus\anaconda3\lib\s
ite-packages (from mlxtend) (1.6.2)
Requirement already satisfied: pandas>=0.24.2 in c:\users\asus\anaconda3\lib
\site-packages (from mlxtend) (1.2.4)
Requirement already satisfied: scikit-learn>=0.20.3 in c:\users\asus\anacond
a3\lib\site-packages (from mlxtend) (0.24.1)
Requirement already satisfied: numpy>=1.16.2 in c:\users\asus\anaconda3\lib
\site-packages (from mlxtend) (1.20.1)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\asus\anaconda3
\lib\site-packages (from mlxtend) (3.3.4)
Requirement already satisfied: joblib>=0.13.2 in c:\users\asus\anaconda3\lib
\site-packages (from mlxtend) (1.0.1)
Requirement already satisfied: cycler>=0.10 in c:\users\asus\anaconda3\lib\s
ite-packages (from matplotlib>=3.0.0->mlxtend) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\asus\anaconda3\lib
\site-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:\users\asus\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend)
(2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\asus\anacond
a3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\asus\anaconda3
\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: six in c:\users\asus\anaconda3\lib\site-packa
ges (from cycler>=0.10->matplotlib>=3.0.0->mlxtend) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\asus\anaconda3\lib\s
ite-packages (from pandas>=0.24.2->mlxtend) (2021.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\asus\anacond
a3\lib\site-packages (from scikit-learn>=0.20.3->mlxtend) (2.1.0)
```

In [10]:

```
book_data = pd.read_csv("book (1).csv")
book_data
```

Out[10]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas
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

Apriori Algorithm

1. Association rules with 10% Support and 70% Confidence

In [12]:

```
# With 10% Support
frequent_itemsets=apriori(book_data,min_support=0.1,use_colnames=True)
frequent_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	(YouthBks, DoItYBks)
17	0.1010	(YouthBks, ArtBks)
18	0.1205	(YouthBks, GeogBks)
19	0.1875	(CookBks, DoItYBks)
20	0.1525	(CookBks, RefBks)
21	0.1670	(CookBks, ArtBks)
22	0.1925	(CookBks, GeogBks)
23	0.1135	(ItalCook, CookBks)
24	0.1055	(RefBks, DoItYBks)
25	0.1235	(ArtBks, DoItYBks)
26	0.1325	(GeogBks, DoItYBks)
27	0.1105	(RefBks, GeogBks)
28	0.1275	(GeogBks, ArtBks)
29	0.1290	(ChildBks, CookBks, YouthBks)
30	0.1460	(ChildBks, CookBks, DoItYBks)
31	0.1225	(ChildBks, CookBks, RefBks)
32	0.1265	(ChildBks, CookBks, ArtBks)

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

In [14]:

```
# with 70% confidence
rules = association_rules(frequent_itemsets,metric='lift', min_threshold=0.7)
rules
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.06030
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.06030
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.07368
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.07368
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.06471
...	...	...	...	...	...	...	...	.
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.05740
96	(GeogBks, ArtBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.04854
97	(CookBks)	(GeogBks, ArtBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.04854
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.05740
99	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.05710

100 rows × 9 columns

In [15]:

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

Out[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
29	(CookBks)	(ItalCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.06458
28	(ItalCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.06458
77	(ChildBks, ArtBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.05715
80	(GeogBks)	(ChildBks, ArtBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.05715
86	(ArtBks)	(CookBks, DoItYBks)	0.2410	0.1875	0.1015	0.421162	2.246196	0.05631
...	...	...	...	...	...	...	...	.
5	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.06471
12	(CookBks)	(YouthBks)	0.4310	0.2475	0.1620	0.375870	1.518667	0.05532
13	(YouthBks)	(CookBks)	0.2475	0.4310	0.1620	0.654545	1.518667	0.05532
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.07368
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.07368

100 rows × 9 columns

In [17]:

```
rules[rules.lift>1]
```

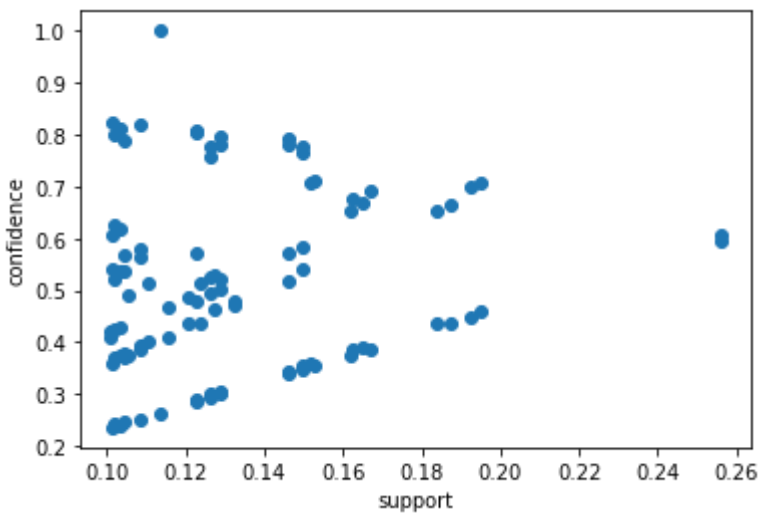
Out[17]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(ChildBks)	(YouthBks)	0.4230	0.2475	0.1650	0.390071	1.576044	0.06030
1	(YouthBks)	(ChildBks)	0.2475	0.4230	0.1650	0.666667	1.576044	0.06030
2	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.07368
3	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.07368
4	(ChildBks)	(DoItYBks)	0.4230	0.2820	0.1840	0.434988	1.542511	0.06471
...	...	...	...	...	...	...	...	.
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.05740
96	(GeogBks, ArtBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.04854
97	(CookBks)	(GeogBks, ArtBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.04854
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.05740
99	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.05710

100 rows × 9 columns

In [18]:

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



Association rules with 20% Support and 60% Confidence

In [19]:

```
# with 20% Support
frequent_itemsets2 = apriori(book_data,min_support=0.20,use_colnames=True)
frequent_itemsets2
```

Out[19]:

	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)

In [20]:

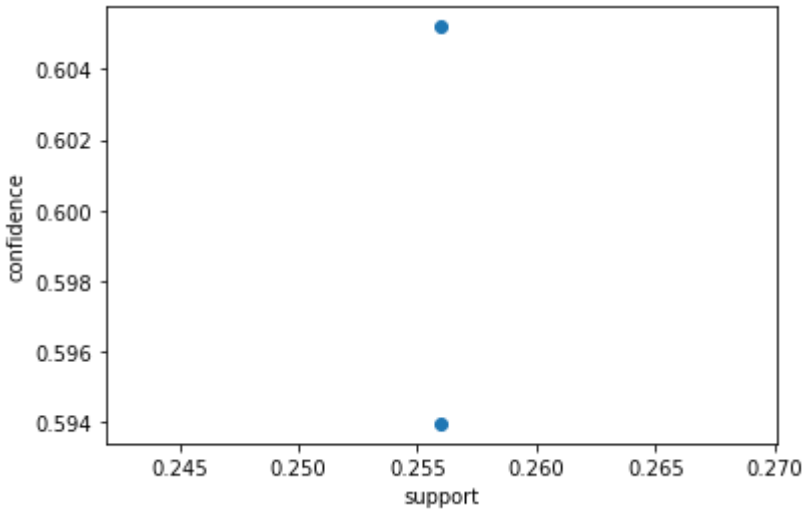
```
# with 60% confidence
rules2 = association_rules(frequent_itemsets2,metric='lift',min_threshold=0.6)
rules2
```

Out[20]:

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

In [21]:

```
plt.scatter(rules2['support'],rules2['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



Association rules with 25% Support and 80% Confidence

In [22]:

```
# with 25% Support
frequent_itemsets3 = apriori(book_data,min_support=0.25,use_colnames=True)
frequent_itemsets3
```

Out[22]:

	support	itemsets
0	0.423	(ChildBks)
1	0.431	(CookBks)
2	0.282	(DoItYBks)
3	0.276	(GeogBks)
4	0.256	(ChildBks, CookBks)

In [23]:

```
# with 80% confidence
rules3 = association_rules(frequent_itemsets3,metric='lift',min_threshold=0.8)
rules3
```

Out[23]:

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

In [24]:

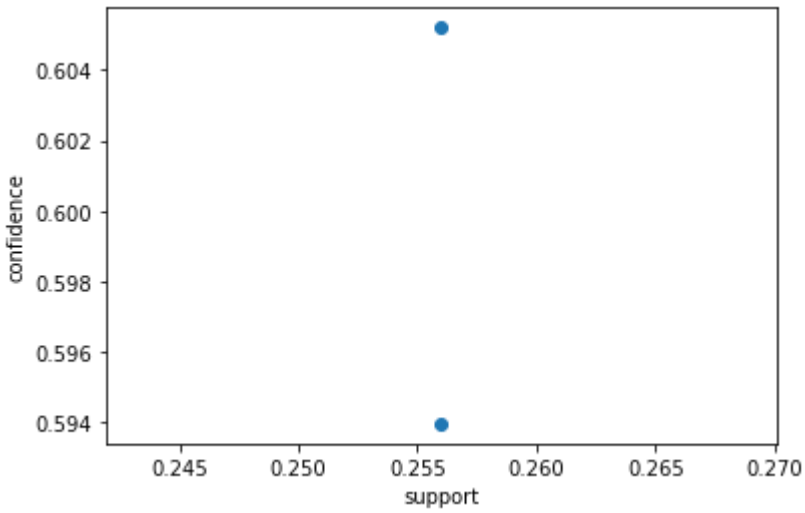
```
rules3[rules3.lift>1]
```

Out[24]:

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

In [25]:

```
plt.scatter(rules3['support'],rules3['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```



In [ ]: