Association Rules Assignment

Data Set : book

1. Import Necessary libraries

In [1]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
```

2. Import Data

In [2]:
```
books = pd.read_csv('book.csv')
books
```

Out[2]:

	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

3. Data Understanding

In [3]:
```
books.head()
```

Out[3]:

	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

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

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

In [5]:
```
books.info()
```

Out[5]:
```
<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
```

In [6]:
```
books.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]:
```
books.describe()
```

Out[7]:

	ChildBks	YouthBks	CookBks	DoItYBks	RefBks	ArtBks	GeogBks	ItalCook	ItalAtlas	ItalArt	Florence
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	0.423000	0.247500	0.43100	0.282000	0.214500	0.241000	0.276000	0.113500	0.037000	0.048500	0.108500
std	0.494159	0.431668	0.495334	0.450066	0.410578	0.427797	0.447129	0.317262	0.188859	0.214874	0.311089
min	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	1.000000	0.000000	1.00000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.00000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

In [8]:
```
books.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
```

4. Apriori Algorithm

In [9]:
```
pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (0.21.0)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: pandas>=0.24.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.4.2)
Requirement already satisfied: numpy>=1.16.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.21.5)
Requirement already satisfied: scipy>=1.2.1 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.7.3)
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (3.5.1)
Requirement already satisfied: setuptools in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (61.2.0)
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: joblib>=0.13.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.1.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.3.2)
Requirement already satisfied: cycler>=0.10 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (3.0.4)
Requirement already satisfied: packaging>=20.0 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (9.0.1)
Requirement already satisfied: pytz>=2020.1 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
```

a) Association Rule for support = 0.1

In [10]:
```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

In [11]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [12]:
```
books_items_1 = apriori(books, min_support = 0.1, use_colnames = True)
books_items_1
```

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	(YouthBks, ChildBks)
10	0.2560	(CookBks, ChildBks)
11	0.1840	(DoItYBks, ChildBks)
12	0.1515	(RefBks, ChildBks)
13	0.1625	(ArtBks, ChildBks)
14	0.1950	(GeogBks, ChildBks)
15	0.1620	(CookBks, YouthBks)
16	0.1155	(DoItYBks, YouthBks)
17	0.1010	(ArtBks, YouthBks)
18	0.1255	(YouthBks, GeogBks)
19	0.1075	(CookBks, DoItYBks)
20	0.1525	(CookBks, RefBks)
21	0.1670	(CookBks, ArtBks)
22	0.1925	(CookBks, GeogBks)
23	0.1135	(CookBks, ItalCook)
24	0.1055	(DoItYBks, RefBks)
25	0.1235	(ArtBks, DoItYBks)
26	0.1325	(DoItYBks, GeogBks)
27	0.1105	(GeogBks, RefBks)
28	0.1275	(ArtBks, GeogBks)
29	0.1260	(CookBks, YouthBks, ChildBks)
30	0.1460	(CookBks, DoItYBks, ChildBks)
31	0.1225	(CookBks, RefBks, ChildBks)
32	0.1265	(CookBks, ArtBks, ChildBks)
33	0.1495	(CookBks, GeogBks, ChildBks)
34	0.1045	(DoItYBks, GeogBks, ChildBks)
35	0.1020	(ArtBks, GeogBks, ChildBks)
36	0.1015	(CookBks, DoItYBks, ArtBks)
37	0.1065	(CookBks, DoItYBks, GeogBks)
38	0.1035	(CookBks, GeogBks, ArtBks)

In [13]:
```
books_rules_1 = apriori(books, min_support = 0.1, use_colnames = True)
books_rules_1
```

Out[13]:

	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	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
3	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
4	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.064714	1.660347
...	...	...	...	...	...	...	...	...	...
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057406	1.904063
96	(ArtBks, GeogBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547	3.022812
97	(CookBks)	(ArtBks, GeogBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547	1.148237
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057406	1.332900
99	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057187	1.415327

100 rows × 9 columns

In [14]:
```
a_1 = books_rules_1[books_rules_1.lift > 1]
a_1
```

Out[14]:

	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	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069
3	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
4	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.064714	1.660347
...	...	...	...	...	...	...	...	...	...
95	(CookBks, ArtBks)	(GeogBks)	0.1670	0.2760	0.1035	0.619760	2.245509	0.057406	1.904063
96	(ArtBks, GeogBks)	(CookBks)	0.1275	0.4310	0.1035	0.811765	1.883445	0.048547	3.022812
97	(CookBks)	(ArtBks, GeogBks)	0.4310	0.1275	0.1035	0.240139	1.883445	0.048547	1.148237
98	(GeogBks)	(CookBks, ArtBks)	0.2760	0.1670	0.1035	0.375000	2.245509	0.057406	1.332900
99	(ArtBks)	(CookBks, GeogBks)	0.2410	0.1925	0.1035	0.429461	2.230964	0.057187	1.415327

100 rows × 9 columns

In [15]:
```
b_1 = a_1.sort_values("lift", ascending = False)
b_1
```

Out[15]:

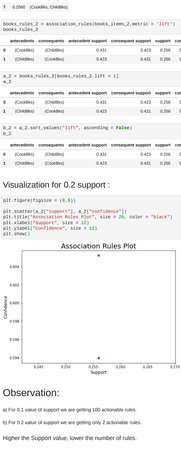
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
28	(CookBks)	(NatCook)	0.4310	0.1135	0.1135	0.263341	2.320186	0.064592	1.203400
29	(NatCook)	(CookBks)	0.1135	0.4310	0.1135	1.000000	2.320186	0.064592	inf
77	(ArtBks, CookBks)	(GeogBks)	0.1625	0.2760	0.1020	0.627692	2.274247	0.057150	1.946828
80	(GeogBks)	(ArtBks, CookBks)	0.2760	0.1625	0.1020	0.369565	2.274247	0.057150	1.339448
87	(ArtBks)	(CookBks, DoItYBks)	0.2410	0.1075	0.1015	0.421162	2.249196	0.056313	1.403674
...	...	...	...	...	...	...	...	...	...
4	(DoItYBks)	(ChildBks)	0.2820	0.4230	0.1840	0.652482	1.542511	0.064714	1.660347
12	(ChildBks)	(YouthBks)	0.4310	0.2475	0.1620	0.375870	1.518667	0.055329	1.205678
13	(YouthBks)	(CookBks)	0.2475	0.4310	0.1620	0.654545	1.518667	0.055329	1.647109
3	(ChildBks)	(CookBks)	0.4230	0.4310	0.2560	0.605201	1.404179	0.073687	1.441240
2	(CookBks)	(ChildBks)	0.4310	0.4230	0.2560	0.593968	1.404179	0.073687	1.421069

100 rows × 9 columns

Visualization for 0.1 support :

In [16]:
```
plt.figure(figsize = (8,6))

plt.scatter(a_1["support"], a_1["confidence"])
plt.title("Association Rules Plot", size = 20, color = "black")
plt.xlabel("Support", size = 12)
plt.ylabel("Confidence", size = 12)
plt.show()
```



b) Association Rule for support = 0.2

In [17]:
```
books_items_2 = apriori(books, min_support = 0.2, use_colnames = True)
books_items_2
```

Out[17]:

	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	(CookBks, ChildBks)

In [18]:
```
books_rules_2 = association_rules(books_items_2,metric = 'lift')
books_rules_2
```

Out[18]:

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

In [19]:
```
a_2 = books_rules_2[books_rules_2.lift > 1]
a_2
```

Out[19]:

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

In [20]:
```
b_2 = a_2.sort_values("lift", ascending = False)
b_2
```

Out[20]:

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

Visualization for 0.2 support :

In [21]:
```
plt.figure(figsize = (8,6))

plt.scatter(a_2["support"], a_2["confidence"])
plt.title("Association Rules Plot", size = 20, color = "black")
plt.xlabel("Support", size = 12)
plt.ylabel("Confidence", size = 12)
plt.show()
```


Observation:

a) For 0.1 value of support we are getting 100 actionable rules.

b) For 0.2 value of support we are getting only 2 actionable rules.

Higher the Support value, lower the number of rules.