Association Rules Assignment

Data Set : my_movies

1. Import Necessary libraries

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

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

2. Import Data

In [2]:
```python
movies = pd.read_csv('my_movies.csv')
movies
```

Out[2]:

	V1	V2	V3	V4	V5	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Patriot	LOTR2	Harry Potter2	LOTR	Braveheart	Green Mile
0	Sixth Sense	LOTR1	Harry Potter1	Green Mile	LOTR2	1	0	1	1	0	1	0	0	0	1
1	Gladiator	Patriot	Braveheart	NaN	NaN	0	1	0	0	1	0	0	0	1	0
2	LOTR1	LOTR2	NaN	NaN	NaN	0	0	1	0	0	1	0	0	0	0
3	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	0	0	0	0	0
4	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	0	0	0	0	0
5	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	0	0	0	0	0
6	Harry Potter1	Harry Potter2	NaN	NaN	NaN	0	0	0	1	0	0	1	0	0	0
7	Gladiator	Patriot	NaN	NaN	NaN	0	1	0	0	1	0	0	0	0	0
8	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	0	0	0	0	0
9	Sixth Sense	LOTR	Gladiator	Green Mile	NaN	1	1	0	0	0	0	0	1	0	1

3. Data Understanding

In [3]:
```python
movies.head()
```

Out[3]:

	V1	V2	V3	V4	V5	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Patriot	LOTR2	Harry Potter2	LOTR	Braveheart	Green Mile
0	Sixth Sense	LOTR1	Harry Potter1	Green Mile	LOTR2	1	0	1	1	0	1	0	0	0	1
1	Gladiator	Patriot	Braveheart	NaN	NaN	0	1	0	0	1	0	0	0	1	0
2	LOTR1	LOTR2	NaN	NaN	NaN	0	0	1	0	0	1	0	0	0	0
3	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	0	0	0	0	0
4	Gladiator	Patriot	Sixth Sense	NaN	NaN	1	1	0	0	1	0	0	0	0	0

In [4]:
```python
movies.shape
```

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

In [5]:
```python
movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   V1             10 non-null     object
 1   V2             10 non-null     object
 2   V3             7 non-null      object
 3   V4             2 non-null      object
 4   V5             1 non-null      object
 5   Sixth Sense    10 non-null     int64 
 6   Gladiator      10 non-null     int64 
 7   LOTR1          10 non-null     int64 
 8   Harry Potter1  10 non-null     int64 
 9   Patriot        10 non-null     int64 
 10  LOTR2          10 non-null     int64 
 11  Harry Potter2  10 non-null     int64 
 12  LOTR           10 non-null     int64 
 13  Braveheart     10 non-null     int64 
 14  Green Mile     10 non-null     int64 
dtypes: int64(10), object(5)
memory usage: 1.3+ KB
```

In [6]:
```python
movies.isna().sum()
```

Out[6]:
```
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 [7]:
```python
movies.describe()
```

Out[7]:

	Sixth Sense	Gladiator	LOTR1	Harry Potter1	Patriot	LOTR2	Harry Potter2	LOTR	Braveheart	Green Mile
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	0.600000	0.700000	0.200000	0.200000	0.600000	0.200000	0.100000	0.100000	0.100000	0.200000
std	0.516398	0.483046	0.421637	0.421637	0.516398	0.421637	0.316228	0.316228	0.316228	0.421637
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.250000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	1.000000	1.000000	0.000000	0.000000	1.000000	0.000000	0.000000	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

In [8]:
```python
movies.dtypes
```

Out[8]:
```
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 [9]:
```python
holly_movies = movies.iloc[:,5:]
holly_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

4. Apriori Algorithm

In [10]:
```python
pip install mlxtend
```

```
Requirement already satisfied: mlxtend in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (0.21.0)
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: joblib>=0.13.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.1.0)
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: pandas>=0.24.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.4.2)
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: scikit-learn>=1.0.2 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (1.0.2)
Requirement already satisfied: setuptools in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from mlxtend) (61.2.0)
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: packaging>=20.0 in c:\users\mohammed faisal khan\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (21.3)
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: 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: 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: 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)
Note: you may need to restart the kernel to use updated packages.
```

a) Association Rule for support = 0.1

In [11]:
```python
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

import warnings
warnings.filterwarnings('ignore')
```

In [12]:
```python
movie_items_1 = apriori(holly_movies, min_support = 0.1, use_colnames = True)
movie_items_1
```

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	(Sixth Sense, Gladiator)
11	0.1	(LOTR1, Sixth Sense)
12	0.1	(Harry Potter1, Sixth Sense)
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	(Gladiator, Braveheart)
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	(Harry Potter1, LOTR2)
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, Sixth Sense, Gladiator)
31	0.1	(LOTR, Sixth Sense, Gladiator)
32	0.1	(Green Mile, Sixth Sense, Gladiator)
33	0.1	(LOTR1, Harry Potter1, Sixth Sense)
34	0.1	(LOTR1, LOTR2, Sixth Sense)
35	0.1	(Green Mile, LOTR1, Sixth Sense)
36	0.1	(LOTR2, Harry Potter1, Sixth Sense)
37	0.1	(Green Mile, Harry Potter1, Sixth Sense)
38	0.1	(Green Mile, LOTR2, Sixth Sense)
39	0.1	(Green Mile, LOTR, Sixth Sense)
40	0.1	(Patriot, Gladiator, Braveheart)
41	0.1	(Green Mile, LOTR, Gladiator)
42	0.1	(LOTR1, Harry Potter1, LOTR2)
43	0.1	(Green Mile, LOTR1, Harry Potter1)
44	0.1	(Green Mile, LOTR1, LOTR2)
45	0.1	(Green Mile, Harry Potter1, LOTR2)
46	0.1	(Green Mile, LOTR, Sixth Sense, Gladiator)
47	0.1	(LOTR2, LOTR1, Harry Potter1, Sixth Sense)
48	0.1	(Green Mile, LOTR1, Harry Potter1, Sixth Sense)
49	0.1	(Green Mile, LOTR1, LOTR2, Sixth Sense)
50	0.1	(Green Mile, Harry Potter1, Sixth Sense, LOTR2)
51	0.1	(Green Mile, LOTR1, Harry Potter1, LOTR2)
52	0.1	(LOTR1, Sixth Sense, Green Mile, LOTR2, Harry ...

In [13]:
```python
movie_rules_1 = association_rules(movie_items_1,metric = 'lift')
movie_rules_1
```

Out[13]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.80
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.40
2	(LOTR1)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.02	0.80
3	(Sixth Sense)	(LOTR1)	0.6	0.2	0.1	0.166667	0.833333	-0.02	0.96
4	(Harry Potter1)	(Sixth Sense)	0.2	0.6	0.1	0.500000	0.833333	-0.02	0.80
...	...	...	...	...	...	...	...	...	...
241	(LOTR1)	(Green Mile, LOTR2, Sixth Sense, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80
242	(Sixth Sense)	(Green Mile, LOTR1, LOTR2, Harry Potter1)	0.6	0.1	0.1	0.166667	1.666667	0.04	1.08
243	(Green Mile)	(LOTR1, LOTR2, Sixth Sense, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80
244	(LOTR2)	(Green Mile, LOTR1, Harry Potter1, Sixth Sense)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80
245	(Harry Potter1)	(Green Mile, LOTR1, LOTR2, Sixth Sense)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80

246 rows × 9 columns

In [14]:
```python
a_1 = movie_rules_1[movie_rules_1.lift > 1]
a_1
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.80
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.40
6	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.20
7	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.20
10	(LOTR)	(Sixth Sense)	0.1	0.6	0.1	1.000000	1.666667	0.04	inf
...	...	...	...	...	...	...	...	...	...
241	(LOTR1)	(Green Mile, LOTR2, Sixth Sense, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80
242	(Sixth Sense)	(Green Mile, LOTR1, LOTR2, Harry Potter1)	0.6	0.1	0.1	0.166667	1.666667	0.04	1.08
243	(Green Mile)	(LOTR1, LOTR2, Sixth Sense, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80
244	(LOTR2)	(Green Mile, LOTR1, Harry Potter1, Sixth Sense)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80
245	(Harry Potter1)	(Green Mile, LOTR1, LOTR2, Sixth Sense)	0.2	0.1	0.1	0.500000	5.000000	0.08	1.80

236 rows × 9 columns

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

Out[15]:

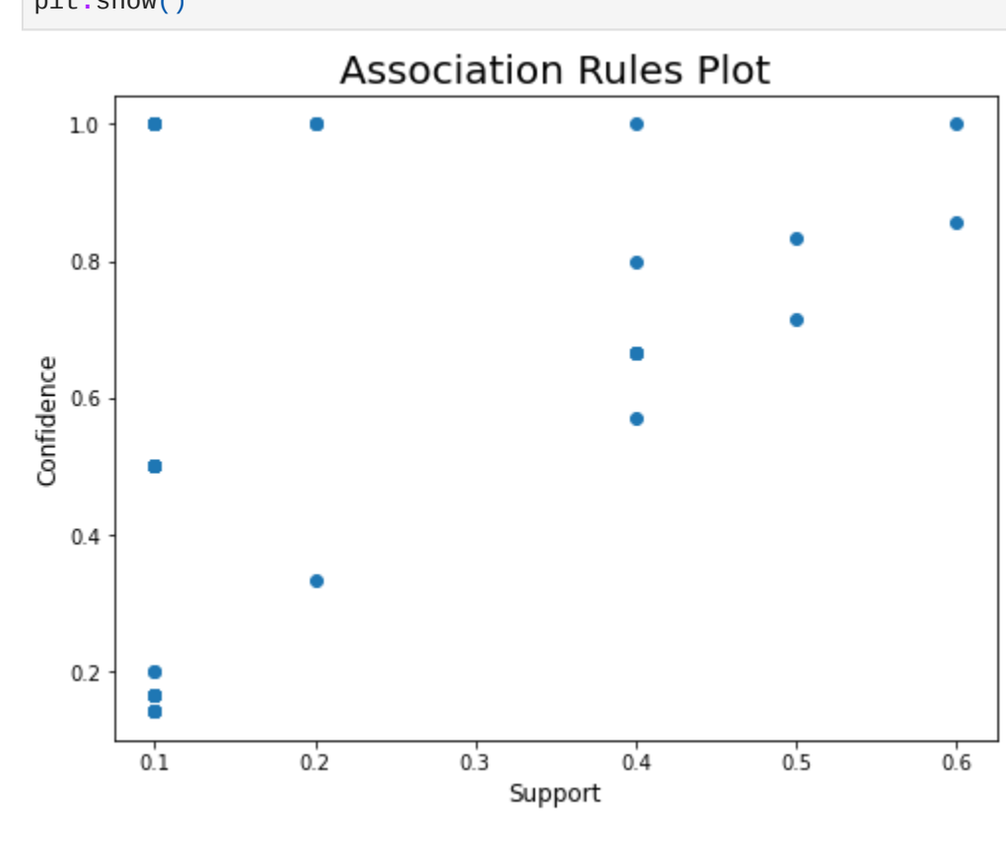
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
168	(LOTR1, Sixth Sense)	(Green Mile, Harry Potter1)	0.1	0.1	0.1	1.000000	10.000000	0.09	inf
222	(LOTR1, LOTR2, Sixth Sense)	(Green Mile, Harry Potter1)	0.1	0.1	0.1	1.000000	10.000000	0.09	inf
192	(Green Mile, Harry Potter1)	(LOTR2, Sixth Sense)	0.1	0.1	0.1	1.000000	10.000000	0.09	inf
134	(Green Mile, Sixth Sense, Gladiator)	(LOTR)	0.1	0.1	0.1	1.000000	10.000000	0.09	inf
183	(LOTR2, Sixth Sense)	(Green Mile, LOTR1)	0.1	0.1	0.1	1.000000	10.000000	0.09	inf
...	...	...	...	...	...	...	...	...	...
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
42	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
39	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
7	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
6	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2

236 rows × 9 columns

Visualization for 0.1 support :

In [16]:
```python
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]:
```python
movie_items_2 = apriori(holly_movies, min_support = 0.2, use_colnames = True)
movie_items_2
```

Out[17]:

	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	(Sixth Sense, Gladiator)
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, Sixth Sense, Gladiator)

In [18]:
```python
movie_rules_2 = association_rules(movie_items_2,metric = 'lift')
movie_rules_2
```

Out[18]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
4	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf
5	(Sixth Sense)	(Green Mile)	0.6	0.2	0.2	0.333333	1.666667	0.08	1.2
6	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
7	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
8	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.16	inf
9	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.16	inf
10	(Patriot, Sixth Sense)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	inf
11	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
12	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	2.0
13	(Patriot)	(Sixth Sense, Gladiator)	0.6	0.5	0.4	0.666667	1.333333	0.10	1.5
14	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
15	(Gladiator)	(Patriot, Sixth Sense)	0.7	0.4	0.4	0.571429	1.428571	0.12	1.4

In [19]:
```python
a_2 = movie_rules_2[movie_rules_2.lift > 1]
a_2
```

Out[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
4	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf
5	(Sixth Sense)	(Green Mile)	0.6	0.2	0.2	0.333333	1.666667	0.08	1.2
6	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
7	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
8	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.16	inf
9	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.16	inf
10	(Patriot, Sixth Sense)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	inf
11	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
12	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	2.0
13	(Patriot)	(Sixth Sense, Gladiator)	0.6	0.5	0.4	0.666667	1.333333	0.10	1.5
14	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
15	(Gladiator)	(Patriot, Sixth Sense)	0.7	0.4	0.4	0.571429	1.428571	0.12	1.4

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

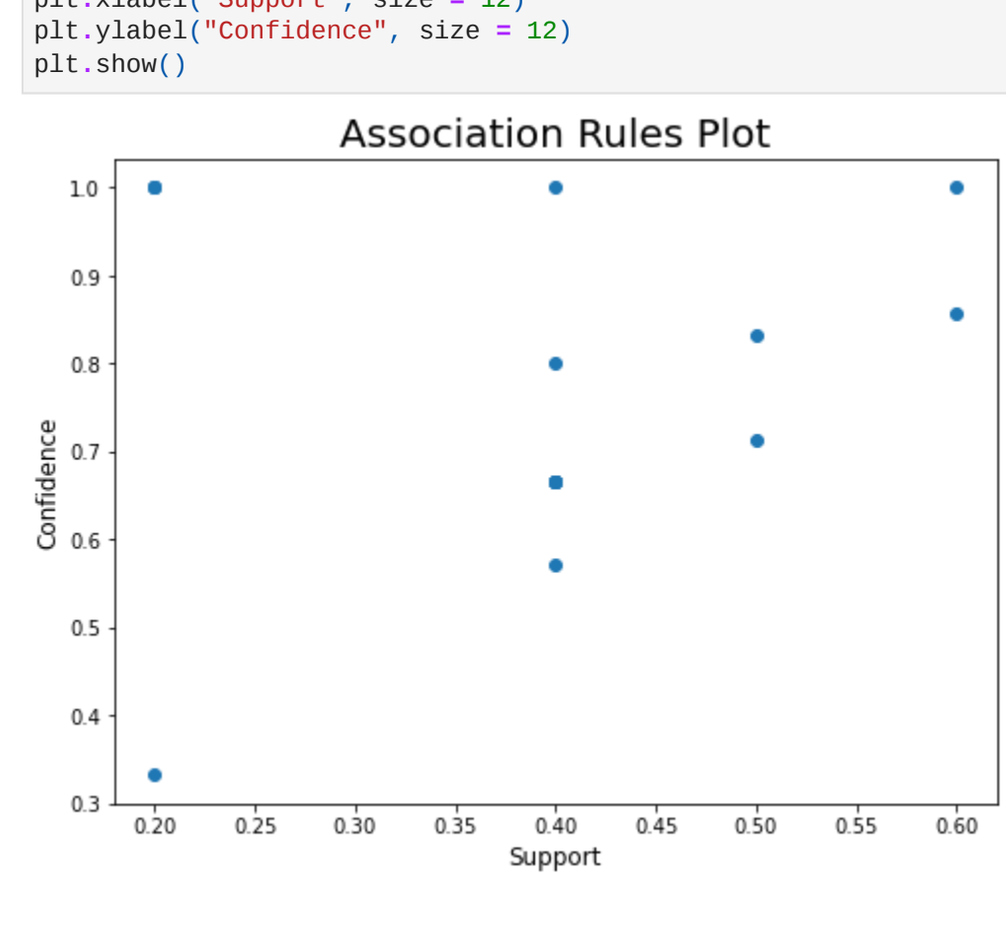
Out[20]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
8	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.16	inf
9	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.16	inf
4	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf
5	(Sixth Sense)	(Green Mile)	0.6	0.2	0.2	0.333333	1.666667	0.08	1.2
7	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
6	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
10	(Patriot, Sixth Sense)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	inf
15	(Gladiator)	(Patriot, Sixth Sense)	0.7	0.4	0.4	0.571429	1.428571	0.12	1.4
12	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	2.0
13	(Patriot)	(Sixth Sense, Gladiator)	0.6	0.5	0.4	0.666667	1.333333	0.10	1.5
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
11	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
14	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2

Visualization for 0.2 support :

In [21]:
```python
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()
```



c) Association Rule for support = 0.3

In [22]:
```python
movie_items_3 = apriori(holly_movies, min_support = 0.3, use_colnames = True)
movie_items_3
```

Out[22]:

	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.6	(Patriot)
3	0.5	(Sixth Sense, Gladiator)
4	0.4	(Patriot, Sixth Sense)
5	0.6	(Patriot, Gladiator)
6	0.4	(Patriot, Sixth Sense, Gladiator)

In [23]:
```python
movie_rules_3 = association_rules(movie_items_3,metric = 'lift')
movie_rules_3
```

Out[23]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
4	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
5	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
6	(Patriot, Sixth Sense)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	inf
7	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
8	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	2.0
9	(Patriot)	(Sixth Sense, Gladiator)	0.6	0.5	0.4	0.666667	1.333333	0.10	1.5
10	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
11	(Gladiator)	(Patriot, Sixth Sense)	0.7	0.4	0.4	0.571429	1.428571	0.12	1.4

In [24]:
```python
a_3 = movie_rules_3[movie_rules_3.lift > 1]
a_3
```

Out[24]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
4	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
5	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
6	(Patriot, Sixth Sense)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	inf
7	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
8	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	2.0
9	(Patriot)	(Sixth Sense, Gladiator)	0.6	0.5	0.4	0.666667	1.333333	0.10	1.5
10	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
11	(Gladiator)	(Patriot, Sixth Sense)	0.7	0.4	0.4	0.571429	1.428571	0.12	1.4

In [25]:
```python
b_3 = a_3.sort_values("lift", ascending = False)
b_3
```

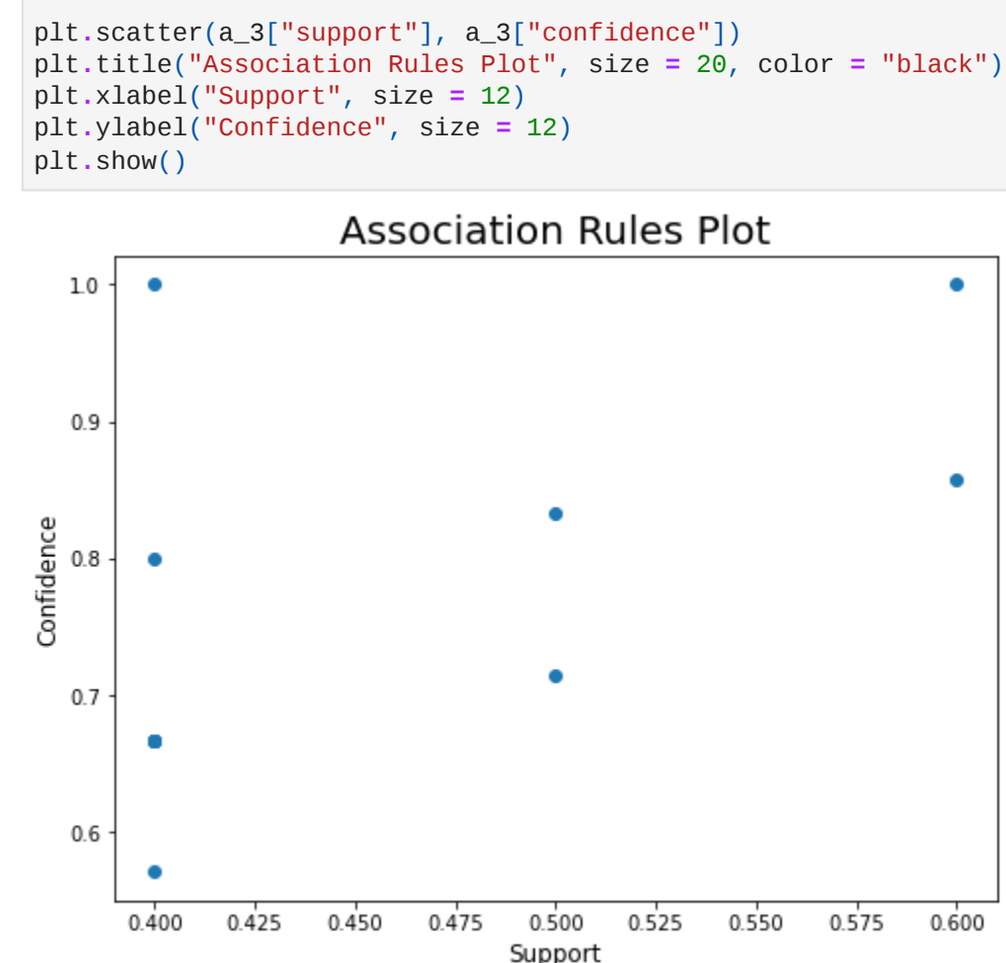
Out[25]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
5	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
4	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
6	(Patriot, Sixth Sense)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.12	inf
11	(Gladiator)	(Patriot, Sixth Sense)	0.7	0.4	0.4	0.571429	1.428571	0.12	1.4
8	(Sixth Sense, Gladiator)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.10	2.0
9	(Patriot)	(Sixth Sense, Gladiator)	0.6	0.5	0.4	0.666667	1.333333	0.10	1.5
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
3	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
7	(Patriot, Gladiator)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2
10	(Sixth Sense)	(Patriot, Gladiator)	0.6	0.6	0.4	0.666667	1.111111	0.04	1.2

Visualization for 0.3 support :

In [26]:
```python
plt.figure(figsize = (8,6))

plt.scatter(a_3["support"], a_3["confidence"])
plt.title("Association Rules Plot", size = 20, color = "black")
plt.xlabel("Support", size = 12)
plt.ylabel("Confidence", size = 12)
plt.show()
```



d) Association Rule for support = 0.5

In [27]:
```python
movie_items_4 = apriori(holly_movies, min_support = 0.5, use_colnames = True)
movie_items_4
```

Out[27]:

	support	itemsets
0	0.6	(Sixth Sense)
1	0.7	(Gladiator)
2	0.6	(Patriot)
3	0.5	(Sixth Sense, Gladiator)
4	0.6	(Patriot, Gladiator)

In [28]:
```python
movie_rules_4 = association_rules(movie_items_4,metric = 'lift')
movie_rules_4
```

Out[28]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
3	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8

In [29]:
```python
a_4 = movie_rules_4[movie_rules_4.lift > 1]
a_4
```

Out[29]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4
2	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
3	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8

In [30]:
```python
b_4 = a_4.sort_values("lift", ascending = False)
b_4
```

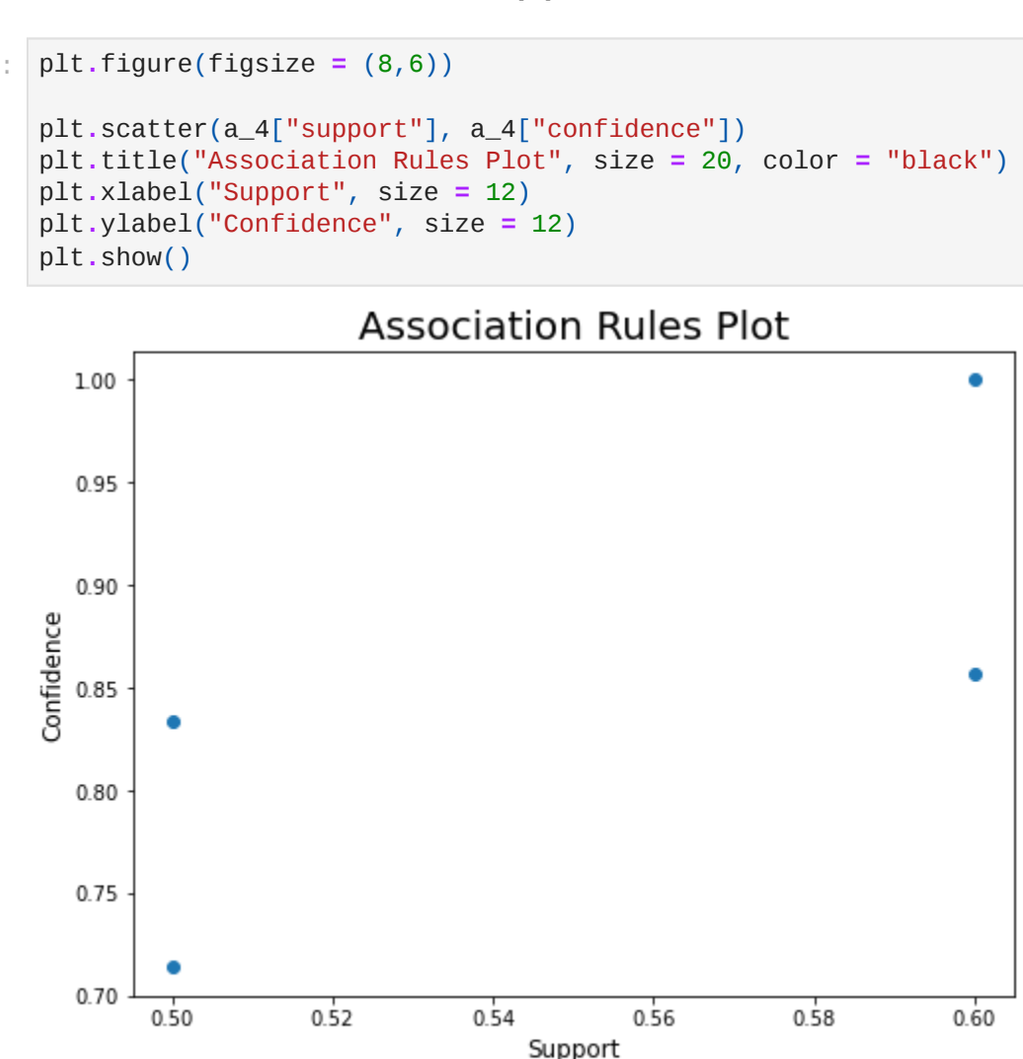
Out[30]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
3	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.18	2.8
2	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.18	inf
0	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.08	1.8
1	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.08	1.4

Visualization for 0.5 support :

In [31]:
```python
plt.figure(figsize = (8,6))

plt.scatter(a_4["support"], a_4["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, 246 rules generated out of which 236 are actionable rules.

b) For 0.2 value of support we got 16 actionable rules.

c) For 0.3 value of support we got 12 actionable rules.

d) For 0.5 value of support we got 4 actionable rules.

Higher the Support value, lower the number of rules.