Importing necessary liabrary

In [1]:

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

Import Data

In [2]:

```python
movies_data=pd.read_csv("my_movies.csv")
movies_data
```

Out[2]:

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

Initial Analysis

In [3]:

```python
movies_data.shape
```

Out[3]:

```
(10, 15)
```

In [4]:

```python
movies_data.dtypes
```

Out[4]:

```
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 [5]:

```python
movies_data.isna().sum()
```

Out[5]:

```
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
```

Data preprocessing

In [6]:

```python
movie_data1=movies_data.iloc[:,5:]
```

In [7]:

```python
movie_data1
```

Out[7]:

	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

Building Association Rules

Association rules with 10% support 95% confidence

In [8]:

```python
#Purpose of Apriori : to build freq item sets
freq_itemsets = apriori(movie_data1,min_support=0.10,use_colnames=True)
freq_itemsets
```

Out[8]:

	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	(Gladiator, Sixth Sense)
11	0.1	(LOTR1, Sixth Sense)
12	0.1	(Sixth Sense, Harry Potter1)
13	0.4	(Sixth Sense, Patriot)
14	0.1	(LOTR2, Sixth Sense)
15	0.1	(Sixth Sense, LOTR)
16	0.2	(Sixth Sense, Green Mile)
17	0.6	(Gladiator, Patriot)
18	0.1	(Gladiator, LOTR)
19	0.1	(Gladiator, Braveheart)
20	0.1	(Gladiator, Green Mile)
21	0.1	(LOTR1, Harry Potter1)
22	0.2	(LOTR2, LOTR1)
23	0.1	(LOTR1, Green Mile)
24	0.1	(LOTR2, Harry Potter1)
25	0.1	(Harry Potter2, Harry Potter1)
26	0.1	(Harry Potter1, Green Mile)
27	0.1	(Patriot, Braveheart)
28	0.1	(LOTR2, Green Mile)
29	0.1	(LOTR, Green Mile)
30	0.4	(Gladiator, Sixth Sense, Patriot)
31	0.1	(Gladiator, Sixth Sense, LOTR)
32	0.1	(Gladiator, Sixth Sense, Green Mile)

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

In [9]:

```python
#Purpose of Association Rules - to generate the best associations
rules= association_rules(df =freq_itemsets ,metric='confidence',min_threshold=0.95)
rules.sort_values(by ='support',axis=0,ascending=False)
```

Out[9]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
2	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.0	1.428571	0
10	(Sixth Sense, Patriot)	(Gladiator)	0.4	0.7	0.4	1.0	1.428571	0
1	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.0	1.666667	0
5	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.0	5.000000	0
6	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.0	5.000000	0
...	...	...	...	...	...	...	...	
39	(LOTR2, Harry Potter1)	(LOTR1)	0.1	0.2	0.1	1.0	5.000000	0
38	(LOTR)	(Gladiator, Green Mile)	0.1	0.1	0.1	1.0	10.000000	0
37	(LOTR, Green Mile)	(Gladiator)	0.1	0.7	0.1	1.0	1.428571	0
36	(Gladiator, Green Mile)	(LOTR)	0.1	0.1	0.1	1.0	10.000000	0
124	(LOTR2, Harry Potter1)	(Sixth Sense, LOTR1, Green Mile)	0.1	0.1	0.1	1.0	10.000000	0

125 rows × 9 columns

In [10]:

```
rules[rules.lift>1]   #lift>1  It means it is best association.
```
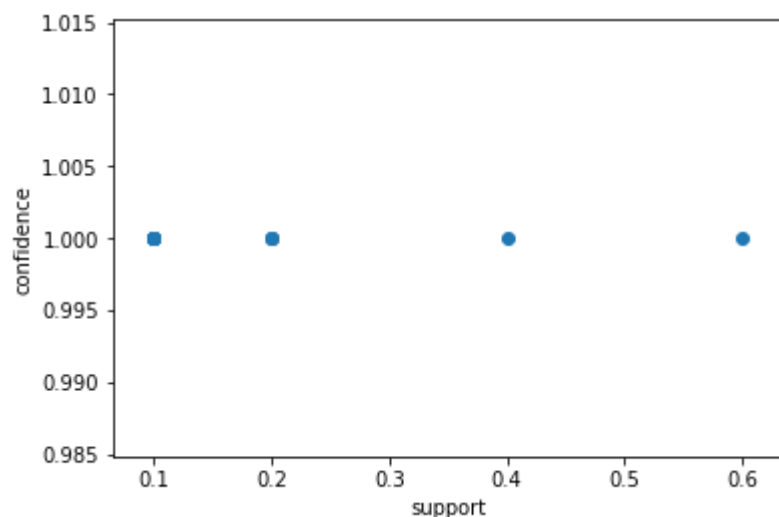
Out[10]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	lever:
0	(LOTR)	(Sixth Sense)	0.1	0.6	0.1	1.0	1.666667	0
1	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.0	1.666667	0
2	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.0	1.428571	0
3	(LOTR)	(Gladiator)	0.1	0.7	0.1	1.0	1.428571	0
4	(Braveheart)	(Gladiator)	0.1	0.7	0.1	1.0	1.428571	0
...	...	...	...	...	...	...	...	
120	(LOTR2, Sixth Sense)	(LOTR1, Harry Potter1, Green Mile)	0.1	0.1	0.1	1.0	10.000000	0
121	(Sixth Sense, Harry Potter1)	(LOTR2, LOTR1, Green Mile)	0.1	0.1	0.1	1.0	10.000000	0
122	(LOTR2, Green Mile)	(Sixth Sense, LOTR1, Harry Potter1)	0.1	0.1	0.1	1.0	10.000000	0
123	(Harry Potter1, Green Mile)	(Sixth Sense, LOTR2, LOTR1)	0.1	0.1	0.1	1.0	10.000000	0
124	(LOTR2, Harry Potter1)	(Sixth Sense, LOTR1, Green Mile)	0.1	0.1	0.1	1.0	10.000000	0

125 rows × 9 columns

In [11]:

```python
# visualization of obtained rule
plt.scatter(rules['support'],rules['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```


Association rules with 20% support and 90% Confidence

In [12]:

```python
#with 20% Support
freq_itemsets1 = apriori(movie_data1,min_support=0.20,use_colnames=True)
freq_itemsets1
```

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	0.2	(Green Mile)
7	0.5	(Gladiator, Sixth Sense)
8	0.4	(Sixth Sense, Patriot)
9	0.2	(Sixth Sense, Green Mile)
10	0.6	(Gladiator, Patriot)
11	0.2	(LOTR2, LOTR1)
12	0.4	(Gladiator, Sixth Sense, Patriot)

In [13]:

```python
#with 90% Confidence
rules1= association_rules(df =freq_itemsets ,metric='lift',min_threshold=0.90)
rules1.sort_values(by ='support',axis=0,ascending=False)
```

Out[13]:

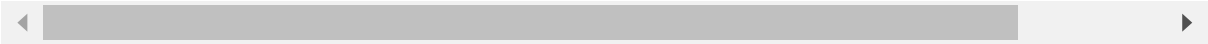|	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	levera
9	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0
8	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0
37	(Patriot)	(Gladiator, Sixth Sense)	0.6	0.5	0.4	0.666667	1.333333	0
...	...	...	...	...	...	...	...	
92	(Patriot)	(Gladiator, Braveheart)	0.6	0.1	0.1	0.166667	1.666667	0
93	(Braveheart)	(Gladiator, Patriot)	0.1	0.6	0.1	1.000000	1.666667	0
94	(Gladiator, LOTR)	(Green Mile)	0.1	0.2	0.1	1.000000	5.000000	0
95	(Gladiator, Green Mile)	(LOTR)	0.1	0.1	0.1	1.000000	10.000000	0
237	(Harry Potter1)	(Sixth Sense, LOTR2, LOTR1, Green Mile)	0.2	0.1	0.1	0.500000	5.000000	0

238 rows × 9 columns
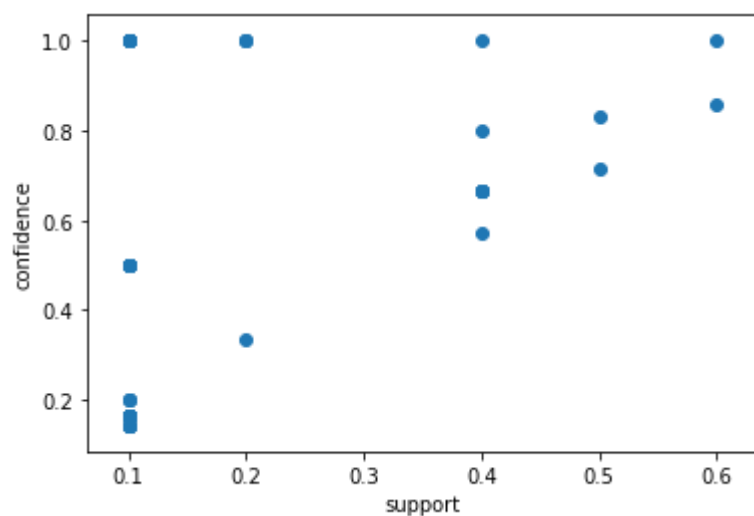
In [14]:

```
rules1[rules1.lift>1]  #Best Association
```

Out[14]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.
2	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.
3	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.
4	(Sixth Sense)	(LOTR)	0.6	0.1	0.1	0.166667	1.666667	0.
...	...	...	...	...	...	...	...	
233	(LOTR1)	(LOTR2, Sixth Sense, Harry Potter1, Green Mile)	0.2	0.1	0.1	0.500000	5.000000	0.
234	(Sixth Sense)	(LOTR2, LOTR1, Harry Potter1, Green Mile)	0.6	0.1	0.1	0.166667	1.666667	0.
235	(Green Mile)	(Sixth Sense, LOTR2, LOTR1, Harry Potter1)	0.2	0.1	0.1	0.500000	5.000000	0.
236	(LOTR2)	(Sixth Sense, LOTR1, Harry Potter1, Green Mile)	0.2	0.1	0.1	0.500000	5.000000	0.
237	(Harry Potter1)	(Sixth Sense, LOTR2, LOTR1, Green Mile)	0.2	0.1	0.1	0.500000	5.000000	0.

236 rows × 9 columns

In [15]:

```python
plt.scatter(rules1['support'],rules1['confidence'])
plt.xlabel('support')
plt.ylabel('confidence')
plt.show()
```


Association rules with 15% support and 80% confidence

In [16]:

```python
#with 15% Support
freq_itemsets2 = apriori(movie_data1,min_support=0.15,use_colnames=True)
freq_itemsets2
```

Out[16]:

	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	(Gladiator, Sixth Sense)
8	0.4	(Sixth Sense, Patriot)
9	0.2	(Sixth Sense, Green Mile)
10	0.6	(Gladiator, Patriot)
11	0.2	(LOTR2, LOTR1)
12	0.4	(Gladiator, Sixth Sense, Patriot)

In [17]:

```python
#with 80% Confidence
rules2= association_rules(df =freq_itemsets1 ,metric='lift',min_threshold=0.90)
rules2.sort_values(by ='support',axis=0,ascending=False)
```

Out[17]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
6	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.1i
7	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.1i
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.0i
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.0i
2	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04
3	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04
10	(Gladiator, Sixth Sense)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.1(
11	(Gladiator, Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.04
12	(Sixth Sense, Patriot)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.1i
13	(Gladiator)	(Sixth Sense, Patriot)	0.7	0.4	0.4	0.571429	1.428571	0.1i
14	(Sixth Sense)	(Gladiator, Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.04
15	(Patriot)	(Gladiator, Sixth Sense)	0.6	0.5	0.4	0.666667	1.333333	0.1(
4	(Sixth Sense)	(Green Mile)	0.6	0.2	0.2	0.333333	1.666667	0.0i
5	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.0i
8	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.1(
9	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.1(

In [18]:

```python
rules2[rules2.lift>1]   #Best association
```

Out[18]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverag
0	(Gladiator)	(Sixth Sense)	0.7	0.6	0.5	0.714286	1.190476	0.0
1	(Sixth Sense)	(Gladiator)	0.6	0.7	0.5	0.833333	1.190476	0.0
2	(Sixth Sense)	(Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.0
3	(Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.0
4	(Sixth Sense)	(Green Mile)	0.6	0.2	0.2	0.333333	1.666667	0.0
5	(Green Mile)	(Sixth Sense)	0.2	0.6	0.2	1.000000	1.666667	0.0
6	(Gladiator)	(Patriot)	0.7	0.6	0.6	0.857143	1.428571	0.1
7	(Patriot)	(Gladiator)	0.6	0.7	0.6	1.000000	1.428571	0.1
8	(LOTR2)	(LOTR1)	0.2	0.2	0.2	1.000000	5.000000	0.1
9	(LOTR1)	(LOTR2)	0.2	0.2	0.2	1.000000	5.000000	0.1
10	(Gladiator, Sixth Sense)	(Patriot)	0.5	0.6	0.4	0.800000	1.333333	0.1
11	(Gladiator, Patriot)	(Sixth Sense)	0.6	0.6	0.4	0.666667	1.111111	0.0
12	(Sixth Sense, Patriot)	(Gladiator)	0.4	0.7	0.4	1.000000	1.428571	0.1
13	(Gladiator)	(Sixth Sense, Patriot)	0.7	0.4	0.4	0.571429	1.428571	0.1
14	(Sixth Sense)	(Gladiator, Patriot)	0.6	0.6	0.4	0.666667	1.111111	0.0
15	(Patriot)	(Gladiator, Sixth Sense)	0.6	0.5	0.4	0.666667	1.333333	0.1

In [19]:

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


>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>The End!!
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<