In [1]:
```
!pip install apyori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Building wheels for collected packages: apyori
  Building wheel for apyori (setup.py): started
  Building wheel for apyori (setup.py): finished with status 'done'
  Created wheel for apyori: filename=apyori-1.1.2-py3-none-any.whl size=5975
sha256=3c0fc5b514573942cb78f91cddf7dc6de69b049da4d377b7fccf495e4bd2723a
  Stored in directory: c:\users\aiml\appdata\local\pip\cache\wheels\77\3d\a6
\d317a6fb32be58a602b1e8c6b5d6f31f79322da554cad2a5ea
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:
```
from apyori import apriori
```

In [6]:
```
data = pd.read_csv("C:\\Users\\aiml\\Downloads\\my_movies.csv")
```

In [7]:
```
data.head()
```

Out[7]:

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

In [8]:
```
data.shape
```

Out[8]:
```
(10, 15)
```

In [11]:
```
#Preprocessing

records = []

for i in range(0, 10):
    records.append([str(data.values[i,j]) for j in range(0,15)])
```

In [12]:
```
#Apply Apriori alogorithm extract rules for each possible combination of items

association_rules = apriori(records, min_support = 0.0045, min_confidence=0.2,
association_results = list(association_rules)
```

In [13]:
```
print(len(association_results))
```

```
112
```

In [14]:
```
print(association_results[0])
```

```
RelationRecord(items=frozenset({'LOTR', 'Green Mile'}), support=0.1, ordered_
statistics=[OrderedStatistic(items_base=frozenset({'Green Mile'}), items_add=
frozenset({'LOTR'}), confidence=0.5, lift=5.0), OrderedStatistic(items_base=f
rozenset({'LOTR'}), items_add=frozenset({'Green Mile'}), confidence=1.0, lift
=5.0)])
```

In [19]:

```
#Calculate the support, the confidence and lift for each rule

#first index of inner list
#contains base item and add item
for item in association_results:
    pair = item[0]
    items = [x for x in pair]
    print("Rule: " + items[0] + "->" + items[1])


#second index of thr inner list
print("Support: " + str(item[1]))


#third index of the list located at 0th of the third index of inner list

print("Confidence: " + str(item[2][0][2]))
print("Lift: " + str(item[2][0][3]))

print("=====================================")
```

```
Rule: LOTR->Green Mile
Rule: Harry Potter2->Harry Potter1
Rule: LOTR1->LOTR2
Rule: 0->LOTR
Rule: 0->Harry Potter1
Rule: 0->LOTR2
Rule: LOTR->1
Rule: Harry Potter2->Harry Potter1
Rule: LOTR1->LOTR2
Rule: Gladiator->LOTR
Rule: LOTR1->Harry Potter1
Rule: LOTR2->Harry Potter1
Rule: Sixth Sense->Harry Potter1
Rule: Sixth Sense->LOTR
Rule: LOTR->nan
Rule: LOTR1->LOTR2
Rule: LOTR1->Sixth Sense
Rule: LOTR2->Sixth Sense
Rule: Harry Potter2->Harry Potter1
Rule: LOTR1->LOTR2
```

In [ ]: