In [1]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
```

In [2]:

```
import nltk
nltk.download('punkt')
from nltk.corpus import stopwords
import string
result = string.punctuation
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Ranjit\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [3]:

```
df = pd.read_csv('bollywoodmovies.csv')
df
```

Out[3]:

	Movie Name	Release Period	Whether Remake	Whether Franchise	Genre	New Actor	New Director	New Music Director	Lead Star
0	Golden Boys	Normal	No	No	suspense	Yes	No	No	Jeet Goswami
1	Kaccha Limboo	Holiday	No	No	drama	Yes	No	Yes	Karan Bhanushali
2	Not A Love Story	Holiday	No	No	thriller	No	No	No	Mahie Gill
3	Qaidi Band	Holiday	No	No	drama	Yes	No	No	Aadar Jain
4	Chaatwali	Holiday	No	No	adult	Yes	Yes	Yes	Aadil Khan
...	...	...	...	...	...	...	...	...	...
1693	Fight Club	Holiday	No	No	action	No	Yes	No	Zayed Khan
1694	Strings Of Paasion	Normal	No	No	drama	No	Yes	Yes	Zeenat Aman
1695	Dunno Y Na Jaane Kyun	Normal	No	No	drama	No	No	No	Zeenat Aman
1696	Taj Mahal - An Eternal Love Story	Normal	No	No	drama	No	Yes	No	Zulfi Sayed
1697	Mr. Hot Mr. Kool	Normal	No	No	comedy	No	No	Yes	Zulfi Sayed

1698 rows × 14 columns

In [ ]:

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Movie Name', 'Release Period', 'Whether Remake', 'Whether Franchis
e',
       'Genre', 'New Actor', 'New Director', 'New Music Director', 'Lead Sta
r',
       'Director', 'Music Director', 'Number of Screens', 'Revenue(INR)',
       'Budget(INR)'],
      dtype='object')
```

In [5]:

```
df.describe()
```

Out[5]:

	Number of Screens	Revenue(INR)	Budget(INR)
count	1698.000000	1.698000e+03	1.698000e+03
mean	553.831567	1.501674e+08	2.377287e+08
std	782.951839	2.434838e+08	6.134398e+08
min	1.000000	3.250000e+05	7.250000e+03
25%	30.000000	1.500000e+07	1.150000e+06
50%	200.000000	5.500000e+07	1.240000e+07
75%	800.000000	1.900000e+08	1.778325e+08
max	4600.000000	2.100000e+09	8.016120e+09

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1698 entries, 0 to 1697
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Movie Name          1698 non-null   object
 1   Release Period      1698 non-null   object
 2   Whether Remake      1698 non-null   object
 3   Whether Franchise   1698 non-null   object
 4   Genre               1698 non-null   object
 5   New Actor           1698 non-null   object
 6   New Director        1698 non-null   object
 7   New Music Director  1698 non-null   object
 8   Lead Star           1698 non-null   object
 9   Director            1698 non-null   object
 10  Music Director      1698 non-null   object
 11  Number of Screens   1698 non-null   int64 
 12  Revenue(INR)        1698 non-null   int64 
 13  Budget(INR)         1698 non-null   int64 
dtypes: int64(3), object(11)
memory usage: 185.8+ KB
```

In [7]:

```
df.isnull().sum()
```

Out[7]:

```
Movie Name            0
Release Period        0
Whether Remake        0
Whether Franchise     0
Genre                 0
New Actor             0
New Director          0
New Music Director    0
Lead Star             0
Director              0
Music Director        0
Number of Screens     0
Revenue(INR)          0
Budget(INR)           0
dtype: int64
```

In [8]:

```
# As there are fewer null values, it's better to drop these values
df.dropna(inplace=True)
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
Movie Name            0
Release Period        0
Whether Remake        0
Whether Franchise     0
Genre                 0
New Actor             0
New Director          0
New Music Director    0
Lead Star             0
Director              0
Music Director        0
Number of Screens     0
Revenue(INR)          0
Budget(INR)           0
dtype: int64
```

In [10]:

```
df.duplicated().sum()
```

Out[10]:

```
2
```

In [11]:

```
# dropping duplicat records
# df.drop_duplicates(keep = 'first',inplace=True)
```

In [12]:

```
df.duplicated().sum()
```

Out[12]:

```
2
```

In [13]:

```
df.columns
```

Out[13]:

```
Index(['Movie Name', 'Release Period', 'Whether Remake', 'Whether Franchis
e',
       'Genre', 'New Actor', 'New Director', 'New Music Director', 'Lead Sta
r',
       'Director', 'Music Director', 'Number of Screens', 'Revenue(INR)',
       'Budget(INR)'],
      dtype='object')
```

In [14]:

```
# removing columns that are not required
df.drop(['Release Period','New Actor','New Music Director','New Director','Number of Screen
```

In [15]:

```
df.tail()
```

Out[15]:

	Movie Name	Whether Remake	Whether Franchise	Genre	Lead Star	Director	Music Director
1693	Fight Club	No	No	action	Zayed Khan	Vikram Chopra	Pritam
1694	Strings Of Paasion	No	No	drama	Zeenat Aman	Sanghamitra Chaudhuri	Dev Sikdar
1695	Dunno Y Na Jaane Kyun	No	No	drama	Zeenat Aman	Sanjay Sharma	Nikhil
1696	Taj Mahal - An Eternal Love Story	No	No	drama	Zulfi Sayed	Akbar Khan	Naushad
1697	Mr. Hot Mr. Kool	No	No	comedy	Zulfi Sayed	Partho Ghosh	Rishi - Ranjit

In [ ]:

```

```

In [16]:

```
# converting column with values so that appropriate vectors of tags can be created
```

In [17]:

```
i=0
for word in df['Whether Remake']:
    if word=='No':
        df['Whether Remake'][i] = 'NotRemake'
    else:
        df['Whether Remake'][i] = 'Remake'
    i+=1
```

In [18]:

```
i=0
for word in df['Whether Franchise']:
    if word=='No':
        df['Whether Franchise'][i] = 'Franchise'
    else:
        df['Whether Franchise'][i] = 'NotFranchise'
    i+=1
```

In [19]:

```
def tag(text):
    s=""
    text = nltk.word_tokenize(text)
    for word in text:
        s+=word
    return s
```

In [20]:

```
df['Music Director'] = df['Music Director'].apply(lambda x:tag(x))
```

In [21]:

```
df['Director'] = df['Director'].apply(lambda x:tag(x))
```

In [22]:

```
df['Lead Star'] = df['Lead Star'].apply(lambda x:tag(x))
```

In [23]:

```
df.head()
```

Out[23]:

	Movie Name	Whether Remake	Whether Franchise	Genre	Lead Star	Director	Music Director
0	Golden Boys	NotRemake	Franchise	suspense	JeetGoswami	RaviVarma	BabaJagirdar
1	Kaccha Limboo	NotRemake	Franchise	drama	KaranBhanushali	SagarBallary	AmardeepNijjer
2	Not A Love Story	NotRemake	Franchise	thriller	MahieGill	RamGopalVerma	SandeepChowta
3	Qaidi Band	NotRemake	Franchise	drama	AadarJain	HabibFaisal	AmitTrivedi
4	Chaatwali	NotRemake	Franchise	adult	AadilKhan	AadilKhan	BablooUstad

In [24]:

```
df['Genre'].value_counts()
```

Out[24]:

```
drama           639
comedy          284
thriller        212
love_story      133
action          127
rom__com         95
adult            78
horror           53
suspense         30
masala           16
mythological     14
fantasy          13
animation         3
documentary       1
Name: Genre, dtype: int64
```

In [25]:

```
i=0
for word in df['Whether Franchise']:
    if word=='No':
        df['Whether Franchise'][i] = 'Franchise'
    else:
        df['Whether Franchise'][i] = 'NotFranchise'
    i+=1
```

In [26]:

```
df['text'] = ""
df.columns
```

Out[26]:

```
Index(['Movie Name', 'Whether Remake', 'Whether Franchise', 'Genre',
       'Lead Star', 'Director', 'Music Director', 'text'],
      dtype='object')
```

In [27]:

```
# combining all colunms to single column
for i in range(0,1695):
    df['text'] = df['Whether Remake']+ " " + df['Whether Franchise']+ " " + df['Genre']+ "
```

In [28]:

```
df['text']
```

Out[28]:

```
0       NotRemake NotFranchise suspense JeetGoswami Ra...
1       NotRemake NotFranchise drama KaranBhanushali S...
2       NotRemake NotFranchise thriller MahieGill RamG...
3       NotRemake NotFranchise drama AadarJain HabibFa...
4       NotRemake NotFranchise adult AadilKhan AadilKh...
                              ...
1693    NotRemake NotFranchise action ZayedKhan Vikram...
1694    NotRemake NotFranchise drama ZeenatAman Sangha...
1695    NotRemake NotFranchise drama ZeenatAman Sanjay...
1696    NotRemake NotFranchise drama ZulfiSayed AkbarK...
1697    NotRemake NotFranchise comedy ZulfiSayed Parth...
Name: text, Length: 1698, dtype: object
```

In [29]:

```
df
```

Out[29]:

	Movie Name	Whether Remake	Whether Franchise	Genre	Lead Star	Director	Mu
0	Golden Boys	NotRemake	NotFranchise	suspense	JeetGoswami	RaviVarma	B
1	Kaccha Limboo	NotRemake	NotFranchise	drama	KaranBhanushali	SagarBallary	Am
2	Not A Love Story	NotRemake	NotFranchise	thriller	MahieGill	RamGopalVerma	San
3	Qaidi Band	NotRemake	NotFranchise	drama	AadarJain	HabibFaisal	
4	Chaatwali	NotRemake	NotFranchise	adult	AadilKhan	AadilKhan	E
...	...	...	...	...	...	...	
1693	Fight Club	NotRemake	NotFranchise	action	ZayedKhan	VikramChopra	
1694	Strings Of Paasion	NotRemake	NotFranchise	drama	ZeenatAman	SanghamitraChaudhuri	
1695	Dunno Y Na Jaane Kyun	NotRemake	NotFranchise	drama	ZeenatAman	SanjaySharma	
1696	Taj Mahal - An Eternal Love Story	NotRemake	NotFranchise	drama	ZulfiSayed	AkbarKhan	
1697	Mr. Hot Mr. Kool	NotRemake	NotFranchise	comedy	ZulfiSayed	ParthoGhosh	

1698 rows × 8 columns

In [30]:

```
df.drop(['Whether Remake', 'Whether Franchise', 'Genre','Lead Star', 'Director', 'Music Dir
```

In [31]:

```
def TextTransform(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y =[]
    for i in text:
        if(i.isalnum()):
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if(i not in stopwords.words('english') and i not in string.punctuation and i.isnume
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))

    return " ".join(y)    # => this will return in series format
```

In [32]:

```
df['text'].apply(lambda x:TextTransform(x))
```

Out[32]:

```
0       notremak notfranchis suspens jeetgoswami raviv...
1       notremak notfranchis drama karanbhanushali sag...
2       notremak notfranchis thriller mahiegil ramgopa...
3       notremak notfranchis drama aadarjain habibfais...
4       notremak notfranchis adult aadilkhan aadilkhan...
                              ...
1693    notremak notfranchis action zayedkhan vikramch...
1694    notremak notfranchis drama zeenataman sanghami...
1695    notremak notfranchis drama zeenataman sanjaysh...
1696    notremak notfranchis drama zulfisay akbarkhan ...
1697    notremak notfranchis comedi zulfisay parthogho...
Name: text, Length: 1698, dtype: object
```

In [33]:

```
df['text'][0]
```

Out[33]:

```
'NotRemake NotFranchise suspense JeetGoswami RaviVarma BabaJagirdar Golden B
oys'
```

In [34]:

```
df['target'] = -1
```

In [35]:

```
df.isnull().sum()
```

Out[35]:

```
Movie Name    0
text          0
target        0
dtype: int64
```

In [36]:

```
index_dict = {} # this dictionary will store the movie name with its index
index_movie = {} # this dictionary will store the index wrt movie name
i=0
while i<1697:
    df['target'][i] = i
    i+=1
```

```
<ipython-input-36-3e1e38d1572d>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/
stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  df['target'][i] = i
```

In [37]:

```
for i in range(0,1696):
    if(i==683 or i==1269):
        continue
    index_dict[df['Movie Name'][i]] = df['target'][i]
    index_movie[df['target'][i]] = df['Movie Name'][i]
```

In [38]:

```
# df
index_dict[df['Movie Name'][1695]]
# index_movie[1695]
```

Out[38]:

```
1695
```

Model Building (by creating vectors)

In [39]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tv = TfidfVectorizer()
```

In [40]:

```
vectors = tv.fit_transform(df['text']).toarray()
```

In [41]:

```
vectors[0]
```

Out[41]:

```
array([0., 0., 0., ..., 0., 0., 0.])
```

In [42]:

```
from sklearn.metrics.pairwise import cosine_similarity
```

In [43]:

```
similarity = cosine_similarity(vectors)
```

In [44]:

```
# import bisect
# ls = list(enumerate(similarity[945])
# bisect.insort_right[ls]
```

In [ ]:

```

```

In [45]:

```
df.drop_duplicates(keep='first')
df['text'].duplicated().sum()
```

Out[45]:

```
3
```

In [46]:

```
ls1 = pd.DataFrame(sorted(list(enumerate(similarity[946])),reverse=True,key=lambda x:x[1])[
ls1[0]
```

Out[46]:

```
0     947
1    1367
2     848
3     484
4     588
Name: 0, dtype: int64
```

In [47]:

```
ls2 = pd.DataFrame(sorted(list(enumerate(similarity[945])),reverse=True,key=lambda x:x[1])[
ls2[0][0]
index_movie[ls2[0][0]]
```

Out[47]:

```
'Love Games'
```

In [ ]:

```

```

In [ ]:

```

```

In [68]:

```
def predict():
    movie = input("Enter Movie Name from the list:- ")
    try:
        movie_index = index_dict[movie]
        movie_index = int(movie_index)
        ls = pd.DataFrame(sorted(list(enumerate(similarity[movie_index])),reverse=True,key=
        for i in ls[0]:
            print(index_movie[i])
    except:
        print("No Record Found !!! ")
```

In [69]:

```
df[df['target']==946]['Movie Name']
```

Out[69]:

```
946    Bahubali - The Beginning
Name: Movie Name, dtype: object
```

In [70]:

```
index_dict['Bahubali - The Beginning']
```

Out[70]:

```
946
```

In [71]:

```
index_movie[946]
```

Out[71]:

```
'Bahubali - The Beginning'
```

In [72]:

```
predict()
```

```
Enter Movie Name from the list:- Bahubali - The Beginning
Bahubali 2 - The Conclusion
Paheli
Dhokha
Lahore
Rog
```

In [73]:

```
predict()
```

```
Enter Movie Name from the list:- Bahubali 2 - The Conclusion
Bahubali - The Beginning
Paheli
Dhokha
Lahore
Rog
```

In [74]:

```
predict()
```

```
Enter Movie Name from the list:- Dangal
Dhoom 3
Taare Zameen Par
Bhoothnath Returns
Chillar Party
3 Idiots
```

In [75]:

```
predict() # as this movie not in list it will not have any record
```

```
Enter Movie Name from the list:- Tanhaji
No Record Found !!!
```

In [ ]:

```

```