MOVIE RECOMMENDATON SYSTEM

NAME - KSHITIJ BHARDWAJ

SECTION - DS

CLASS ROLL NO. - 31

UNIV. ROLL NO. - 2013652

SUBJECT - BIG DATA STORAGE AND PROCESSING

In [3]:

```
import pandas as pd
import numpy as np
pd.set_option('display.max_colwidth', None)
pd.set_option("display.max_rows", None, "display.max_columns", None)
import warnings
warnings.filterwarnings("ignore")
```

In [4]:

```
movie = pd.read_csv('D:\\work\\1python for datascience\\ibm project\\imdb dataset\\movies.csv')
rating = pd.read_csv('D:\\work\\1python for datascience\\ibm project\\imdb dataset\\ratings.csv')
```

In [5]:

```
#let's take quick peak at our first dataset
movie.head()
```

Out[5]:

	imdb_title_id	title	original_title	year	date_published	genre	duration	coun
0	tt0000009	Miss Jerry	Miss Jerry	1894	1894-10-09	Romance	45	U
1	tt0000574	The Story of the Kelly Gang	The Story of the Kelly Gang	1906	1906-12-26	Biography, Crime, Drama	70	Austra
2	tt0001892	Den sorte drøm	Den sorte drøm	1911	1911-08-19	Drama	53	Germa Denm
3	tt0002101	Cleopatra	Cleopatra	1912	1912-11-13	Drama, History	100	U

	imdb_title_id	title	original_title	year	date_published	genre	duration	coun
4	tt0002130	L'Inferno	L'Inferno	1911	1911-03-06	Adventure, Drama, Fantasy	68	It

In [6]:

```
# let's check the no. of rows
movie.shape
```

Out[6]:

```
(85855, 22)
```

In [7]:

```
#let's take quick peak at our second dataset
rating.head()
```

Out[7]:

	imdb_title_id	weighted_average_vote	total_votes	mean_vote	median_vote	votes_10	v
0	tt0000009	5.9	154	5.9	6.0	12	
1	tt0000574	6.1	589	6.3	6.0	57	
2	tt0001892	5.8	188	6.0	6.0	6	
3	tt0002101	5.2	446	5.3	5.0	15	
4	tt0002130	7.0	2237	6.9	7.0	210	

In [8]:

```
# let's check the no. of rows
rating.shape
```

Out[8]:

```
(85855, 49)
```

In [9]:

```
#now we're gonna drop the colummns which are not required

movie.drop('original_title', inplace = True , axis = 1)
movie.drop('year', inplace = True , axis = 1)
movie.drop('date_published', inplace = True , axis = 1)
movie.drop('duration', inplace = True , axis = 1)
movie.drop('language', inplace = True , axis = 1)
movie.drop('director', inplace = True , axis = 1)
movie.drop('writer', inplace = True , axis = 1)
movie.drop('production_company', inplace = True , axis = 1)
movie.drop('actors', inplace = True , axis = 1)
movie.drop('description', inplace = True , axis = 1)
movie.drop('budget', inplace = True , axis = 1)
movie.drop('usa_gross_income', inplace = True , axis = 1)
movie.drop('worlwide_gross_income', inplace = True , axis = 1)
movie.drop('metascore', inplace = True , axis = 1)
movie.drop('reviews_from_users', inplace = True , axis = 1)
movie.drop('reviews_from_critics', inplace = True , axis = 1)
```

In [10]:

```python
# repeating the same for our second dataset.

rating.drop(rating.loc[:, 'votes_10':'non_us_voters_votes'].columns,
inplace =True, axis = 1)
```

In [11]:

```python
# for further processing we will merge these two datasets together.

train = pd.merge(left = movie , right = rating , how = "left" , left
_on = 'imdb_title_id' ,\
                right_on = 'imdb_title_id')
```

In [12]:

```python
#let's check the dataset now.
train.head()
```

Out[12]:

	imdb_title_id	title	genre	country	avg_vote	votes	weighted_average_vote
0	tt0000009	Miss Jerry	Romance	USA	5.9	154	5.9
1	tt0000574	The Story of the Kelly Gang	Biography, Crime, Drama	Australia	6.1	589	6.1
2	tt0001892	Den sorte drøm	Drama	Germany, Denmark	5.8	188	5.8
3	tt0002101	Cleopatra	Drama, History	USA	5.2	446	5.2
4	tt0002130	L'Inferno	Adventure, Drama, Fantasy	Italy	7.0	2237	7.0

In [13]:

```python
#to create dummies in our project we'll extract genres of the movies
and seperate them.
genres = set()

for i in range(0,len(train['genre'])):
    l=[]
    x = train['genre'][i].split(',')
    for y in x:
        y = y.strip()
        l.append(y)
        genres.add(y)
    train['genre'][i]=l
```

In [14]:

```
#let's look at the genres
genres
```

Out[14]:

```
{'Action',
 'Adult',
 'Adventure',
 'Animation',
 'Biography',
 'Comedy',
 'Crime',
 'Documentary',
 'Drama',
 'Family',
 'Fantasy',
 'Film-Noir',
 'History',
 'Horror',
 'Music',
 'Musical',
 'Mystery',
 'News',
 'Reality-TV',
 'Romance',
 'Sci-Fi',
 'Sport',
 'Thriller',
 'War',
 'Western'}
```

In [15]:

```
# now let's make them into columns

train['Action']=0
train['Adult']=0
train['Adventure']=0
train['Animation']=0
train['Biography']=0
train['Comedy']=0
train['Crime']=0
train['Drama']=0
train['Family']=0
train['Fantasy']=0
train['Film-Noir']=0
train['History']=0
train['Horror']=0
train['Music']=0
train['Musical']=0
train['Mystery']=0
train['News']=0
train['Reality-TV']=0
train['Romance']=0
train['Sci-Fi']=0
train['Sport']=0
train['Thriller']=0
train['War']=0
train['Western']=0
train['Documentary']=0
```

In [16]:

```
#here we wil fill these recently created columns with binary no.s
# if a film is of a specific genre, that genre will be marked as 1 and
#every non-matching one will be marked as 0.
for i in range(0,len(train['genre'])):
    for j in train['genre'][i]:
        for k in genres:
            if(j==k):
                train["%s" %k][i] = 1

train.drop('genre',inplace =True , axis =1 )
```

In [17]:

```
train.head()
```

Out[17]:

	imdb_title_id	title	country	avg_vote	votes	weighted_average_vote	total_votes
0	tt0000009	Miss Jerry	USA	5.9	154	5.9	154
1	tt0000574	The Story of the Kelly Gang	Australia	6.1	589	6.1	589
2	tt0001892	Den sorte drøm	Germany, Denmark	5.8	188	5.8	188
3	tt0002101	Cleopatra	USA	5.2	446	5.2	446
4	tt0002130	L'Inferno	Italy	7.0	2237	7.0	2237

In [18]:

```
# this function here is used to get the user prefered ratings.
def recommend1(gen,rate):
    recom = train[train[gen]==1]
    recom = recom[train['weighted_average_vote']>=rate]
    return recom
def recommend2(gen,gen1,rate):
    recom = train[train[gen]==1]
    recom = train[train[gen1]==1]
    recom = recom[train['weighted_average_vote']>=rate]
    return recom
```

In [21]:

```
 #recommendation time!
#here we will take the user favourite genre and how high ratings
#(s)he wants for his/her recommendation
#and it gets output in the form of a list of top 10 movies

rate= int(input("enter the rating threshold - "))
gen = input("enter the Genre you're interested in - ")
choice=input("do you want to enter another genre? (y/n)")
if(choice == 'y'):
    gen1 = input("enter another Genre you're interested in - ")
    rec = recommend2(gen,gen1,rate).head(10)
else:
    print("okay then! \n")
    rec = recommend1(gen,rate).head(10)
print(rec['title'])
```

```
enter the rating threshold - 7
enter the Genre you're interested in - Comedy
do you want to enter another genre? (y/n)y
enter another Genre you're interested in - Horror
79                        Pikovaya dama
164                        Per la patria
165    Il gabinetto del dottor Caligari
190               Dr. Jekyll e Mr. Hyde
196      Il Golem - Come venne al mondo
214               La vedova del pastore
252          Il carrettiere della morte
297              Nosferatu - Il vampiro
327              Il gobbo di Notre Dame
382                        Orlacs Hände
Name: title, dtype: object
```

In [ ]: