Movie Recommendation System

Import Libraries ¶

In [169]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Import Dataset for ID and Title

In [170]:

```
movie_title_df = pd.read_csv('F:\Data Scientist\ML projects in Resume\Movie
movie_title_df
```

Out[170]:

	item_id	title
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)
...	...	...
1677	1678	Mat' i syn (1997)
1678	1679	B. Monkey (1998)
1679	1680	Sliding Doors (1998)
1680	1681	You So Crazy (1994)
1681	1682	Scream of Stone (Schrei aus Stein) (1991)

1682 rows × 2 columns

Importing Dataset for User ID, Movie ID, Rating and Timestamp

In [171]:

```
movie_rating_df = pd.read_csv('F:\Data Scientist\ML projects in Resume\Movie
movie_rating_df
```

Out[171]:

	user_id	item_id	rating	timestamp
0	0	50	5	881250949
1	0	172	5	881250949
2	0	133	1	881250949
3	196	242	3	881250949
4	186	302	3	891717742
...	...	...	...	...
99998	880	476	3	880175444
99999	716	204	5	879795543
100000	276	1090	1	874795795
100001	13	225	2	882399156
100002	12	203	3	879959583

100003 rows × 4 columns

EDA For Movies Title

In [172]:

```
movie_title_df.head()
```

Out[172]:

	item_id	title
0	1	Toy Story (1995)
1	2	GoldenEye (1995)
2	3	Four Rooms (1995)
3	4	Get Shorty (1995)
4	5	Copycat (1995)

In [173]:

```
movie_title_df.tail()
```

Out[173]:

	item_id	title
1677	1678	Mat' i syn (1997)
1678	1679	B. Monkey (1998)
1679	1680	Sliding Doors (1998)
1680	1681	You So Crazy (1994)
1681	1682	Scream of Stone (Schrei aus Stein) (1991)

In [174]:

```
movie_title_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1682 entries, 0 to 1681
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   item_id  1682 non-null   int64 
 1   title    1682 non-null   object
dtypes: int64(1), object(1)
memory usage: 26.4+ KB
```

EDA For Movies Rating

In [175]:

```
movie_rating_df.head()
```

Out[175]:

	user_id	item_id	rating	timestamp
0	0	50	5	881250949
1	0	172	5	881250949
2	0	133	1	881250949
3	196	242	3	881250949
4	186	302	3	891717742

In [176]:

```
movie_rating_df.tail()
```

Out[176]:

	user_id	item_id	rating	timestamp
99998	880	476	3	880175444
99999	716	204	5	879795543
100000	276	1090	1	874795795
100001	13	225	2	882399156
100002	12	203	3	879959583

In [177]:

```
movie_rating_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100003 entries, 0 to 100002
Data columns (total 4 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   user_id    100003 non-null  int64
 1   item_id    100003 non-null  int64
 2   rating     100003 non-null  int64
 3   timestamp  100003 non-null  int64
dtypes: int64(4)
memory usage: 3.1 MB
```

In [178]:
```
movie_rating_df.describe()
```

Out[178]:

	user_id	item_id	rating	timestamp
count	100003.000000	100003.000000	100003.000000	1.000030e+05
mean	462.470876	425.520914	3.529864	8.835288e+08
std	266.622454	330.797791	1.125704	5.343791e+06
min	0.000000	1.000000	1.000000	8.747247e+08
25%	254.000000	175.000000	3.000000	8.794487e+08
50%	447.000000	322.000000	4.000000	8.828269e+08
75%	682.000000	631.000000	4.000000	8.882600e+08
max	943.000000	1682.000000	5.000000	8.932866e+08

Combine Two Dataset (This will be helpful in searching related movie on basis of movie title)

In [179]:
```
movie_rating_df_comb = pd.merge(movie_rating_df ,movie_title_df,on = 'item_i
```

In [180]:
```
movie_rating_df_comb
```

Out[180]:

	user_id	item_id	rating	timestamp	title
0	0	50	5	881250949	Star Wars (1977)
1	290	50	5	880473582	Star Wars (1977)
2	79	50	4	891271545	Star Wars (1977)
3	2	50	5	888552084	Star Wars (1977)
4	8	50	5	879362124	Star Wars (1977)
...	...	...	...	...	...
99998	840	1674	4	891211682	Mamma Roma (1962)
99999	655	1640	3	888474646	Eighth Day, The (1996)
100000	655	1637	3	888984255	Girls Town (1996)
100001	655	1630	3	887428735	Silence of the Palace, The (Saimt el Qusur) (1...
100002	655	1641	3	887427810	Dadetown (1995)

Droping unnecessary field (timestamp)

In [181]:
```
movie_rating_df_comb.drop('timestamp',axis=1,inplace=True)
```

```
In [182]: movie_rating_df_comb
```

```
Out[182]:
        user_id  item_id  rating                                              title
0             0       50       5                                   Star Wars (1977)
1           290       50       5                                   Star Wars (1977)
2            79       50       4                                   Star Wars (1977)
3             2       50       5                                   Star Wars (1977)
4             8       50       5                                   Star Wars (1977)
...         ...      ...     ...                                                ...
99998       840     1674       4                                 Mamma Roma (1962)
99999       655     1640       3                             Eighth Day, The (1996)
100000      655     1637       3                                 Girls Town (1996)
100001      655     1630       3  Silence of the Palace, The (Saimt el Qusur) (1...
100002      655     1641       3                                   Dadetown (1995)
```

Creating a new dataset for Title on basis of Rating

```
In [183]: dataset = movie_rating_df_comb.groupby('title')['rating'].describe()
```

```
In [184]: dataset
```

```
Out[184]:
                                    count      mean       std  min   25%  50%  75%  max
                             title
          'Til There Was You (1997)   9.0  2.333333  1.000000  1.0  2.00  2.0  3.0  4.0
                       1-900 (1994)   5.0  2.600000  1.516575  1.0  1.00  3.0  4.0  4.0
              101 Dalmatians (1996) 109.0  2.908257  1.076184  1.0  2.00  3.0  4.0  5.0
                12 Angry Men (1957) 125.0  4.344000  0.719588  2.0  4.00  4.0  5.0  5.0
                         187 (1997)  41.0  3.024390  1.172344  1.0  2.00  3.0  4.0  5.0
                                ...   ...       ...       ...  ...   ...  ...  ...  ...
               Young Guns II (1990)  44.0  2.772727  1.008421  1.0  2.00  3.0  3.0  5.0
  Young Poisoner's Handbook, The
                             (1995)  41.0  3.341463  1.237129  1.0  3.00  4.0  4.0  5.0
          Zeus and Roxanne (1997)   6.0  2.166667  0.983192  1.0  1.25  2.5  3.0  3.0
                            unknown   9.0  3.444444  1.130388  1.0  3.00  4.0  4.0  5.0
 Á köldum klaka (Cold Fever) (1994)   1.0  3.000000       NaN  3.0  3.00  3.0  3.0  3.0

1664 rows × 8 columns
```

```
In [185]: dataset = dataset.reset_index()
```

In [186]:
```
dataset
```

Out[186]:

	title	count	mean	std	min	25%	50%	75%	max
0	'Til There Was You (1997)	9.0	2.333333	1.000000	1.0	2.00	2.0	3.0	4.0
1	1-900 (1994)	5.0	2.600000	1.516575	1.0	1.00	3.0	4.0	4.0
2	101 Dalmatians (1996)	109.0	2.908257	1.076184	1.0	2.00	3.0	4.0	5.0
3	12 Angry Men (1957)	125.0	4.344000	0.719588	2.0	4.00	4.0	5.0	5.0
4	187 (1997)	41.0	3.024390	1.172344	1.0	2.00	3.0	4.0	5.0
...	...	...	...	...	...	...	...	...	...
1659	Young Guns II (1990)	44.0	2.772727	1.008421	1.0	2.00	3.0	3.0	5.0
1660	Young Poisoner's Handbook, The (1995)	41.0	3.341463	1.237129	1.0	3.00	4.0	4.0	5.0
1661	Zeus and Roxanne (1997)	6.0	2.166667	0.983192	1.0	1.25	2.5	3.0	3.0
1662	unknown	9.0	3.444444	1.130388	1.0	3.00	4.0	4.0	5.0
1663	Á köldum klaka (Cold Fever)	1.0	3.000000	NaN	3.0	3.00	3.0	3.0	3.0

Keeping only the important fields in the dataset

In [187]:
```
dataset = dataset[['title','count','mean']]
```

In [188]:
```
dataset
```

Out[188]:

	title	count	mean
0	'Til There Was You (1997)	9.0	2.333333
1	1-900 (1994)	5.0	2.600000
2	101 Dalmatians (1996)	109.0	2.908257
3	12 Angry Men (1957)	125.0	4.344000
4	187 (1997)	41.0	3.024390
...	...	...	...
1659	Young Guns II (1990)	44.0	2.772727
1660	Young Poisoner's Handbook, The (1995)	41.0	3.341463
1661	Zeus and Roxanne (1997)	6.0	2.166667
1662	unknown	9.0	3.444444
1663	Á köldum klaka (Cold Fever) (1994)	1.0	3.000000

1664 rows × 3 columns

Modelling

In [189]:
```
matrix = movie_rating_df_comb.pivot_table(index='user_id',columns='title',va
```

In [190]:
```
matrix
```

Out[190]:

title	'Til There Was You (1997)	1-900 (1994)	101 Dalmatians (1996)	12 Angry Men (1957)	187 (1997)	2 Days in the Valley (1996)	20,000 Leagues Under the Sea (1954)	2001: A Space Odyssey (1968)	3 Ninjas: High Noon At Mega Mountain (1998)	S (
user_id										
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	2.0	5.0	NaN	NaN	3.0	4.0	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1.0	
3	NaN	NaN	NaN	NaN	2.0	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
939	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Taking the Title of Movie for which recommendation system will work

In [191]:
```
x = matrix['Year of the Horse (1997)']
```

In [192]:
```
x
```

Out[192]:
```
user_id
0      NaN
1      NaN
2      NaN
3      NaN
4      NaN
        ..
939    NaN
940    NaN
941    NaN
942    NaN
943    NaN
Name: Year of the Horse (1997), Length: 944, dtype: float64
```

Finding correlation between Matrix and x and creating a new column 'Correlation value' for keeping the correlation values

In [193]:
```
x_correlations = pd.DataFrame(matrix.corrwith(x),columns=['Correlation value
```

```
C:\Users\HP_9046\anaconda3\lib\site-packages\numpy\lib\function_base.py:26
83: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
C:\Users\HP_9046\anaconda3\lib\site-packages\numpy\lib\function_base.py:25
42: RuntimeWarning: divide by zero encountered in true_divide
  c *= np.true_divide(1, fact)
```

In [194]:

```
x_correlations
```

Out[194]:

	Correlation value
title	
'Til There Was You (1997)	NaN
1-900 (1994)	NaN
101 Dalmatians (1996)	-1.000000
12 Angry Men (1957)	NaN
187 (1997)	0.866025
...	...
Young Guns II (1990)	NaN
Young Poisoner's Handbook, The (1995)	NaN
Zeus and Roxanne (1997)	NaN
unknown	NaN

Dropping not available values

In [195]:

```
x_correlations.dropna(inplace=True)
```

In [196]:

```
x_correlations
```

Out[196]:

	Correlation value
title	
101 Dalmatians (1996)	-1.000000
187 (1997)	0.866025
39 Steps, The (1935)	1.000000
Adventures of Pinocchio, The (1996)	1.000000
Adventures of Priscilla, Queen of the Desert, The (1994)	1.000000
...	...
Wolf (1994)	1.000000
Wonderland (1997)	0.666667
Year of the Horse (1997)	1.000000
Young Frankenstein (1974)	-1.000000

Merging dataset table with x_correlation table

In [197]:

```
x_correlations = pd.merge(x_correlations, dataset, on = 'title')
```

In [198]:
```
x_correlations
```

Out[198]:

	title	Correlation value	count	mean
0	101 Dalmatians (1996)	-1.000000	109.0	2.908257
1	187 (1997)	0.866025	41.0	3.024390
2	39 Steps, The (1935)	1.000000	59.0	4.050847
3	Adventures of Pinocchio, The (1996)	1.000000	39.0	3.051282
4	Adventures of Priscilla, Queen of the Desert, ...	1.000000	111.0	3.594595
...	...	...	...	...
343	Wolf (1994)	1.000000	67.0	2.701493
344	Wonderland (1997)	0.666667	10.0	3.200000
345	Year of the Horse (1997)	1.000000	7.0	3.285714
346	Young Frankenstein (1974)	-1.000000	200.0	3.945000
347	Young Guns (1988)	1.000000	101.0	3.207921

Sorting Movies on basis of correlation value, Higher correlation means better match

In [199]:
```
x_correlations = x_correlations.sort_values('Correlation value',ascending=Fa
```

In [200]:
```
x_correlations
```

Out[200]:

	title	Correlation value	count	mean
347	Young Guns (1988)	1.0	101.0	3.207921
277	Seventh Seal, The (Sjunde inseglet, Det) (1957)	1.0	72.0	3.541667
118	Frighteners, The (1996)	1.0	115.0	3.234783
119	From Dusk Till Dawn (1996)	1.0	92.0	3.119565
120	Fugitive, The (1993)	1.0	336.0	4.044643
...	...	...	...	...
318	Tomorrow Never Dies (1997)	-1.0	180.0	3.427778
319	Top Gun (1986)	-1.0	220.0	3.481818
124	Gandhi (1982)	-1.0	195.0	4.020513
223	My Favorite Year (1982)	-1.0	62.0	3.532258
147	Hoop Dreams (1994)	-1.0	117.0	4.094017

In addition to high correlation, high count for review is also required for better recommendation, assuming only movies with review count more than 80 will be considered.

In [201]:

```
x_correlations[x_correlations['count']>=80].head()
```

Out[201]:

	title	Correlation value	count	mean
347	Young Guns (1988)	1.0	101.0	3.207921
118	Frighteners, The (1996)	1.0	115.0	3.234783
119	From Dusk Till Dawn (1996)	1.0	92.0	3.119565
120	Fugitive, The (1993)	1.0	336.0	4.044643
314	Titanic (1997)	1.0	350.0	4.245714