MOVIELENS CASE STUDY

DATA ANALYSIS AND PREDICTION MODEL

MODULES USED: NUMPY AND PANDAS

In [ ]:

```
# Movielens case study
```

In [3]:

```
import os
```

In [9]:

```
pwd
```

Out[9]:

```
'/home/labsuser/Desktop'
```

In [4]:

```
import pandas as pd
import numpy as np
```

In [3]:

```
# Analysing the data
# Task 1
# Importing the three Datasets
```

In [5]:

```
data_movies=pd.read_csv('movies.dat',sep='::',engine='python',names=['MovieID', 'Title', 'Genres'])
data_users=pd.read_csv('users.dat',sep='::',engine='python',names=['UserID', 'Gender', 'Age','Occupation','Zip-code'])
data_ratings=pd.read_csv('ratings.dat',sep='::',engine='python',names=['UserID', 'MovieID', 'Rating','Timestamp'])
```

In [6]:

```
data_movies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3883 entries, 0 to 3882
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   MovieID  3883 non-null   int64
 1   Title    3883 non-null   object
 2   Genres   3883 non-null   object
dtypes: int64(1), object(2)
memory usage: 91.1+ KB
```

In [14]:

```
data_ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000209 entries, 0 to 1000208
Data columns (total 4 columns):
 #   Column     Non-Null Count    Dtype
```

```
 0   UserID     1000209 non-null  int64
 1   MovieID    1000209 non-null  int64
 2   Rating     1000209 non-null  int64
 3   Timestamp  1000209 non-null  int64
dtypes: int64(4)
memory usage: 30.5 MB
```

In [5]:

```
data_ratings['Timestamp']=pd.to_datetime(data_ratings["Timestamp"],unit='s')
data_ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000209 entries, 0 to 1000208
Data columns (total 4 columns):
 #   Column     Non-Null Count    Dtype
---  ------     --------------    -----
 0   UserID     1000209 non-null  int64
 1   MovieID    1000209 non-null  int64
 2   Rating     1000209 non-null  int64
 3   Timestamp  1000209 non-null  datetime64[ns]
dtypes: datetime64[ns](1), int64(3)
memory usage: 30.5 MB
```

In [16]:

```
data_movies
```

Out[16]:

	MovieID	Title	Genres
0	1	Toy Story (1995)	Animation|Children's|Comedy
1	2	Jumanji (1995)	Adventure|Children's|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
3878	3948	Meet the Parents (2000)	Comedy
3879	3949	Requiem for a Dream (2000)	Drama
3880	3950	Tigerland (2000)	Drama
3881	3951	Two Family House (2000)	Drama
3882	3952	Contender, The (2000)	Drama|Thriller

3883 rows × 3 columns

In [17]:

```
# Creating new dataset by merging
```

In [10]:

```
merge_movie_ratings = pd.merge(data_movies, data_ratings, on='MovieID')
merge_movie_ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000209 entries, 0 to 1000208
Data columns (total 6 columns):
 #   Column     Non-Null Count    Dtype
---  ------     --------------    -----
 0   MovieID    1000209 non-null  int64
 1   Title      1000209 non-null  object
 2   Genres     1000209 non-null  object
 3   UserID     1000209 non-null  int64
 4   Rating     1000209 non-null  int64
 5   Timestamp  1000209 non-null  int64
dtypes: int64(4), object(2)
memory usage: 53.4+ MB
```

In [8]:

```
from datetime import datetime
```

```python
import time
def convert(seconds):
    return time.strftime("%H:%M:%S", time.gmtime(n))
```

In [11]:

```python
merge_movie_ratings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000209 entries, 0 to 1000208
Data columns (total 6 columns):
 #   Column     Non-Null Count    Dtype
---  ------     --------------    -----
 0   MovieID    1000209 non-null  int64
 1   Title      1000209 non-null  object
 2   Genres     1000209 non-null  object
 3   UserID     1000209 non-null  int64
 4   Rating     1000209 non-null  int64
 5   Timestamp  1000209 non-null  int64
dtypes: int64(4), object(2)
memory usage: 53.4+ MB
```

In [12]:

```python
merge_movie_ratings.head()
```

Out[12]:

	MovieID	Title	Genres	UserID	Rating	Timestamp
0	1	Toy Story (1995)	Animation|Children's|Comedy	1	5	978824268
1	1	Toy Story (1995)	Animation|Children's|Comedy	6	4	978237008
2	1	Toy Story (1995)	Animation|Children's|Comedy	8	4	978233496
3	1	Toy Story (1995)	Animation|Children's|Comedy	9	5	978225952
4	1	Toy Story (1995)	Animation|Children's|Comedy	10	5	978226474

In [13]:

```python
master_data=pd.merge(merge_movie_ratings,data_users, on='UserID')
```

In [14]:

```python
master_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000209 entries, 0 to 1000208
Data columns (total 10 columns):
 #   Column      Non-Null Count    Dtype
---  ------      --------------    -----
 0   MovieID     1000209 non-null  int64
 1   Title       1000209 non-null  object
 2   Genres      1000209 non-null  object
 3   UserID      1000209 non-null  int64
 4   Rating      1000209 non-null  int64
 5   Timestamp   1000209 non-null  int64
 6   Gender      1000209 non-null  object
 7   Age         1000209 non-null  int64
 8   Occupation  1000209 non-null  int64
 9   Zip-code    1000209 non-null  object
dtypes: int64(6), object(4)
memory usage: 83.9+ MB
```

In [15]:

```python
master_data.head()
```

Out[15]:

	MovieID	Title	Genres	UserID	Rating	Timestamp	Gender	Age	Occupation	Zip-code
	MovieID	Title	Genres	UserID	Rating	Timestamp	Gender	Age	Occupation	Zip-code
0	1	Toy Story (1995)	Animation|Children's|Comedy	1	5	978824268	F	1	10	48067
1	48	Pocahontas (1995)	Animation|Children's|Musical|Romance	1	5	978824351	F	1	10	48067
2	150	Apollo 13	Drama	1	5	978301777	F	1	10	48067
3	260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Fantasy|Sci-Fi	1	4	978300760	F	1	10	48067
4	527	Schindler's List (1993)	Drama|War	1	5	978824195	F	1	10	48067

In [16]:

```
master_data.describe()
```

Out[16]:

	MovieID	UserID	Rating	Timestamp	Age	Occupation
count	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06
mean	1.865540e+03	3.024512e+03	3.581564e+00	9.722437e+08	2.973831e+01	8.036138e+00
std	1.096041e+03	1.728413e+03	1.117102e+00	1.215256e+07	1.175198e+01	6.531336e+00
min	1.000000e+00	1.000000e+00	1.000000e+00	9.567039e+08	1.000000e+00	0.000000e+00
25%	1.030000e+03	1.506000e+03	3.000000e+00	9.653026e+08	2.500000e+01	2.000000e+00
50%	1.835000e+03	3.070000e+03	4.000000e+00	9.730180e+08	2.500000e+01	7.000000e+00
75%	2.770000e+03	4.476000e+03	4.000000e+00	9.752209e+08	3.500000e+01	1.400000e+01
max	3.952000e+03	6.040000e+03	5.000000e+00	1.046455e+09	5.600000e+01	2.000000e+01

In [23]:

```
# visual representation of User Age distribution
```

In [18]:

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib
```

```
Using matplotlib backend: Qt5Agg
```

In [25]:

```
# user age distribution
plt.figure(figsize=(10,8))
master_data['Age'].hist(bins=40)
plt.title("Age distribution of Users")
plt.xlabel("Age")
plt.ylabel("No. of Users")
plt.show()
```

In [26]:

```
#User rating of the movie "Toy Story"
```

In [26]:

```
Toy_story=master_data[master_data['MovieID']==1]
Toy_story.head()
```

Out[26]:

	MovieID	Title	Genres	UserID	Rating	Timestamp	Gender	Age	Occupation	Zip-code
0	1	Toy Story (1995)	Animation|Children's|Comedy	1	5	978824268	F	1	10	48067
53	1	Toy Story (1995)	Animation|Children's|Comedy	6	4	978237008	F	50	9	55117
124	1	Toy Story (1995)	Animation|Children's|Comedy	8	4	978233496	M	25	12	11413
263	1	Toy Story (1995)	Animation|Children's|Comedy	9	5	978225952	M	25	17	61614
369	1	Toy Story (1995)	Animation|Children's|Comedy	10	5	978226474	F	35	1	95370

In [30]:

```
plt.figure(figsize=(8,6))
Toy_story['Rating'].hist(bins=30)
plt.title("User_Rating for Toy Story")
plt.xlabel("Rating")
plt.ylabel("No of users")
```

Out[30]:

```
Text(0, 0.5, 'No of users')
```

In [31]:

```
master_data['Rating'][master_data['Title']=="Toy Story (1995)"].value_counts()
```

Out[31]:

```
4    835
5    820
3    345
2     61
1     16
Name: Rating, dtype: int64
```

In [30]:

```
#Top 25 movies by viewership rating
```

In [33]:

```
rating_avg = merge_movie_ratings.groupby('Title')['Rating'].mean()
rating_avg.head()
```

Out[33]:

```
Title
$1,000,000 Duck (1971)          3.027027
'Night Mother (1986)            3.371429
'Til There Was You (1997)       2.692308
'burbs, The (1989)              2.910891
...And Justice for All (1979)   3.713568
Name: Rating, dtype: float64
```

In [34]:

```
rating_avg = rating_avg.sort_values(ascending=False)
rating_avg.head()
```

Out[34]:

```
Title
Gate of Heavenly Peace, The (1995)     5.0
Lured (1947)                           5.0
```

Ulysses (Ulisse) (1954)	5.0

```
Smashing Time (1967)                          5.0
Follow the Bitch (1998)                       5.0
Name: Rating, dtype: float64
```

In [36]:

```
rating_count = master_data.groupby('Title')['Rating']
rating_count = rating_count.count().sort_values(ascending=False)
rating_count[:25]
```

Out[36]:

```
Title
American Beauty (1999)                                        3428
Star Wars: Episode IV - A New Hope (1977)                     2991
Star Wars: Episode V - The Empire Strikes Back (1980)         2990
Star Wars: Episode VI - Return of the Jedi (1983)             2883
Jurassic Park (1993)                                          2672
Saving Private Ryan (1998)                                    2653
Terminator 2: Judgment Day (1991)                             2649
Matrix, The (1999)                                            2590
Back to the Future (1985)                                     2583
Silence of the Lambs, The (1991)                              2578
Men in Black (1997)                                           2538
Raiders of the Lost Ark (1981)                                2514
Fargo (1996)                                                  2513
Sixth Sense, The (1999)                                       2459
Braveheart (1995)                                             2443
Shakespeare in Love (1998)                                    2369
Princess Bride, The (1987)                                    2318
Schindler's List (1993)                                       2304
L.A. Confidential (1997)                                      2288
Groundhog Day (1993)                                          2278
E.T. the Extra-Terrestrial (1982)                             2269
Star Wars: Episode I - The Phantom Menace (1999)              2250
Being John Malkovich (1999)                                   2241
Shawshank Redemption, The (1994)                              2227
Godfather, The (1972)                                         2223
Name: Rating, dtype: int64
```

In [37]:

```
rating_avg_count = pd.DataFrame(data=rating_avg)
rating_avg_count['number_of_ratings'] = pd.DataFrame(rating_count)
rating_avg_count.head()
```

Out[37]:

	Rating	number_of_ratings
Title		
Gate of Heavenly Peace, The (1995)	5.0	3
Lured (1947)	5.0	1
Ulysses (Ulisse) (1954)	5.0	1
Smashing Time (1967)	5.0	2
Follow the Bitch (1998)	5.0	1

In [38]:

```
rating_avg_count.describe()
```

Out[38]:

	Rating	number_of_ratings
count	3706.000000	3706.000000
mean	3.238892	269.889099

	Rating	number_of_ratings
std	0.672925	384.047838
min	1.000000	1.000000
25%	2.822705	33.000000
50%	3.331546	123.500000
75%	3.740741	350.000000
max	5.000000	3428.000000

In [39]:

```
# Top 25 movies by viewership rating excluding movies with less than 10 ratings

filter_data = rating_avg_count[rating_avg_count['number_of_ratings'] > 10]
filter_data[:25]
```

Out[39]:

Title	Rating	number_of_ratings
Sanjuro (1962)	4.608696	69
Seven Samurai (The Magnificent Seven) (Shichinin no samurai) (1954)	4.560510	628
Shawshank Redemption, The (1994)	4.554558	2227
Godfather, The (1972)	4.524966	2223
Close Shave, A (1995)	4.520548	657
Usual Suspects, The (1995)	4.517106	1783
Schindler's List (1993)	4.510417	2304
Wrong Trousers, The (1993)	4.507937	882
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	4.491489	470
Raiders of the Lost Ark (1981)	4.477725	2514
Rear Window (1954)	4.476190	1050
Paths of Glory (1957)	4.473913	230
Star Wars: Episode IV - A New Hope (1977)	4.453694	2991
Third Man, The (1949)	4.452083	480
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1963)	4.449890	1367
For All Mankind (1989)	4.444444	27
Wallace & Gromit: The Best of Aardman Animation (1996)	4.426941	438
To Kill a Mockingbird (1962)	4.425647	928
Double Indemnity (1944)	4.415608	551
Casablanca (1942)	4.412822	1669
World of Apu, The (Apur Sansar) (1959)	4.410714	56
Sixth Sense, The (1999)	4.406263	2459
Yojimbo (1961)	4.404651	215
Pather Panchali (1955)	4.404255	47
Lawrence of Arabia (1962)	4.401925	831

In [32]:

```
# the ratings for all the movies reviewed by for a particular user of user id = 2696
```

In [45]:

```
user_2696 = master_data[master_data['UserID'] == 2696]
user_2696
```

Out[45]:

	MovieID	Title	Genres	UserID	Rating	Timestamp	Gender	Age	Occupation	Zip-code
991035	350	Client, The (1994)	Drama|Mystery|Thriller	2696	3	973308886	M	25	7	24210
991036	800	Lone Star (1996)	Drama|Mystery	2696	5	973308842	M	25	7	24210
991037	1092	Basic Instinct (1992)	Mystery|Thriller	2696	4	973308886	M	25	7	24210
991038	1097	E.T. the Extra-Terrestrial (1982)	Children's|Drama|Fantasy|Sci-Fi	2696	3	973308690	M	25	7	24210
991039	1258	Shining, The (1980)	Horror	2696	4	973308710	M	25	7	24210
991040	1270	Back to the Future (1985)	Comedy|Sci-Fi	2696	2	973308676	M	25	7	24210
991041	1589	d 7)		2696	3	973308865	M	25	7	24210
991042	1617	L.A. Confidential (1997)	Crime|Film-Noir|Mystery|Thriller	2696	4	973308842	M	25	7	24210
991043	1625	Game, The (1997)	Mystery|Thriller	2696	4	973308842	M	25	7	24210
991044	1644	I Know What You Did Last Summer (1997)	Horror|Mystery|Thriller	2696	2	973308920	M	25	7	24210
991045	1645	Devil's Advocate, The (1997)	Crime|Horror|Mystery|Thriller	2696	4	973308904	M	25	7	24210
991046	1711	Midnight in the Garden of Good and Evil (1997)	Comedy|Crime|Drama|Mystery	2696	4	973308904	M	25	7	24210
991047	1783	Palmetto	Film-Noir|Mystery|Thriller	2696	4	973308865	M	25	7	24210
991048	1805	Wild Things (1998)	Crime|Drama|Mystery|Thriller	2696	4	973308886	M	25	7	24210
991049	1892	Murder, A (1998)	Mystery|Thriller	2696	4	973308904	M	25	7	24210
991050	2338	I Still Know What You Did Last Summer (1998)	Horror|Mystery|Thriller	2696	2	973308920	M	25	7	24210
991051	2389	Psycho (1998)	Crime|Horror|Thriller	2696	4	973308710	M	25	7	24210
991052	2713	Lake Placid (1999)	Horror|Thriller	2696	1	973308710	M	25	7	24210
991053	3176	Talented Mr. Ripley, The (1999)	Drama|Mystery|Thriller	2696	4	973308865	M	25	7	24210
991054	3386	JFK (1991)	Drama|Mystery	2696	1	973308842	M	25	7	24210

In [38]:

```
# Unique Genres
```

In [48]:

```
merge_movie_ratings['Genres'].value_counts().head()
```

Out[48]:

```
Comedy            116883
Drama             111423
Comedy|Romance     42712
Comedy|Drama       42245
Drama|Romance      29170
Name: Genres, dtype: int64
```

In [49]:

```
master_data['Genres'].unique().tolist()
```

Out[49]:

```
["Animation|Children's|Comedy",
 "Animation|Children's|Musical|Romance",
 'Drama',
 'Action|Adventure|Fantasy|Sci-Fi',
 'Drama|War',
 "Children's|Drama",
 "Animation|Children's|Comedy|Musical",
 "Animation|Children's|Musical",
 'Crime|Drama|Thriller',
 'Animation',
 'Animation|Comedy|Thriller',
 'Musical|Romance',
 "Adventure|Children's|Drama|Musical",
 'Musical',
 "Children's|Comedy|Musical",
 "Children's|Drama|Fantasy|Sci-Fi",
 'Action|Adventure|Comedy|Romance',
 'Comedy|Sci-Fi',
 'Action|Adventure|Drama',
 "Adventure|Animation|Children's|Comedy|Musical",
 'Drama|Romance',
 "Animation|Children's",
 'Action|Drama|War',
 'Comedy',
 'Romance',
 'Action|Crime|Romance',
 'Thriller',
 'Comedy|Fantasy',
 'Comedy|Drama',
 "Children's|Comedy|Drama",
 'Drama|Musical',
 'Drama|Romance|War|Western',
 'Crime|Drama',
 'Action|Comedy|Western',
 'Action|Romance|Thriller',
 'Western',
 "Children's|Comedy",
 'Adventure|Drama|Western',
 'Comedy|Romance',
 'Comedy|Drama|Romance',
 'Drama|Romance|War',
 "Children's|Comedy|Western",
 "Adventure|Animation|Children's|Musical",
 'Action|Romance',
 'Action|Adventure|Romance|Sci-Fi|War',
 'Comedy|Musical|Romance',
 'Drama|Romance|Thriller',
 "Adventure|Children's|Comedy",
 'Action|Adventure|Romance',
 "Children's|Fantasy|Musical",
 "Animation|Children's|Comedy|Musical|Romance",
 'Comedy|Fantasy|Romance',
```

```
    'Action|Drama',
    'Comedy|Musical',
    'Action',
    'Adventure|Drama|Romance|Sci-Fi',
    'Action|Crime',
    'Drama|Thriller',
    'Drama|Sci-Fi',
    'Action|Crime|Drama',
    'Drama|Thriller|War',
    'Drama|Horror',
    'Action|Thriller',
    'Action|Adventure|Thriller',
    'Action|Adventure|Sci-Fi',
    'Action|Sci-Fi|Thriller',
    'Animation|Sci-Fi',
    'Adventure|Animation|Sci-Fi|Thriller',
    'Action|Drama|Romance',
    'Action|Drama|Thriller|War',
    'Action|Adventure|Comedy|Sci-Fi',
    'Crime|Drama|Mystery',
    'Drama|Sci-Fi|Thriller',
    'Comedy|Crime|Drama|Mystery',
    'Action|Comedy|Drama',
    'Action|Crime|Thriller',
    "Adventure|Children's|Drama",
    'Drama|Mystery',
    'Action|Comedy|Sci-Fi|Thriller',
    'Action|Adventure|Sci-Fi|Thriller',
    'Action|Drama|Romance|Thriller',
    'Crime|Thriller',
    'Documentary',
    'Comedy|Crime|Fantasy',
    'Animation|Comedy',
    'Comedy|Crime',
    'Crime|Film-Noir|Mystery|Thriller',
    'Sci-Fi|Thriller',
    'Action|Sci-Fi',
    'Horror|Sci-Fi|Thriller',
    "Adventure|Children's|Fantasy",
    'Action|Adventure|Comedy|Crime',
    'Action|Adventure',
    'Action|Drama|Thriller',
    "Children's|Comedy|Fantasy",
    'Comedy|Romance|War',
    'Film-Noir|Sci-Fi',
    'Comedy|Romance|Thriller',
    'Action|Adventure|Crime|Drama',
    'Action|Adventure|Mystery',
    'Action|Adventure|Fantasy',
    'Sci-Fi|War',
    'Action|Sci-Fi|War',
    'Mystery|Thriller',
    'Film-Noir|Mystery',
    'Drama|Mystery|Sci-Fi|Thriller',
    'Action|Adventure|Romance|War',
    "Adventure|Children's",
    "Adventure|Children's|Fantasy|Sci-Fi",
    "Adventure|Children's|Musical",
    "Adventure|Children's|Comedy|Fantasy",
    'Action|Adventure|Drama|Sci-Fi|War',
    'Action|Sci-Fi|Thriller|War',
    'Action|Western',
    'Adventure|War',
    'Action|Horror|Sci-Fi|Thriller',
    'Action|Adventure|Comedy|Horror|Sci-Fi',
    'Action|Comedy|Musical',
    'Film-Noir|Mystery|Thriller',
    'Adventure',
    'Comedy|War',
    'Adventure|Comedy|Drama',
    'Comedy|Mystery|Thriller',
    'Comedy|Horror',
```

```
    'Horror|Romance',
    'Horror',
    'Action|Horror',
    'Action|Romance|War',
    "Children's|Fantasy",
    "Children's|Drama|Fantasy",
    'Action|Adventure|Sci-Fi|War',
    'Action|Horror|Sci-Fi',
    'Action|Comedy|Crime|Drama',
    'War',
    'Comedy|Sci-Fi|Western',
    'Fantasy|Sci-Fi',
    "Action|Adventure|Children's|Comedy",
    "Adventure|Children's|Drama|Romance",
    "Adventure|Children's|Sci-Fi",
    "Children's",
    "Adventure|Children's|Comedy|Fantasy|Sci-Fi",
    "Animation|Children's|Fantasy|Musical",
    "Children's|Sci-Fi",
    'Adventure|Comedy',
    'Adventure|Musical',
    "Animation|Children's|Drama|Fantasy",
    "Children's|Fantasy|Sci-Fi",
    'Drama|Fantasy',
    'Action|Adventure|Horror|Thriller',
    'Comedy|Horror|Musical|Sci-Fi',
    'Comedy|Horror|Musical',
    'Action|Horror|Thriller',
    'Action|Drama|Fantasy|Romance',
    'Adventure|Fantasy|Sci-Fi',
    'Comedy|Drama|War',
    'Comedy|Drama|Western',
    'Adventure|Comedy|Sci-Fi',
    "Action|Children's|Fantasy",
    'Adventure|Fantasy',
    'Comedy|Western',
    'Crime|Drama|Sci-Fi',
    'Adventure|Sci-Fi',
    'Adventure|Drama',
    'Action|Adventure|Drama|Romance',
    'Action|Comedy|Musical|Sci-Fi',
    'Action|Adventure|Crime',
    'Action|Comedy|War',
    'Action|Comedy',
    'Comedy|Crime|Horror',
    "Action|Adventure|Children's|Sci-Fi",
    'Action|Adventure|Comedy',
    'Action|Adventure|Romance|Thriller',
    'Film-Noir|Thriller',
    'Action|Comedy|Sci-Fi|War',
    'Comedy|Crime|Mystery|Thriller',
    "Action|Children's",
    'Crime|Drama|Mystery|Thriller',
    'Action|Drama|Sci-Fi|Thriller',
    "Children's|Musical",
    "Adventure|Animation|Children's|Sci-Fi",
    'Adventure|Fantasy|Romance',
    'Action|Adventure|Horror',
    'Action|Comedy|Fantasy',
    'Animation|Musical',
    'Action|War',
    'Comedy|Crime|Thriller',
    'Action|Sci-Fi|Western',
    'Adventure|Animation|Film-Noir',
    'Adventure|Romance|Sci-Fi',
    'Adventure|Drama|Thriller',
    'Adventure|Western',
    'Action|Crime|Sci-Fi',
    'Sci-Fi',
    'Horror|Thriller',
    'Action|Adventure|Comedy|Horror',
    'Horror|Sci-Fi',
```

```
'Action|Mystery|Romance|Thriller',
'Horror|Mystery|Thriller',
'Crime|Horror|Mystery|Thriller',
'Mystery|Sci-Fi|Thriller',
'Comedy|Documentary',
'Action|Sci-Fi|Thriller|Western',
'Drama|Mystery|Thriller',
'Action|Romance|Sci-Fi',
'Action|Adventure|Animation',
'Adventure|Animation|Sci-Fi',
'Action|Comedy|Crime|Horror|Thriller',
'Crime|Drama|Romance|Thriller',
'Action|Adventure|Animation|Horror|Sci-Fi',
'Comedy|Fantasy|Romance|Sci-Fi',
'Comedy|Mystery|Romance|Thriller',
'Crime|Drama|Film-Noir',
'Crime|Film-Noir|Thriller',
'Crime',
'Film-Noir|Sci-Fi|Thriller',
'Comedy|Thriller',
'Action|Crime|Drama|Thriller',
'Mystery|Sci-Fi',
'Action|Adventure|Sci-Fi|Thriller|War',
'Crime|Film-Noir',
'Adventure|Thriller',
'Mystery|Romance|Thriller',
'Comedy|Crime|Drama',
'Adventure|Crime|Sci-Fi|Thriller',
'Action|Adventure|Mystery|Sci-Fi',
'Action|Adventure|Western',
'Action|Drama|Mystery',
"Adventure|Animation|Children's|Comedy|Fantasy",
'Drama|Musical|War',
'Comedy|Mystery',
'Adventure|Sci-Fi|Thriller',
"Children's|Comedy|Sci-Fi",
'Adventure|Romance',
'Drama|Mystery|Romance',
'Adventure|Drama|Romance',
'Comedy|Drama|Sci-Fi',
'Romance|Thriller',
'Film-Noir|Romance|Thriller',
'Crime|Drama|Film-Noir|Thriller',
'Drama|Fantasy|Romance|Thriller',
'Action|Drama|Mystery|Romance|Thriller',
'Action|Thriller|War',
"Animation|Children's|Fantasy|War",
'Documentary|Musical',
'Adventure|Comedy|Romance',
"Adventure|Children's|Comedy|Musical",
'Action|Mystery|Thriller',
"Children's|Horror",
'Adventure|Musical|Romance',
"Children's|Comedy|Mystery",
'Romance|War',
'Action|Comedy|Romance|Thriller',
'Musical|Romance|War',
"Animation|Children's|Comedy|Romance",
'Comedy|Mystery|Romance',
'Action|Drama|Western',
"Action|Animation|Children's|Sci-Fi|Thriller|War",
'Comedy|Drama|Musical',
'Adventure|Comedy|Musical',
'Action|Crime|Mystery|Thriller',
'Action|Adventure|Drama|Thriller',
'Action|Adventure|Comedy|War',
'Mystery',
'Drama|Western',
'Action|Adventure|Crime|Thriller',
'Action|Mystery|Sci-Fi|Thriller',
"Adventure|Children's|Comedy|Fantasy|Romance",
"Adventure|Children's|Romance",
```

```
 "Action|Adventure|Animation|Children's|Fantasy",
 "Action|Adventure|Children's",
 "Adventure|Animation|Children's",
 'Musical|War',
 'Action|Crime|Mystery',
 "Adventure|Animation|Children's|Fantasy",
 'Comedy|Horror|Thriller',
 'Film-Noir',
 'Crime|Film-Noir|Mystery',
 'Drama|Film-Noir|Thriller',
 'Drama|Film-Noir',
 'Action|Adventure|War',
 'Crime|Drama|Romance',
 'Documentary|War',
 'Sci-Fi|Thriller|War',
 'Action|Comedy|Crime',
 'Crime|Horror',
 'Drama|Romance|Sci-Fi',
 'Crime|Mystery',
 'Comedy|Drama|Thriller',
 'Crime|Horror|Thriller',
 'Horror|Mystery',
 'Documentary|Drama',
 'Drama|Horror|Thriller',
 'Comedy|Horror|Sci-Fi',
 "Action|Adventure|Children's|Fantasy",
 'Animation|Mystery',
 'Comedy|Romance|Sci-Fi',
 'Romance|Western',
 'Drama|Romance|Western',
 'Comedy|Film-Noir|Thriller',
 'Film-Noir|Horror',
 'Fantasy']
```

In [51]:

```
# Splitting of genres into different columns
```

In [ ]:

```
# Genre category with a one-hot encoding ( 1 and 0)
```

In [52]:

```
split_data = master_data[[
    'Gender',
    'Age',
    'Occupation',
    'Rating',
    'Genres'
]]
```

In [59]:

```
Genre = master_data['Genres']
Genre = Genre.str.get_dummies().add_prefix('Genres_')
genres_df = pd.concat(
       [split_data.drop(
        ['Genres'],
        axis=1
    ),
     Genre],
    axis=1,

)
genres_df.head()
```

Out[59]:

Gender Age Occupation Rating Genres_Action Genres_Adventure Genres_Animation Genres_Children's Genres_Comed

	Gender	Age	Occupation	Rating	Genres_Action	Genres_Adventure	Genres_Animation	Genres_Children's	Genres_Comed
0	F	1	10	5	0	0	1	1	
1	F	1	10	5	0	0	1	1	
2	F	1	10	5	0	0	0	0	
3	F	1	10	4	1	1	0	0	
4	F	1	10	5	0	0	0	0	

5 rows × 22 columns

In [60]:

```
genres_df = pd.get_dummies(
    genres_df,
    columns=['Gender']
)
```

In [61]:

```
genres_df.head()
```

Out[61]:

	Age	Occupation	Rating	Genres_Action	Genres_Adventure	Genres_Animation	Genres_Children's	Genres_Comedy	Genre
0	1	10	5	0	0	1	1	1	
1	1	10	5	0	0	1	1	0	
2	1	10	5	0	0	0	0	0	
3	1	10	4	1	1	0	0	0	
4	1	10	5	0	0	0	0	0	

5 rows × 23 columns

In [62]:

```
genres_df.columns
```

Out[62]:

```
Index(['Age', 'Occupation', 'Rating', 'Genres_Action', 'Genres_Adventure',
       'Genres_Animation', 'Genres_Children's', 'Genres_Comedy',
       'Genres_Crime', 'Genres_Documentary', 'Genres_Drama', 'Genres_Fantasy',
       'Genres_Film-Noir', 'Genres_Horror', 'Genres_Musical', 'Genres_Mystery',
       'Genres_Romance', 'Genres_Sci-Fi', 'Genres_Thriller', 'Genres_War',
       'Genres_Western', 'Gender_F', 'Gender_M'],
      dtype='object')
```

In [63]:

```
genres_df.describe()
```

Out[63]:

	Age	Occupation	Rating	Genres_Action	Genres_Adventure	Genres_Animation	Genres_Children's G
count	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06	1.000209e+06
mean	2.973831e+01	8.036138e+00	3.581564e+00	2.574032e-01	1.339250e-01	4.328395e-02	7.217092e-02
std	1.175198e+01	6.531336e+00	1.117102e+00	4.372036e-01	3.405719e-01	2.034957e-01	2.587708e-01
min	1.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2.500000e+01	2.000000e+00	3.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.500000e+01	7.000000e+00	4.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

	Age	Occupation	Rating	Genres_Action	Genres_Adventure	Genres_Animation	Genres_Children's	G
75%	3.500000e+01	1.400000e+01	4.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
max	5.600000e+01	2.000000e+01	5.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	

8 rows × 23 columns

In [ ]:

```
# Features affecting the ratings of any particular movie
```

In [65]:

```
genres_df.dtypes
```

Out[65]:

```
Age                      int64
Occupation               int64
Rating                   int64
Genres_Action            int64
Genres_Adventure         int64
Genres_Animation         int64
Genres_Children's        int64
Genres_Comedy            int64
Genres_Crime             int64
Genres_Documentary       int64
Genres_Drama             int64
Genres_Fantasy           int64
Genres_Film-Noir         int64
Genres_Horror            int64
Genres_Musical           int64
Genres_Mystery           int64
Genres_Romance           int64
Genres_Sci-Fi            int64
Genres_Thriller          int64
Genres_War               int64
Genres_Western           int64
Gender_F                 uint8
Gender_M                 uint8
dtype: object
```

In [71]:

```
# Linear Regression
# Model for prediction
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

from sklearn import metrics

lineReg = LinearRegression(
    copy_X=True,
    fit_intercept=True,
    n_jobs=1,
    normalize=False
)
```

In [72]:

```
sample_df = genres_df.sample(
    n=50000,
    random_state=0
)
sample_df.head()
```

Out[72]:

	Age	Occupation	Rating	Genres_Action	Genres_Adventure	Genres_Animation	Genres_Children's	Genres_Comedy

	Age	Occupation	Rating	Genres_Action	Genres_Adventure	Genres_Animation	Genres_Children's	Genres_Comedy
324271	18	4	4	0	0	0	0	1
818637	18	4	3	0	0	1	1	0
148677	18	14	5	0	0	0	0	0
778790	50	7	4	0	0	0	0	1
525489	25	2	5	0	0	0	0	0

5 rows × 23 columns

In [73]:

```
X = sample_df.drop('Rating', axis=1)
y = sample_df['Rating']
```

In [74]:

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.20,
    random_state=0
)
```

In [77]:

```
X.shape,y.shape
```

Out[77]:

```
((50000, 22), (50000,))
```

In [78]:

```
lin_reg = LinearRegression()
```

In [80]:

```
lin_reg.fit(X_train, y_train)
```

Out[80]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [83]:

```
y_pred = lin_reg.predict(X_test)
y_pred
```

Out[83]:

```
array([4.3223625 , 3.43954818, 3.40859324, ..., 3.78446548, 3.45475971,
       3.52162023])
```

In [84]:

```
print(
    'y-intercept: ',
    lin_reg.intercept_
)
print(
    'Beta coefficients: ',
    lin_reg.coef_
)
print(
    'Mean Abs Error  MAE: ',
    metrics.mean_absolute_error(y_test, y_pred)
)
print(
    'Mean Sq Error  MSE: ',
```

```
    metrics.mean_squared_error(y_test, y_pred)
)
print(
    'Root Mean Sq Error RMSE:',
    np.sqrt(metrics.mean_squared_error(y_test, y_pred))
)
print(
    'r2 value: ',
    metrics.r2_score(y_test, y_pred)
)
```

```
y-intercept:  3.3714137555159627
Beta coefficients:  [ 0.00406322  0.00098825 -0.0933231   0.00822898  0.41190314 -0.32536
968
 -0.00937548  0.07845926  0.43311855  0.22781148  0.07368389  0.3951835
 -0.29085584  0.12523149  0.02288591  0.00234758 -0.01347635  0.06128953
  0.30880281  0.14777492  0.01440465 -0.01440465]
Mean Abs ErrorMAE:    0.8978299534841195
Mean Sq Error  MSE:  1.1977731707567232
Root Mean Sq Error RMSE: 1.0944282391992282
r2 value:  0.03795269985311833
```

In [ ]:

```
Main features affecting the ratings for the movies- 1.Age and 2.Occupation
```

In [85]:

```
# Prediction of movie ratings
X_train.dtypes
```

Out[85]:

```
Age                   int64
Occupation            int64
Genres_Action         int64
Genres_Adventure      int64
Genres_Animation      int64
Genres_Children's     int64
Genres_Comedy         int64
Genres_Crime          int64
Genres_Documentary    int64
Genres_Drama          int64
Genres_Fantasy        int64
Genres_Film-Noir      int64
Genres_Horror         int64
Genres_Musical        int64
Genres_Mystery        int64
Genres_Romance        int64
Genres_Sci-Fi         int64
Genres_Thriller       int64
Genres_War            int64
Genres_Western        int64
Gender_F              uint8
Gender_M              uint8
dtype: object
```

In [87]:

```
prediction_df = pd.DataFrame({'Test': y_test, 'Prediction': y_pred})
prediction_df.head()
```

Out[87]:

	Test	Prediction
187446	4	4.322363
69421	4	3.439548
941725	3	3.408593
841836	4	3.652663

869012	Test 4	Prediction 3.559436

In [ ]:

```

```