In [472…
```
#importing pandas
import pandas as pd

#importing numpy
import numpy as np

#importing warning to ignore unwanted warning
import warnings
warnings.filterwarnings('ignore')

#importing seaborn
import seaborn as sns

#importing pandas profiling for generating a basic report about the dataframe
import pandas_profiling as pf

#importing matplolib for ploting graphs
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

#importing regular-expression
import re

#importing train test split from sklearn for predictive modeling
from sklearn.model_selection import train_test_split

#importing accuracy score from sklearn to calculates the accuracy score for a set of predicted labels against t
from sklearn.metrics import accuracy_score

#importing Labelencoder from sklearn that allows us to assign ordinal levels to categorical data
from sklearn.preprocessing import LabelEncoder

#importing LGBMClassifier
from lightgbm import LGBMClassifier

#importing xgboost
import xgboost
```

Importing the datasets : Rating, User, Movie

In [473…
```
rating = ['UserID','MovieID','Rating','Timestamp']
user = ['UserID','Gender','Age','Occupation','Zip-code']
movie = ['MovieID','Title','Genres']
```

In [474…
```
rating_df = pd.read_csv('ratings.dat',header=None,delimiter='::',names=rating)
print(rating_df.head())
print()
print(rating_df.shape)
```

```
   UserID  MovieID  Rating  Timestamp
0       1     1193       5  978300760
1       1      661       3  978302109
2       1      914       3  978301968
3       1     3408       4  978300275
4       1     2355       5  978824291

(1000209, 4)
```

In [475…
```
user_df = pd.read_csv('users.dat',header=None,delimiter='::',names=user)
print(user_df.head())
print()
print(user_df.shape)
```

```
   UserID Gender  Age  Occupation Zip-code
0       1      F    1          10    48067
1       2      M   56          16    70072
2       3      M   25          15    55117
3       4      M   45           7    02460
4       5      M   25          20    55455

(6040, 5)
```

In [476…
```
movie_df = pd.read_csv('movies.dat',header=None,delimiter='::',names=movie, encoding='latin-1')
print(movie_df.head())
print()
print(movie_df.shape)
```

```
   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

(3883, 3)
```

In [477…
```
df = rating_df.merge(user_df,how='outer',on='UserID')
df = df.merge(movie_df,how='outer',on='MovieID')
df.head()
```

Out[477]:

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code	Title	Genres
0	1.0	1193	5.0	978300760.0	F	1.0	10.0	48067	One Flew Over the Cuckoo's Nest (1975)	Drama
1	2.0	1193	5.0	978298413.0	M	56.0	16.0	70072	One Flew Over the Cuckoo's Nest (1975)	Drama
2	12.0	1193	4.0	978220179.0	M	25.0	12.0	32793	One Flew Over the Cuckoo's Nest (1975)	Drama
3	15.0	1193	4.0	978199279.0	M	25.0	7.0	22903	One Flew Over the Cuckoo's Nest (1975)	Drama
4	17.0	1193	5.0	978158471.0	M	50.0	1.0	95350	One Flew Over the Cuckoo's Nest (1975)	Drama

In [478…
```
df.info()
```

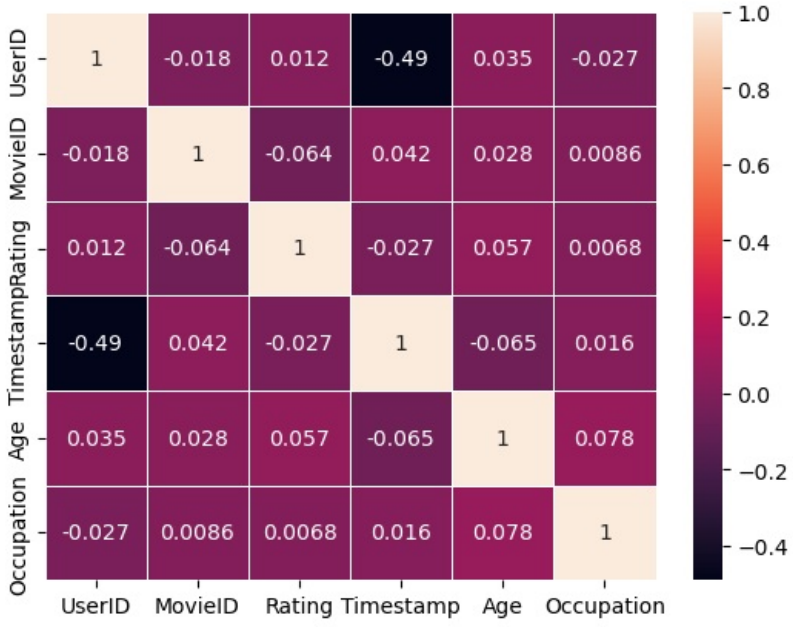
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000386 entries, 0 to 1000385
Data columns (total 10 columns):
 #   Column      Non-Null Count    Dtype  
---  ------      --------------    -----  
 0   UserID      1000209 non-null  float64
 1   MovieID     1000386 non-null  int64  
 2   Rating      1000209 non-null  float64
 3   Timestamp   1000209 non-null  float64
 4   Gender      1000209 non-null  object 
 5   Age         1000209 non-null  float64
 6   Occupation  1000209 non-null  float64
 7   Zip-code    1000209 non-null  object 
 8   Title       1000386 non-null  object 
 9   Genres      1000386 non-null  object 
dtypes: float64(5), int64(1), object(4)
memory usage: 84.0+ MB
```

In [479…
```
df.shape
```

Out[479]: (1000386, 10)

In [480…
```
corr = df.corr()
sns.heatmap(corr,annot= True,linewidths=0.5)
```

Out[480]: <AxesSubplot:>



Extracting the pandas profiling report

In [481…
```
df.describe()
pfr = pf.ProfileReport(df)
pfr.to_file('Movielens_pfr.html')
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
```

```
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

In [482…
```
print('Na values in the data frame is :')
def is_na(x):
    for i in x.columns:
        print(i,'column',' :',x[i].isna().sum(),'\n')
is_na(df)
```

```
Na values in the data frame is :
UserID column  : 177

MovieID column  : 0

Rating column  : 177

Timestamp column  : 177

Gender column  : 177

Age column  : 177

Occupation column  : 177

Zip-code column  : 177

Title column  : 0

Genres column  : 0
```

In [483…
```
df.dropna(inplace=True)
```

In [484…
```
df.Rating.isna().value_counts()
```

Out[484]:
```
False    1000209
Name: Rating, dtype: int64
```

In [485…
```
def df_unique(X):
    for i in X.columns:
        print('Column : ',i,'\n',X[i].unique(), '\n Total unique values is: ', X[i].nunique())
        print('-------------------------------------------------------------------------')
df_unique(df)
```

```
Column :  UserID
 [1.000e+00 2.000e+00 1.200e+01 ... 2.982e+03 3.893e+03 4.211e+03]
 Total unique values is:  6040
-------------------------------------------------------------------------
Column :  MovieID
 [1193  661  914 ... 2845 3607 2909]
 Total unique values is:  3706
-------------------------------------------------------------------------
Column :  Rating
 [5. 4. 3. 2. 1.]
 Total unique values is:  5
-------------------------------------------------------------------------
Column :  Timestamp
 [9.78300760e+08 9.78298413e+08 9.78220179e+08 ... 9.58846401e+08
 9.76029116e+08 9.57273353e+08]
 Total unique values is:  458455
-------------------------------------------------------------------------
Column :  Gender
 ['F' 'M']
 Total unique values is:  2
-------------------------------------------------------------------------
Column :  Age
 [ 1. 56. 25. 50. 18. 45. 35.]
 Total unique values is:  7
-------------------------------------------------------------------------
Column :  Occupation
 [10. 16. 12.  7.  1.  3.  4.  8. 17.  0.  2.  9. 19. 18. 15. 11. 20. 13.
  5. 14.  6.]
 Total unique values is:  21
-------------------------------------------------------------------------
Column :  Zip-code
 ['48067' '70072' '32793' ... '74403' '79401' '77662']
 Total unique values is:  3439
-------------------------------------------------------------------------
Column :  Title
 ["One Flew Over the Cuckoo's Nest (1975)"
 'James and the Giant Peach (1996)' 'My Fair Lady (1964)' ...
 'White Boys (1999)' 'One Little Indian (1973)'
 'Five Wives, Three Secretaries and Me (1998)']
 Total unique values is:  3706
-------------------------------------------------------------------------
Column :  Genres
 ['Drama' "Animation|Children's|Musical" 'Musical|Romance'
 "Animation|Children's|Comedy" 'Action|Adventure|Comedy|Romance'
```

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

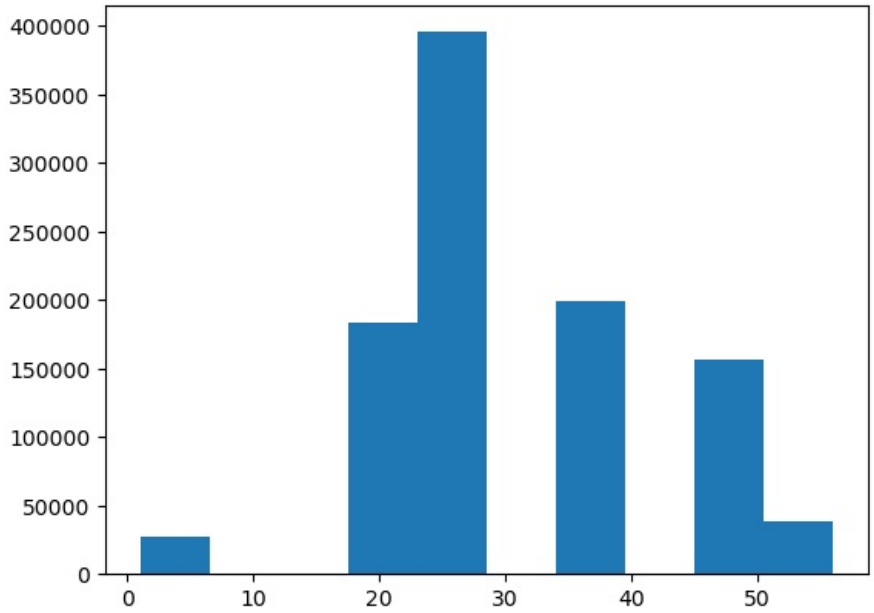
```
 'Action|Adventure|Drama|Romance' "Children's|Musical"
 'Action|Comedy|Musical|Sci-Fi' 'Action|Drama|Sci-Fi|Thriller'
 'Action|Comedy|Fantasy' 'Action|War' 'Action|Comedy|Sci-Fi|War'
 'Comedy|Crime|Horror' 'Action|Comedy|War'
 "Action|Adventure|Children's|Sci-Fi" "Action|Children's"
 'Comedy|Documentary' 'Action|Adventure|Animation'
 'Action|Mystery|Thriller'
 "Action|Animation|Children's|Sci-Fi|Thriller|War" 'Crime|Drama|Romance'
 'Crime|Film-Noir' 'Mystery|Romance|Thriller'
 'Comedy|Mystery|Romance|Thriller' 'Action|Adventure|Sci-Fi|Thriller|War'
 'Adventure|Crime|Sci-Fi|Thriller' 'Action|Adventure|Western'
 "Animation|Children's|Fantasy|War" 'Action|Adventure|Comedy|War'
 "Children's|Comedy|Sci-Fi"
 "Adventure|Animation|Children's|Comedy|Fantasy" 'Drama|Musical|War'
 'Drama|Mystery|Romance' 'Adventure|Drama|Romance' 'Film-Noir'
 'Film-Noir|Romance|Thriller' 'Drama|Film-Noir' 'Romance|Thriller'
 'Action|Adventure|War' 'Mystery' 'Action|Adventure|Drama|Thriller'
 'Musical|Romance|War' 'Drama|Western'
 'Action|Drama|Mystery|Romance|Thriller' 'Adventure|Comedy|Musical'
 'Documentary|Musical' 'Action|Thriller|War' 'Adventure|Comedy|Romance'
 "Adventure|Children's|Comedy|Fantasy|Romance" 'Romance|War'
 'Comedy|Romance|Sci-Fi' 'Action|Mystery|Sci-Fi|Thriller'
 "Children's|Horror" 'Adventure|Musical|Romance'
 "Adventure|Children's|Comedy|Musical" "Children's|Comedy|Mystery"
 'Action|Comedy|Romance|Thriller' 'Action|Drama|Western'
 "Animation|Children's|Comedy|Romance" 'Comedy|Mystery|Romance'
 'Action|Crime|Mystery' 'Comedy|Drama|Thriller' 'Musical|War'
 'Documentary|Drama' 'Action|Adventure|Crime|Thriller'
 "Action|Adventure|Children's" "Adventure|Children's|Romance"
 "Adventure|Animation|Children's"
 "Action|Adventure|Animation|Children's|Fantasy"
 "Adventure|Animation|Children's|Fantasy" 'Drama|Film-Noir|Thriller'
 'Crime|Mystery' 'Documentary|War' 'Action|Comedy|Crime'
 'Drama|Romance|Sci-Fi' 'Horror|Mystery' 'Drama|Horror|Thriller'
 "Action|Adventure|Children's|Fantasy" 'Animation|Mystery'
 'Drama|Romance|Western' 'Romance|Western' 'Comedy|Film-Noir|Thriller'
 'Fantasy' 'Film-Noir|Horror']
Total unique values is:  301
```

Exploring the datasets using visual representations

Visualizing the User Age Distribution

In [486…
```
df.Age.hist(grid=False)
```

Out[486]:
```
<AxesSubplot:>
```



Visualizing User rating of the movie “Toy Story”

In [487…
```
def fn(x):
    return re.search("Toy Story".lower(), x.lower())!=None
title = df.iloc[0].Title
title
```

Out[487]:
```
"One Flew Over the Cuckoo's Nest (1975)"
```

In [488…
```
re_tit = df["Title"].apply(fn)
```

```
re_tit.head()
```

Out[488]:
```
0    False
1    False
2    False
3    False
4    False
Name: Title, dtype: bool
```

In [489…
```
toystory = df[df["Title"].apply(fn)]
toystory
```

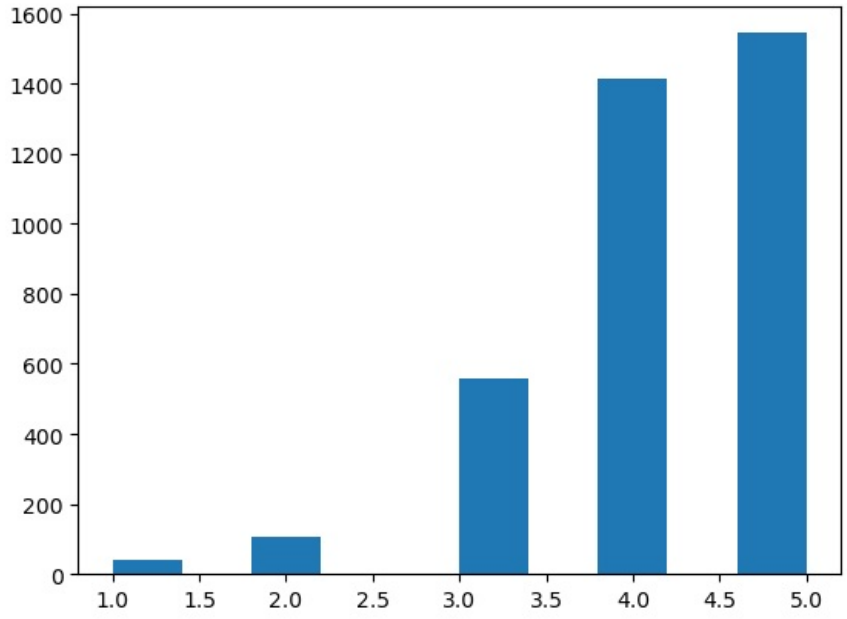
Out[489]:

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code	Title	Genres
41626	1.0	1	5.0	978824268.0	F	1.0	10.0	48067	Toy Story (1995)	Animation|Children's|Comedy
41627	6.0	1	4.0	978237008.0	F	50.0	9.0	55117	Toy Story (1995)	Animation|Children's|Comedy
41628	8.0	1	4.0	978233496.0	M	25.0	12.0	11413	Toy Story (1995)	Animation|Children's|Comedy
41629	9.0	1	5.0	978225952.0	M	25.0	17.0	61614	Toy Story (1995)	Animation|Children's|Comedy
41630	10.0	1	5.0	978226474.0	F	35.0	1.0	95370	Toy Story (1995)	Animation|Children's|Comedy
...	...	...	...	...	...	...	...	...	...	...
56826	6022.0	3114	5.0	956755741.0	M	25.0	17.0	57006	Toy Story 2 (1999)	Animation|Children's|Comedy
56827	6024.0	3114	4.0	956749447.0	M	25.0	12.0	53705	Toy Story 2 (1999)	Animation|Children's|Comedy
56828	6027.0	3114	4.0	956726766.0	M	18.0	4.0	20742	Toy Story 2 (1999)	Animation|Children's|Comedy
56829	6036.0	3114	4.0	956710231.0	F	25.0	15.0	32603	Toy Story 2 (1999)	Animation|Children's|Comedy
56830	6037.0	3114	4.0	956719174.0	F	45.0	1.0	76006	Toy Story 2 (1999)	Animation|Children's|Comedy

3662 rows × 10 columns

In [490…
```
toystory.Rating.hist(grid=False)
```

Out[490]:
```
<AxesSubplot:>
```



Top 25 movies by viewership rating

In [491…
```
top_25 = df.groupby(["MovieID", "Title"]).Timestamp.count().sort_values(ascending=False)
top_25
```

Out[491]:
```
MovieID  Title
2858     American Beauty (1999)                                  3428
260      Star Wars: Episode IV - A New Hope (1977)               2991
1196     Star Wars: Episode V - The Empire Strikes Back (1980)   2990
1210     Star Wars: Episode VI - Return of the Jedi (1983)       2883
480      Jurassic Park (1993)                                    2672
                                                                 ...
3237     Kestrel's Eye (Falkens öga) (1998)                         1
763      Last of the High Kings, The (a.k.a. Summer Fling) (1996)   1
624      Condition Red (1995)                                       1
2563     Beauty (1998)                                              1
3290     Soft Toilet Seats (1999)                                   1
Name: Timestamp, Length: 3706, dtype: int64
```

In [492…
```
print('Top 25 movies by viewership rating')
print(top_25[:25])
```

```
Top 25 movies by viewership rating
MovieID  Title
2858     American Beauty (1999)                                3428
260      Star Wars: Episode IV - A New Hope (1977)             2991
1196     Star Wars: Episode V - The Empire Strikes Back (1980) 2990
1210     Star Wars: Episode VI - Return of the Jedi (1983)     2883
480      Jurassic Park (1993)                                  2672
2028     Saving Private Ryan (1998)                            2653
589      Terminator 2: Judgment Day (1991)                     2649
2571     Matrix, The (1999)                                    2590
1270     Back to the Future (1985)                             2583
593      Silence of the Lambs, The (1991)                      2578
1580     Men in Black (1997)                                   2538
1198     Raiders of the Lost Ark (1981)                        2514
608      Fargo (1996)                                          2513
2762     Sixth Sense, The (1999)                               2459
110      Braveheart (1995)                                     2443
2396     Shakespeare in Love (1998)                            2369
1197     Princess Bride, The (1987)                            2318
527      Schindler's List (1993)                               2304
1617     L.A. Confidential (1997)                              2288
1265     Groundhog Day (1993)                                  2278
1097     E.T. the Extra-Terrestrial (1982)                     2269
2628     Star Wars: Episode I - The Phantom Menace (1999)      2250
2997     Being John Malkovich (1999)                           2241
318      Shawshank Redemption, The (1994)                      2227
858      Godfather, The (1972)                                 2223
Name: Timestamp, dtype: int64
```

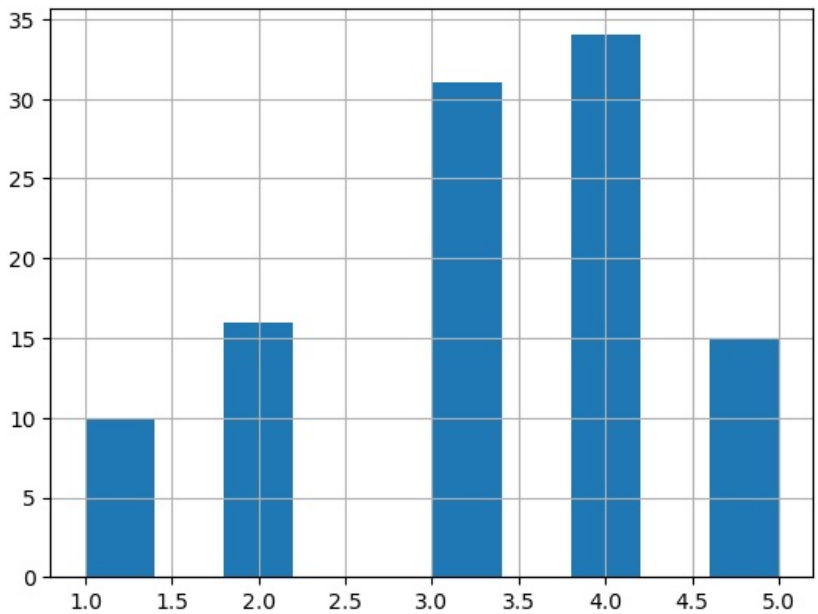
The ratings for all the movies reviewed by for a particular user of user id = 2696

In [540…
```
usr_2696 = df.loc[df.UserID==2696, "Rating"].sort_values(ascending=False)
usr_2696.head(),usr_2696.shape
```

Out[540]:
```
(811       5.0
 420296    5.0
 127592    5.0
 120959    5.0
 6987      5.0
 Name: Rating, dtype: float64,
 (106,))
```

In [542…
```
usr_3000.hist()
```

Out[542]:
```
<AxesSubplot:>
```



Finding all the unique genres

In [495…
```
df.Genres.unique()
```

Out[495]:
```
array(['Drama', "Animation|Children's|Musical", 'Musical|Romance',
       "Animation|Children's|Comedy", 'Action|Adventure|Comedy|Romance',
       'Action|Adventure|Drama', 'Comedy|Drama',
       "Adventure|Children's|Drama|Musical", 'Musical', 'Comedy',
       "Animation|Children's", 'Comedy|Fantasy', 'Animation',
       'Comedy|Sci-Fi', 'Drama|War', 'Romance',
       "Animation|Children's|Musical|Romance",
       "Children's|Drama|Fantasy|Sci-Fi", 'Drama|Romance',
       'Animation|Comedy|Thriller',
       "Adventure|Animation|Children's|Comedy|Musical",
```

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

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

In [496…
```
Genres_list = df.Genres.tolist()
genre_list = []
i = 0
while(i<len(Genres_list)):
    genre_list+= Genres_list[i].split('|')
    i+=1
```

In [497…
```
unique_gen = list(set(genre_list))
print(unique_gen)
print()
print("Length of the unique Genre : ",len(unique_gen))
```

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

Length of the unique Genre :  18
```

Creating a separate column for each genre category with a one-hot encoding (1 and 0)

In [498…
```
new_data = pd.concat([df,df.Genres.str.get_dummies()], axis=1)
print(new_data.columns)
```

```
Index(['UserID', 'MovieID', 'Rating', 'Timestamp', 'Gender', 'Age',
       'Occupation', 'Zip-code', 'Title', 'Genres', 'Action', 'Adventure',
       'Animation', 'Children's', 'Comedy', 'Crime', 'Documentary', 'Drama',
       'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance',
       'Sci-Fi', 'Thriller', 'War', 'Western'],
      dtype='object')
```

In [499…
```
new_data.head()
```

Out[499]:

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code	Title	Genres	...	Fantasy	Film-Noir	Horror	Musical	Mystery
0	1.0	1193	5.0	978300760.0	F	1.0	10.0	48067	One Flew Over the Cuckoo's Nest (1975)	Drama	...	0	0	0	0	0
1	2.0	1193	5.0	978298413.0	M	56.0	16.0	70072	One Flew Over the Cuckoo's Nest (1975)	Drama	...	0	0	0	0	0
2	12.0	1193	4.0	978220179.0	M	25.0	12.0	32793	One Flew Over the Cuckoo's Nest (1975)	Drama	...	0	0	0	0	0
3	15.0	1193	4.0	978199279.0	M	25.0	7.0	22903	One Flew Over the Cuckoo's Nest (1975)	Drama	...	0	0	0	0	0
4	17.0	1193	5.0	978158471.0	M	50.0	1.0	95350	One Flew Over the Cuckoo's Nest (1975)	Drama	...	0	0	0	0	0

5 rows × 28 columns

In [500…
```
df_new = new_data.drop(['Title','Zip-code','Timestamp','Genres'],axis=1)
df_new.head()
```

Out[500]:

	UserID	MovieID	Rating	Gender	Age	Occupation	Action	Adventure	Animation	Children's	...	Fantasy	Film-Noir	Horror	Musical	Myste
0	1.0	1193	5.0	F	1.0	10.0	0	0	0	0	...	0	0	0	0	
1	2.0	1193	5.0	M	56.0	16.0	0	0	0	0	...	0	0	0	0	
2	12.0	1193	4.0	M	25.0	12.0	0	0	0	0	...	0	0	0	0	
3	15.0	1193	4.0	M	25.0	7.0	0	0	0	0	...	0	0	0	0	
4	17.0	1193	5.0	M	50.0	1.0	0	0	0	0	...	0	0	0	0	

5 rows × 24 columns

In [501…
```
print(df_new.columns)
```

```
Index(['UserID', 'MovieID', 'Rating', 'Gender', 'Age', 'Occupation', 'Action',
       'Adventure', 'Animation', 'Children's', 'Comedy', 'Crime',
       'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical',
       'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western'],
      dtype='object')
```

Encoding the gender column

In [517…
```
df_new.Gender = pd.get_dummies(df_new.Gender, drop_first=True)
```

In [502…
```
df_new['Gender'] = df['Gender'].replace(['M','F'],[0,1])
print (df_new)
```

```
         UserID  MovieID  Rating  Gender   Age  Occupation  Action  Adventure  \
0           1.0     1193     5.0       1   1.0        10.0       0          0
1           2.0     1193     5.0       0  56.0        16.0       0          0
2          12.0     1193     4.0       0  25.0        12.0       0          0
3          15.0     1193     4.0       0  25.0         7.0       0          0
4          17.0     1193     5.0       0  50.0         1.0       0          0
...         ...      ...     ...     ...   ...         ...     ...        ...
1000204  5949.0     2198     5.0       0  18.0        17.0       0          0
1000205  5675.0     2703     3.0       0  35.0        14.0       0          0
1000206  5780.0     2845     1.0       0  18.0        17.0       0          0
1000207  5851.0     3607     5.0       1  18.0        20.0       0          0
1000208  5938.0     2909     4.0       0  25.0         1.0       0          0

         Animation  Children's  ...  Fantasy  Film-Noir  Horror  Musical  \
0                0           0  ...        0          0       0        0
1                0           0  ...        0          0       0        0
2                0           0  ...        0          0       0        0
3                0           0  ...        0          0       0        0
4                0           0  ...        0          0       0        0
...            ...         ...  ...      ...        ...     ...      ...
1000204          0           0  ...        0          0       0        0
1000205          0           0  ...        0          0       0        0
1000206          0           0  ...        0          0       0        0
1000207          0           0  ...        0          0       0        0
1000208          0           0  ...        0          0       0        0

         Mystery  Romance  Sci-Fi  Thriller  War  Western
0              0        0       0         0    0        0
1              0        0       0         0    0        0
2              0        0       0         0    0        0
3              0        0       0         0    0        0
4              0        0       0         0    0        0
...          ...      ...     ...       ...  ...      ...
1000204        0        0       0         0    0        0
1000205        0        0       0         0    0        0
1000206        0        0       0         0    0        0
1000207        0        0       0         0    0        1
1000208        0        0       0         0    0        0

[1000209 rows x 24 columns]
```

In [518…
```
x = data.drop(['UserID','MovieID','Rating'],axis=1)
x.shape
```

Out[518]:
```
(1000209, 21)
```

The features affecting the ratings of any particular movie.

In [519…
```
print('The features affecting the ratings of any particular movie:')
print()
print(x.columns)
```

```
The features affecting the ratings of any particular movie:

Index(['Age', 'Occupation', 'Action', 'Adventure', 'Animation', 'Children's',
       'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir',
       'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War',
       'Western', 'Gender'],
      dtype='object')
```

In [520…
```
y = data.Rating
y.shape
```

Out[520]:
```
(1000209,)
```

In [521…
```
x.Occupation.value_counts()
```

Out[521]:
```
4.0     131032
0.0     130499
7.0     105425
1.0      85351
17.0     72816
20.0     60397
12.0     57214
2.0      50068
14.0     49109
16.0     46021
6.0      37205
3.0      31623
10.0     23290
15.0     22951
5.0      21850
11.0     20563
19.0     14904
13.0     13754
18.0     12086
9.0      11345
8.0       2706
Name: Occupation, dtype: int64
```

In [522…
```
x = x.join(pd.get_dummies(x.Occupation,prefix='Occupation'))
x.head(),x.columns
```

Out[522]:
```
(    Age  Occupation  Action  Adventure  Animation  Children's  Comedy  Crime  \
 0   1.0        10.0       0          0          0           0       0      0
 1  56.0        16.0       0          0          0           0       0      0
 2  25.0        12.0       0          0          0           0       0      0
 3  25.0         7.0       0          0          0           0       0      0
 4  50.0         1.0       0          0          0           0       0      0

    Documentary  Drama  ...  Occupation_11.0  Occupation_12.0  Occupation_13.0  \
 0            0      1  ...                0                0                0
 1            0      1  ...                0                0                0
 2            0      1  ...                0                1                0
 3            0      1  ...                0                0                0
 4            0      1  ...                0                0                0

    Occupation_14.0  Occupation_15.0  Occupation_16.0  Occupation_17.0  \
 0                0                0                0                0
 1                0                0                1                0
 2                0                0                0                0
 3                0                0                0                0
 4                0                0                0                0

    Occupation_18.0  Occupation_19.0  Occupation_20.0
 0                0                0                0
 1                0                0                0
 2                0                0                0
 3                0                0                0
 4                0                0                0

 [5 rows x 42 columns],
 Index(['Age', 'Occupation', 'Action', 'Adventure', 'Animation', 'Children's',
        'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir',
        'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War',
        'Western', 'Gender', 'Occupation_0.0', 'Occupation_1.0',
        'Occupation_2.0', 'Occupation_3.0', 'Occupation_4.0', 'Occupation_5.0',
        'Occupation_6.0', 'Occupation_7.0', 'Occupation_8.0', 'Occupation_9.0',
        'Occupation_10.0', 'Occupation_11.0', 'Occupation_12.0',
        'Occupation_13.0', 'Occupation_14.0', 'Occupation_15.0',
        'Occupation_16.0', 'Occupation_17.0', 'Occupation_18.0',
        'Occupation_19.0', 'Occupation_20.0'],
       dtype='object'))
```

In [523…
```
x = x.drop(['Occupation','Occupation_0.0'],axis=1)
x.head(3),x.shape
```

Out[523]:
```
(    Age  Action  Adventure  Animation  Children's  Comedy  Crime  Documentary  \
 0   1.0       0          0          0           0       0      0            0
 1  56.0       0          0          0           0       0      0            0
 2  25.0       0          0          0           0       0      0            0

    Drama  Fantasy  ...  Occupation_11.0  Occupation_12.0  Occupation_13.0  \
 0      1        0  ...                0                0                0
 1      1        0  ...                0                0                0
 2      1        0  ...                0                1                0

    Occupation_14.0  Occupation_15.0  Occupation_16.0  Occupation_17.0  \
 0                0                0                0                0
 1                0                0                1                0
 2                0                0                0                0

    Occupation_18.0  Occupation_19.0  Occupation_20.0
 0                0                0                0
 1                0                0                0
 2                0                0                0

 [3 rows x 40 columns],
 (1000209, 40))
```

Deploying the hold out method

In [524…
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_state = 10,stratify=y)
```

Deploying the model

In [525…
```
lgb = LGBMClassifier(boosting_type = 'gbdt',n_jobs= -1,objective='multiclass')
```

In [526…
```
lgb.fit(x_train,y_train)
```

Out[526]:
```
LGBMClassifier(objective='multiclass')
```

In [527…
```
y_pred = lgb.predict(x_test)
```

In [528…
```
print('LGBM accuracy score is : ', accuracy_score(y_test,y_pred)*100)
```

```
LGBM accuracy score is :  36.32887093710321
```

In [535…
```
xgb = xgboost.XGBClassifier(n_jobs = 1)
```

In [536…
```
xgb.fit(x_train,y_train)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20260\248593607.py in <module>
----> 1 xgb.fit(x_train,y_train)

~\anaconda3\lib\site-packages\xgboost\core.py in inner_f(*args, **kwargs)
    618             for k, arg in zip(sig.parameters, args):
    619                 kwargs[k] = arg
--> 620             return func(**kwargs)
    621
    622         return inner_f

~\anaconda3\lib\site-packages\xgboost\sklearn.py in fit(self, X, y, sample_weight, base_margin, eval_set, eval_
metric, early_stopping_rounds, verbose, xgb_model, sample_weight_eval_set, base_margin_eval_set, feature_weight
s, callbacks)
   1438                 or not (self.classes_ == expected_classes).all()
   1439             ):
-> 1440                 raise ValueError(
   1441                     f"Invalid classes inferred from unique values of `y`.  "
   1442                     f"Expected: {expected_classes}, got {self.classes_}"

ValueError: Invalid classes inferred from unique values of `y`.  Expected: [0 1 2 3 4], got [1. 2. 3. 4. 5.]
```

In [531…
```
y_pred_xgb = xgb.predict(x_test)
```

```
---------------------------------------------------------------------------
NotFittedError                            Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_20260\3480273728.py in <module>
----> 1 y_pred_xgb = xgb.predict(x_test)

~\anaconda3\lib\site-packages\xgboost\sklearn.py in predict(self, X, output_margin, ntree_limit, validate_featu
res, base_margin, iteration_range)
   1523     ) -> np.ndarray:
   1524         with config_context(verbosity=self.verbosity):
-> 1525             class_probs = super().predict(
   1526                 X=X,
   1527                 output_margin=output_margin,

~\anaconda3\lib\site-packages\xgboost\sklearn.py in predict(self, X, output_margin, ntree_limit, validate_featu
res, base_margin, iteration_range)
   1107         with config_context(verbosity=self.verbosity):
   1108             iteration_range = _convert_ntree_limit(
-> 1109                 self.get_booster(), ntree_limit, iteration_range
   1110             )
   1111             iteration_range = self._get_iteration_range(iteration_range)

~\anaconda3\lib\site-packages\xgboost\sklearn.py in get_booster(self)
    647             from sklearn.exceptions import NotFittedError
    648
--> 649             raise NotFittedError("need to call fit or load_model beforehand")
    650         return self._Booster
    651

NotFittedError: need to call fit or load_model beforehand
```

In [ ]:
```
print('XGB accuracy score is : ', accuracy_score(y_test,y_pred_xgb )*100)
```

Accuracy score check : LGBM & XGB models

LGBM accuracy score is : 36.32%

XGB accuracy score is : Continuous Error

In [ ]:

In [ ]: