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

```
df_user=pd.read_csv('users.dat',sep="::",names=['UserID','Gender','Age
','Occupation','Zip Code'],engine='python')
```

```
df_user
```

```
      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
...      ...    ...  ...         ...      ...
6035    6036      F   25          15    32603
6036    6037      F   45           1    76006
6037    6038      F   56           1    14706
6038    6039      F   45           0    01060
6039    6040      M   25           6    11106

[6040 rows x 5 columns]
```

```
df_movies=pd.read_csv('movies.dat',sep="::",names=['MovieID','Title','
Generes'],engine='python',encoding='latin-1')
```

```
df_movies
```

```
      MovieID                               Title  \
0           1                    Toy Story (1995)   
1           2                      Jumanji (1995)   
2           3             Grumpier Old Men (1995)   
3           4            Waiting to Exhale (1995)   
4           5  Father of the Bride Part II (1995)   
...       ...                                 ...   
3878     3948             Meet the Parents (2000)   
3879     3949          Requiem for a Dream (2000)   
3880     3950                    Tigerland (2000)   
3881     3951             Two Family House (2000)   
3882     3952               Contender, The (2000)   

                           Generes  
0      Animation|Children's|Comedy  
1     Adventure|Children's|Fantasy  
2                   Comedy|Romance  
3                     Comedy|Drama  
4                           Comedy  
...                            ...  
3878                        Comedy  
```

```
3879                         Drama
3880                         Drama
3881                         Drama
3882                Drama|Thriller

[3883 rows x 3 columns]
```

```
df_ratings=pd.read_csv('ratings.dat',sep="::",names=['UserID','MovieID','Rating','Timestamp'],engine='python')
```

```
df_ratings
```

```
         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
...         ...      ...     ...        ...
1000204    6040     1091       1  956716541
1000205    6040     1094       5  956704887
1000206    6040      562       5  956704746
1000207    6040     1096       4  956715648
1000208    6040     1097       4  956715569

[1000209 rows x 4 columns]
```

```
df_ratings.shape
```

```
(1000209, 4)
```

```
df_movies.shape
```

```
(3883, 3)
```

```
df_user.shape
```

```
(6040, 5)
```

Create a new dataset [Master_Data] with the following columns MovieID Title UserID Age Gender Occupation Rating. (Hint: (i) Merge two tables at a time. (ii) Merge the tables using two primary keys MovieID & UserId)

```
dfMovieRatings=df_movies.merge(df_ratings,on='MovieID',how='inner')
```

```
dfMovieRatings.shape
```

```
(1000209, 6)
```

```
dfMaster=dfMovieRatings.merge(df_user,on='UserID',how='inner')
```

```
dfMaster
```

```
         MovieID                                      Title  \
0              1                           Toy Story (1995)
1             48                          Pocahontas (1995)
2            150                           Apollo 13 (1995)
3            260  Star Wars: Episode IV - A New Hope (1977)
4            527                    Schindler's List (1993)
...          ...                                        ...
1000204     3513                 Rules of Engagement (2000)
1000205     3535                     American Psycho (2000)
1000206     3536                   Keeping the Faith (2000)
1000207     3555                               U-571 (2000)
1000208     3578                           Gladiator (2000)

                                      Generes  UserID  Rating
Timestamp  \
0                 Animation|Children's|Comedy       1       5
978824268
1        Animation|Children's|Musical|Romance       1       5
978824351
2                                       Drama       1       5
978301777
3             Action|Adventure|Fantasy|Sci-Fi       1       4
978300760
4                                   Drama|War       1       5
978824195
...                                       ...     ...     ...        .
..
1000204                        Drama|Thriller    5727       4
958489970
1000205                Comedy|Horror|Thriller    5727       2
958489970
1000206                        Comedy|Romance    5727       5
958489902
1000207                       Action|Thriller    5727       3
958490699
1000208                          Action|Drama    5727       5
958490171

        Gender  Age  Occupation Zip Code
0            F    1          10    48067
1            F    1          10    48067
2            F    1          10    48067
3            F    1          10    48067
4            F    1          10    48067
...        ...  ...         ...      ...
1000204      M   25           4    92843
1000205      M   25           4    92843
1000206      M   25           4    92843
1000207      M   25           4    92843
1000208      M   25           4    92843
```

```
[1000209 rows x 10 columns]
```

```
dfMaster.head(10)
```

```
   MovieID                                      Title  \
0        1                           Toy Story (1995)   
1       48                          Pocahontas (1995)   
2      150                           Apollo 13 (1995)   
3      260  Star Wars: Episode IV - A New Hope (1977)   
4      527                    Schindler's List (1993)   
5      531                  Secret Garden, The (1993)   
6      588                             Aladdin (1992)   
7      594     Snow White and the Seven Dwarfs (1937)   
8      595                Beauty and the Beast (1991)   
9      608                               Fargo (1996)   

                                Generes  UserID  Rating  Timestamp
Gender  \
0           Animation|Children's|Comedy       1       5  978824268
F   
1  Animation|Children's|Musical|Romance       1       5  978824351
F   
2                                 Drama       1       5  978301777
F   
3       Action|Adventure|Fantasy|Sci-Fi       1       4  978300760
F   
4                             Drama|War       1       5  978824195
F   
5                      Children's|Drama       1       4  978302149
F   
6   Animation|Children's|Comedy|Musical       1       4  978824268
F   
7          Animation|Children's|Musical       1       4  978302268
F   
8          Animation|Children's|Musical       1       5  978824268
F   
9                  Crime|Drama|Thriller       1       4  978301398
F   

   Age  Occupation Zip Code
0    1          10    48067
1    1          10    48067
2    1          10    48067
3    1          10    48067
4    1          10    48067
5    1          10    48067
6    1          10    48067
7    1          10    48067
```

```
8    1          10    48067
9    1          10    48067
```

```
dfMaster.tail(10)
```

```
         MovieID                          Title  
Generes  \
1000199     3408         Erin Brockovich (2000)  
Drama   
1000200     3409       Final Destination (2000)          Drama|
Thriller   
1000201     3481           High Fidelity (2000)  
Comedy   
1000202     3483  Road to El Dorado, The (2000)    Animation|
Children's   
1000203     3484             Skulls, The (2000)  
Thriller   
1000204     3513     Rules of Engagement (2000)          Drama|
Thriller   
1000205     3535         American Psycho (2000)  Comedy|Horror|
Thriller   
1000206     3536       Keeping the Faith (2000)          Comedy|
Romance   
1000207     3555                   U-571 (2000)         Action|
Thriller   
1000208     3578               Gladiator (2000)            Action|
Drama   

         UserID  Rating  Timestamp Gender  Age  Occupation Zip Code  
1000199    5727       5  958489879      M   25           4    92843  
1000200    5727       4  958490143      M   25           4    92843  
1000201    5727       4  958489879      M   25           4    92843  
1000202    5727       3  958490143      M   25           4    92843  
1000203    5727       1  958489902      M   25           4    92843  
1000204    5727       4  958489970      M   25           4    92843  
1000205    5727       2  958489970      M   25           4    92843  
1000206    5727       5  958489902      M   25           4    92843  
1000207    5727       3  958490699      M   25           4    92843  
1000208    5727       5  958490171      M   25           4    92843  
```

```
# To csv file
dfMaster.to_csv('Master Data.csv')
```

Explore the datasets using visual representations (graphs or tables), also include your comments on the following: 1)User Age Distribution 2)User rating of the movie “Toy Story” 3)Top 25 movies by viewership rating Find the ratings for all the movies reviewed by for a particular user of user id = 2696

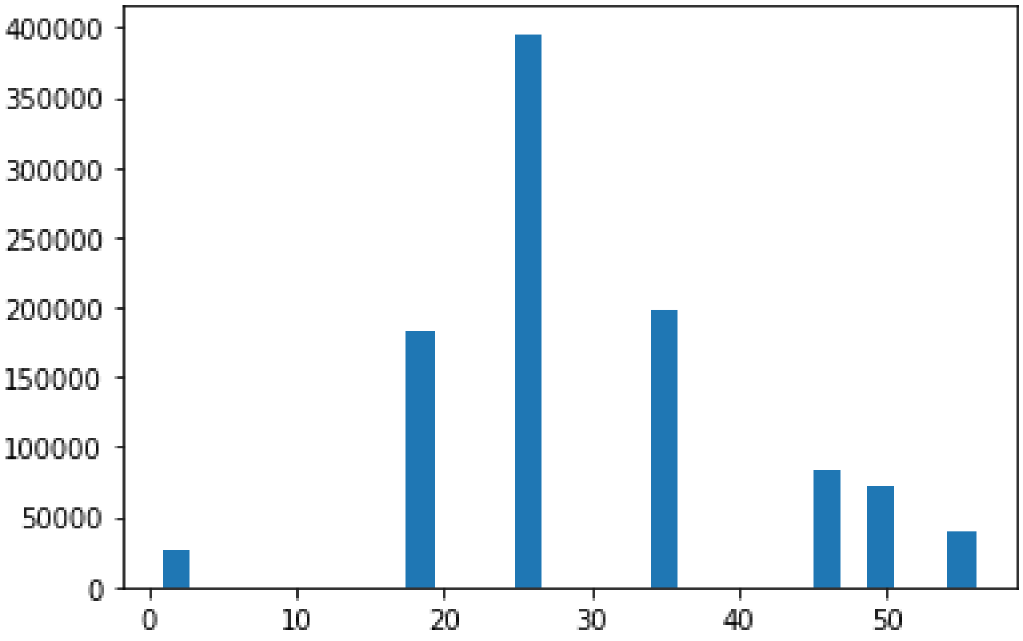
```
dfMaster.columns
```

```
Index(['MovieID', 'Title', 'Generes', 'UserID', 'Rating', 'Timestamp',
       'Gender', 'Age', 'Occupation', 'Zip Code'],
      dtype='object')
```

```
dfMaster.isna().sum(0)
```

```
MovieID       0
Title         0
Generes       0
UserID        0
Rating        0
Timestamp     0
Gender        0
Age           0
Occupation    0
Zip Code      0
dtype: int64
```

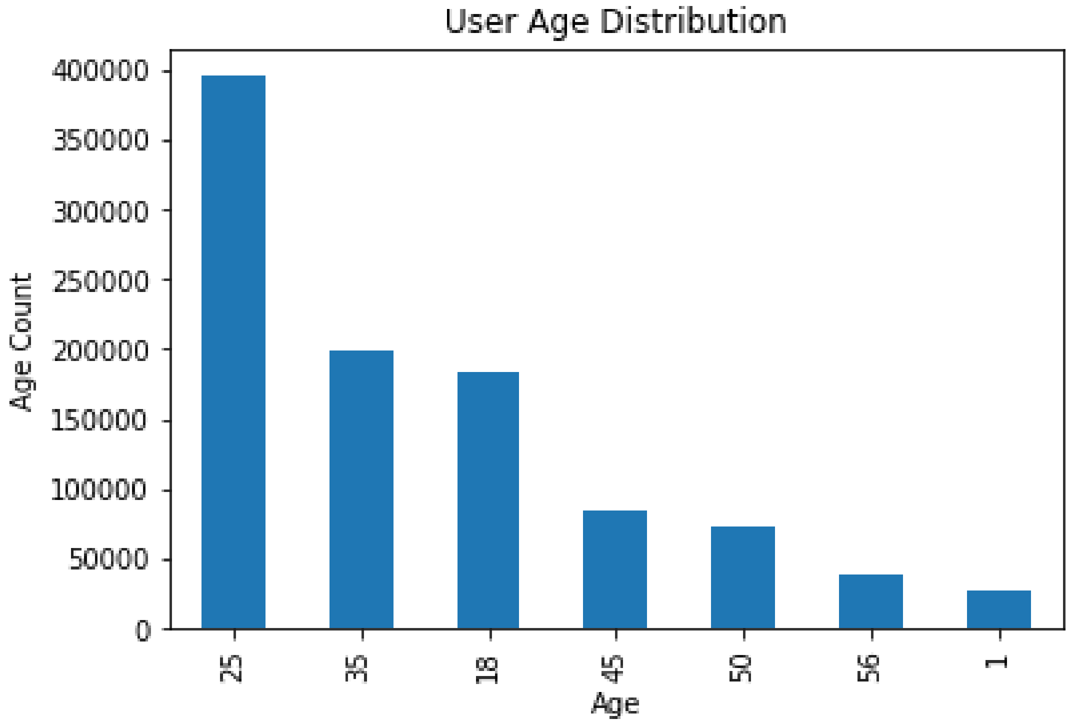
```
plt.hist(dfMaster["Age"],bins=30)
plt.show()
```



```
dfMaster['Age'].value_counts()
```

```
25    395556
35    199003
18    183536
45     83633
50     72490
56     38780
1      27211
Name: Age, dtype: int64
```

```
dfMaster['Age'].value_counts().plot(kind='bar')
plt.xlabel('Age')
plt.title('User Age Distribution')
plt.ylabel('Age Count')
plt.show()
```



The distribution Shows That Age count Max Is 25 And min is 1

2)User rating of the movie "Toy Story"

```
dfMaster.head()
```

```
   MovieID                                         Title  \
0        1                              Toy Story (1995)
1       48                             Pocahontas (1995)
2      150                              Apollo 13 (1995)
3      260  Star Wars: Episode IV - A New Hope (1977)
4      527                       Schindler's List (1993)

                                Generes  UserID  Rating  Timestamp
Gender  \
0           Animation|Children's|Comedy       1       5  978824268
F
1  Animation|Children's|Musical|Romance       1       5  978824351
F
2                                 Drama       1       5  978301777
F
3        Action|Adventure|Fantasy|Sci-Fi       1       4  978300760
```

```
F
4                              Drama|War       1       5  978824195
F

   Age  Occupation Zip Code
0    1          10    48067
1    1          10    48067
2    1          10    48067
3    1          10    48067
4    1          10    48067
```

```
toystory=dfMaster[dfMaster['Title'].str.contains('Toy Story')==True]
```

```
toystory
```

```
         MovieID               Title                         Generes
UserID  \
0              1    Toy Story (1995)  Animation|Children's|Comedy
1
50          3114  Toy Story 2 (1999)  Animation|Children's|Comedy
1
53             1    Toy Story (1995)  Animation|Children's|Comedy
6
124            1    Toy Story (1995)  Animation|Children's|Comedy
8
263            1    Toy Story (1995)  Animation|Children's|Comedy
9
...          ...                 ...                         ...
...
998988      3114  Toy Story 2 (1999)  Animation|Children's|Comedy
3023
999027      3114  Toy Story 2 (1999)  Animation|Children's|Comedy
5800
999486      3114  Toy Story 2 (1999)  Animation|Children's|Comedy
2189
999869      3114  Toy Story 2 (1999)  Animation|Children's|Comedy
159
1000192     3114  Toy Story 2 (1999)  Animation|Children's|Comedy
5727

         Rating  Timestamp Gender  Age  Occupation Zip Code
0             5  978824268      F    1          10    48067
50            4  978302174      F    1          10    48067
53            4  978237008      F   50           9    55117
124           4  978233496      M   25          12    11413
263           5  978225952      M   25          17    61614
...         ...        ...    ...  ...         ...      ...
998988        4  970471948      F   25           7    92108
999027        5  958015250      M   35          18    90804
999486        4  974607816      M    1          10    60148
999869        4  989966944      F   45           0    37922
```

```
1000192       5  958492554      M   25           4    92843

[3662 rows x 10 columns]
```

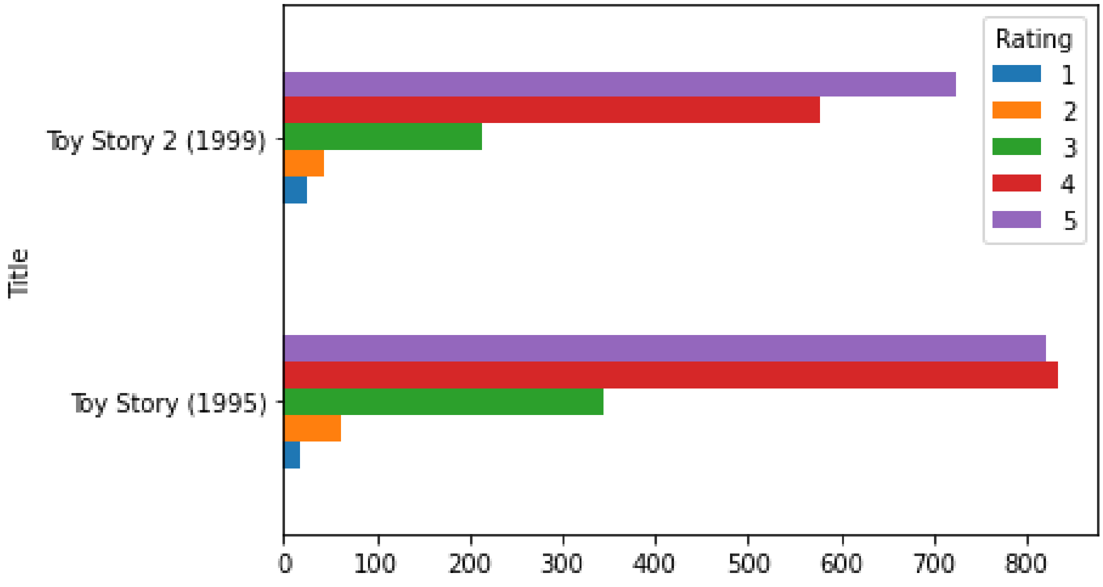
```
toystory.groupby(['Title','Rating']).size()
```

```
Title                 Rating
Toy Story (1995)      1          16
                      2          61
                      3         345
                      4         835
                      5         820
Toy Story 2 (1999)    1          25
                      2          44
                      3         214
                      4         578
                      5         724
dtype: int64
```

Observation Raing 5 has accured alot

```
toystory.groupby(['Title','Rating']).size().unstack().plot(kind='barh'
,legend=True) # Unstack will return Pivot
```

```
<AxesSubplot:ylabel='Title'>
```

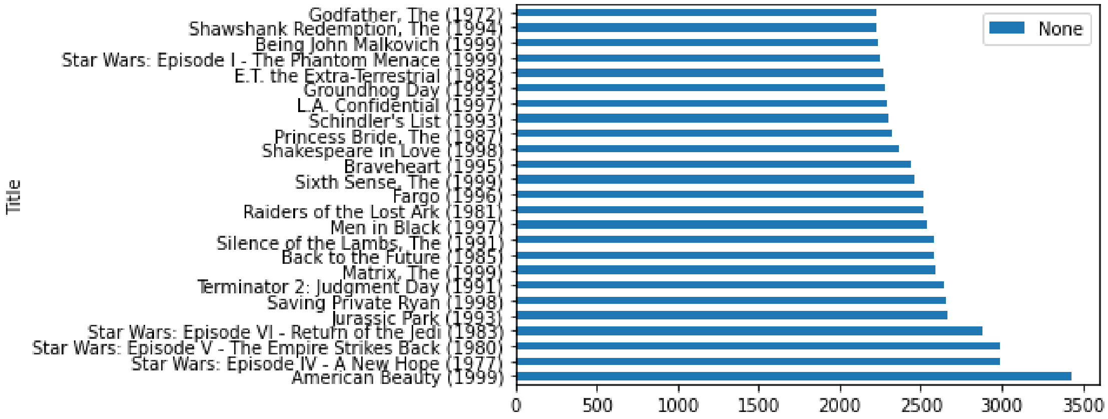


```
dfTop25=dfMaster.groupby(dfMaster['Title']).size().sort_values(ascendi
ng=False)[:25]
dfTop25
```

```
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
dtype: int64
```

```
dfTop25.plot(kind='barh',legend=True)
```

```
<AxesSubplot:ylabel='Title'>
```



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

```
user_2696 = dfMaster.loc[dfMaster.UserID==2696, "Rating"]
```

```
user_2696.shape
```

```
(20,)
```

Feature Engineering: Use column genres: 1-Find out all the unique genres (Hint: split the data in column genre making a list and then process the data to find out only the unique categories of genres) 2-Create a separate column for each genre category with a one-hot encoding (1 and 0) whether or not the movie belongs to that genre. 3-Determine the features affecting the ratings of any particular movie. 4-Develop an appropriate model to predict the movie ratings

1)Find out all the unique genres (Hint: split the data in column genre making a list and then process the data to find out only the unique categories of genres)

```
dfMaster['Generes']
```

```
0                   Animation|Children's|Comedy
1          Animation|Children's|Musical|Romance
2                                         Drama
3               Action|Adventure|Fantasy|Sci-Fi
4                                     Drama|War
                           ...
1000204                          Drama|Thriller
1000205                  Comedy|Horror|Thriller
1000206                          Comedy|Romance
1000207                         Action|Thriller
1000208                            Action|Drama
Name: Generes, Length: 1000209, dtype: object
```

```
dfGeneres=dfMaster['Generes'].str.split('|') # Split Convers Str to list
```

```
dfGeneres
```

```
0                    [Animation, Children's, Comedy]
1          [Animation, Children's, Musical, Romance]
2                                            [Drama]
3               [Action, Adventure, Fantasy, Sci-Fi]
4                                       [Drama, War]
                             ...
1000204                            [Drama, Thriller]
1000205                   [Comedy, Horror, Thriller]
1000206                            [Comedy, Romance]
1000207                           [Action, Thriller]
1000208                              [Action, Drama]
Name: Generes, Length: 1000209, dtype: object
```

```
listgeneres=set()
for genre in dfGeneres:
    listgeneres=listgeneres.union(set(genre))
```

```
listgeneres
```

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

```
len(listgeneres) # Count is 18
```

```
18
```

2-Create a separate column for each genre category with a one-hot encoding (1 and 0) whether or not the movie belongs to that genre.

```
GeneresOnehot=dfMaster['Generes'].str.get_dummies('|')
```

```
GeneresOnehot # used one hot Method And seperated
```

```
         Action  Adventure  Animation  Children's  Comedy  Crime
Documentary  \
0             0          0          1           1       1      0
0
1             0          0          1           1       0      0
0
2             0          0          0           0       0      0
0
3             1          1          0           0       0      0
0
4             0          0          0           0       0      0
0
...         ...        ...        ...         ...     ...    ...
...
1000204       0          0          0           0       0      0
0
1000205       0          0          0           0       1      0
0
1000206       0          0          0           0       1      0
0
1000207       1          0          0           0       0      0
0
1000208       1          0          0           0       0      0
```

```
0

         Drama  Fantasy  Film-Noir  Horror  Musical  Mystery  Romance
Sci-Fi  \
0            0        0          0       0        0        0        0
0
1            0        0          0       0        1        0        1
0
2            1        0          0       0        0        0        0
0
3            0        1          0       0        0        0        0
1
4            1        0          0       0        0        0        0
0
...        ...      ...        ...     ...      ...      ...      ...
...
1000204      1        0          0       0        0        0        0
0
1000205      0        0          0       1        0        0        0
0
1000206      0        0          0       0        0        0        1
0
1000207      0        0          0       0        0        0        0
0
1000208      1        0          0       0        0        0        0
0

         Thriller  War  Western
0               0    0        0
1               0    0        0
2               0    0        0
3               0    0        0
4               0    1        0
...           ...  ...      ...
1000204         1    0        0
1000205         1    0        0
1000206         0    0        0
1000207         1    0        0
1000208         0    0        0

[1000209 rows x 18 columns]
```

```
dfMaster=pd.concat([dfMaster,GeneresOnehot],axis=1)
```

```
dfMaster # Added At the end of the Data
```

```
         MovieID                                     Title  \
0              1                          Toy Story (1995)
1             48                         Pocahontas (1995)
2            150                          Apollo 13 (1995)
```

```
3            260  Star Wars: Episode IV - A New Hope (1977)
4            527                    Schindler's List (1993)
...          ...                                        ...
1000204     3513                 Rules of Engagement (2000)
1000205     3535                     American Psycho (2000)
1000206     3536                   Keeping the Faith (2000)
1000207     3555                               U-571 (2000)
1000208     3578                           Gladiator (2000)

                                      Generes  UserID  Rating
Timestamp  \
0                  Animation|Children's|Comedy       1       5
978824268
1        Animation|Children's|Musical|Romance       1       5
978824351
2                                       Drama       1       5
978301777
3             Action|Adventure|Fantasy|Sci-Fi       1       4
978300760
4                                   Drama|War       1       5
978824195
...                                       ...     ...     ...        .
..
1000204                        Drama|Thriller    5727       4
958489970
1000205                Comedy|Horror|Thriller    5727       2
958489970
1000206                        Comedy|Romance    5727       5
958489902
1000207                       Action|Thriller    5727       3
958490699
1000208                          Action|Drama    5727       5
958490171

        Gender  Age  Occupation Zip Code  ...  Fantasy  Film-Noir
Horror  \
0            F    1          10    48067  ...        0          0
0
1            F    1          10    48067  ...        0          0
0
2            F    1          10    48067  ...        0          0
0
3            F    1          10    48067  ...        1          0
0
4            F    1          10    48067  ...        0          0
0
...        ...  ...         ...      ...  ...      ...        ...
...
1000204      M   25           4    92843  ...        0          0
0
```

```
1000205      M   25          4    92843  ...        0          0
1
1000206      M   25          4    92843  ...        0          0
0
1000207      M   25          4    92843  ...        0          0
0
1000208      M   25          4    92843  ...        0          0
0

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

[1000209 rows x 28 columns]
```

```
dfMaster.to_csv('NewMaster.csv')
```

3-Determine the features affecting the ratings of any particular movie

convert Gender Male=0 Female-1 and convet to integer

```
dfMaster.dtypes
```

```
MovieID         int64
Title          object
Generes        object
UserID          int64
Rating          int64
Timestamp       int64
Gender         object
Age             int64
Occupation      int64
Zip Code       object
Action          int64
Adventure       int64
Animation       int64
Children's      int64
Comedy          int64
Crime           int64
Documentary     int64
Drama           int64
Fantasy         int64
```

```
Film-Noir       int64
Horror          int64
Musical         int64
Mystery         int64
Romance         int64
Sci-Fi          int64
Thriller        int64
War             int64
Western         int64
dtype: object
```

```
dfMaster.columns
```

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

```
dfMaster['Gender']=dfMaster['Gender'].replace('M','0')
dfMaster['Gender']=dfMaster['Gender'].replace('F','1') #Converted
Gender Male=0 Female-1 and convet to integer
```

```
dfMaster["Gender"].astype('int') #Convert type to int
```

```
0          1
1          1
2          1
3          1
4          1
          ..
1000204    0
1000205    0
1000206    0
1000207    0
1000208    0
Name: Gender, Length: 1000209, dtype: int64
```

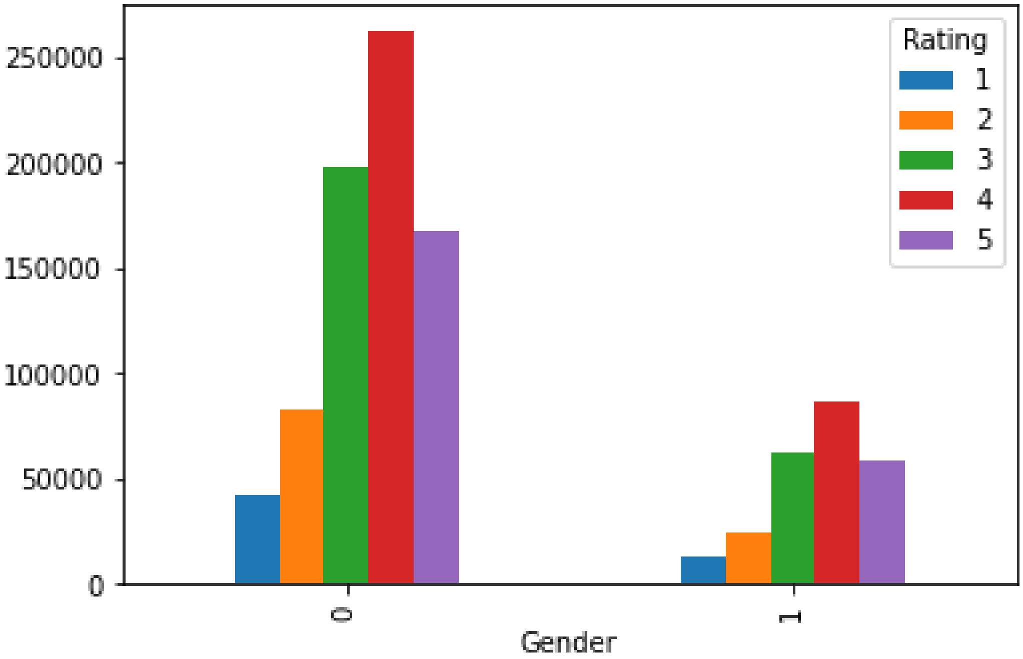
```
# Gender vs rating
GenderAffecting=dfMaster.groupby('Gender').size().sort_values(ascendin
g=False)[:25]
```

```
GenderAffecting #Male Tend to Rate More
```

```
Gender
0    753769
1    246440
dtype: int64
```

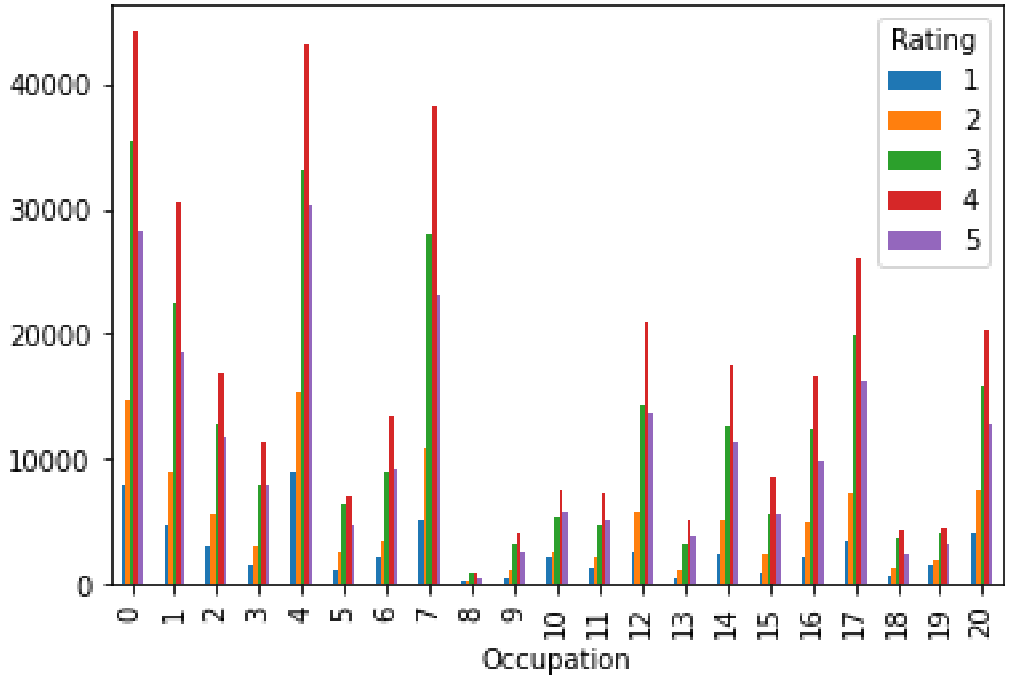
```
dfMaster.groupby(['Gender','Rating']).size().unstack().plot(kind='bar'
,legend=True)
```

```
<AxesSubplot:xlabel='Gender'>
```



```
#Occupation vs Rating
dfMaster.groupby(['Occupation','Rating']).size().unstack().plot(kind='
bar',legend=True)
```

```
<AxesSubplot:xlabel='Occupation'>
```



Observations 0- "other" or not specified "executive/managerial" 4-"college/grad student" The above Have Rated Alot 8- "farmer" Have Rated Less

4-Develop an appropriate model to predict the movie ratings

```
# First 500 Records
new_data=dfMaster[:500]
```

```
new_data.shape
```

```
(500, 28)
```

```
new_data
```

```
     MovieID                                         Title  \
0          1                              Toy Story (1995)
1         48                             Pocahontas (1995)
2        150                              Apollo 13 (1995)
3        260  Star Wars: Episode IV - A New Hope (1977)
4        527                       Schindler's List (1993)
..       ...                                           ...
495     1197                   Princess Bride, The (1987)
496     1198               Raiders of the Lost Ark (1981)
497     1200                                 Aliens (1986)
498     1201      Good, The Bad and The Ugly, The (1966)
499     1203                          12 Angry Men (1957)

                                  Generes  UserID  Rating  Timestamp
Gender  \
0             Animation|Children's|Comedy       1       5  978824268
1
1    Animation|Children's|Musical|Romance       1       5  978824351
1
2                                   Drama       1       5  978301777
1
3          Action|Adventure|Fantasy|Sci-Fi       1       4  978300760
1
4                               Drama|War       1       5  978824195
1
..                                    ...     ...     ...        ...
...
495       Action|Adventure|Comedy|Romance      10       5  979167660
1
496                      Action|Adventure      10       5  978225630
1
497            Action|Sci-Fi|Thriller|War      10       5  979168160
1
498                        Action|Western      10       2  978225853
1
499                                 Drama      10       3  979775159
1

     Age  Occupation Zip Code  ...  Fantasy  Film-Noir  Horror
Musical  \
0      1          10    48067  ...        0          0       0
0
1      1          10    48067  ...        0          0       0
1
2      1          10    48067  ...        0          0       0
0
3      1          10    48067  ...        1          0       0
0
4      1          10    48067  ...        0          0       0
0
..   ...         ...      ...  ...      ...        ...     ...      ..
.
495   35           1    95370  ...        0          0       0
0
496   35           1    95370  ...        0          0       0
0
497   35           1    95370  ...        0          0       0
0
498   35           1    95370  ...        0          0       0
0
499   35           1    95370  ...        0          0       0
0

     Mystery  Romance  Sci-Fi  Thriller  War  Western
0          0        0       0         0    0        0
```

```
1          0        1       0         0    0        0
2          0        0       0         0    0        0
3          0        0       1         0    0        0
4          0        0       0         0    1        0
..       ...      ...     ...       ...  ...      ...
495        0        1       0         0    0        0
496        0        0       0         0    0        0
497        0        0       1         1    1        0
498        0        0       0         0    0        1
499        0        0       0         0    0        0

[500 rows x 28 columns]
```

```
new_data.columns
```

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

```
features=new_data[['MovieID', 'Age', 'Occupation','Gender',]].values
```

```
features
```

```
array([[1, 1, 10, '1'],
       [48, 1, 10, '1'],
       [150, 1, 10, '1'],
       ...,
       [1200, 35, 1, '1'],
       [1201, 35, 1, '1'],
       [1203, 35, 1, '1']], dtype=object)
```

```
#Output
label=new_data[['Rating']].values
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(features,label,test_siz
e=0.20,random_state=42)
```

```
X_test.shape
```

```
(100, 4)
```

```
X_train.shape
```

```
(400, 4)
```

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

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

```
LinearRegression()
```

```
y_pred=lr.predict(X_test) # Model Applied to Predict the Ratings
```

```
y_pred
```

```
array([[3.46596348],
       [3.96471588],
       [4.31047093],
       [3.99788954],
       [3.68849118],
       [4.27254115],
       [4.30172875],
       [4.05316268],
       [4.02252707],
       [4.18751639],
       [4.2936504 ],
       [3.81923886],
       [4.26506799],
       [3.90352061],
       [4.31625205],
       [4.28057831],
       [4.14845859],
       [4.13671417],
       [3.75346841],
       [4.24955767],
       [4.15804678],
       [4.15720077],
       [3.90518754],
       [3.49627911],
       [3.96076779],
       [4.18032524],
       [4.14803558],
       [3.7680548 ],
       [3.75335251],
       [3.65335635],
       [4.17130106],
       [3.64912626],
       [4.19047745],
       [3.92527304],
       [4.35822974],
       [3.65561239],
       [4.24391756],
       [3.97811115],
       [3.78314211],
       [4.05270169],
       [4.28791046],
       [3.82230294],
       [4.16199486],
       [3.84659342],
       [4.10287347],
       [3.95963977],
       [4.37923917],
       [4.27785808],
       [4.23954647],
       [4.2111637 ],
       [3.97698313],
       [3.86196274],
       [4.04822758],
       [4.24293053],
       [3.82022588],
       [4.10191171],
       [4.01354086],
       [4.23442918],
       [3.80866364],
       [3.95329464],
       [4.14859959],
       [3.55584542],
       [3.73415101],
       [4.16340489],
       [3.9514616 ],
       [4.04943336],
       [4.18709338],
       [3.98499503],
       [3.6421392 ],
       [4.27747625],
       [3.51785255],
       [3.58736699],
       [3.91646778],
       [3.86711676],
       [3.74881532],
       [3.80296043],
       [3.95566659],
       [3.98854537],
       [4.18497834],
       [4.17398011],
       [4.00014559],
       [3.92396604],
       [4.16791699],
       [3.94882053],
       [3.73965013],
       [3.83217314],
       [3.87366598],
       [4.24067449],
       [3.77292682],
       [3.91237869],
       [4.29185854],
       [4.17200607],
       [3.45270921],
       [3.72625485],
       [4.2121919 ],
       [3.53308087],
       [3.88547224],
       [3.96147281],
       [4.19625857],
       [4.2497985 ]])
```

```
y_test # actual Rating
```

```
array([[4],
       [3],
       [4],
       [2],
       [5],
       [4],
       [4],
       [4],
       [5],
       [5],
       [4],
       [3],
       [3],
       [4],
       [4],
       [5],
       [5],
       [4],
       [5],
       [4],
       [4],
       [4],
       [3],
       [4],
       [5],
       [5],
       [5],
       [3],
       [3],
       [4],
       [5],
       [5],
       [4],
       [5],
       [5],
       [4],
       [4],
       [4],
[5],
[4],
[4],
[3],
[4],
[4],
[5],
[4],
[5],
[3],
[3],
[5],
[4],
[4],
[4],
[4],
[5],
[4],
[3],
[5],
[4],
[5],
[5],
[4],
[5],
[3],
[4],
[3],
[3],
[4],
[5],
[5],
[4],
[4],
[3],
[5],
[4],
[3],
[3],
[4],
[4],
[5],
[4],
[3],
[5],
[5],
[4],
[5],
[5],
[4],
       [3],
       [5],
       [2],
       [3],
       [3],
       [4],
       [5],
       [4],
       [4],
       [4],
       [3],
       [4]])
```

```
# Accuracy of the Data Or error
# error
from sklearn.metrics import mean_squared_error
print('Mean Squared Error',mean_squared_error(y_test,y_pred))
```

```
Mean Squared Error 0.6489142338657047
```

The error of the Developed Model is 65 %

```
from sklearn.metrics import r2_score
print('R2 score',r2_score(y_test,y_pred))
```

```
R2 score -0.07240825295935327
```