Project : Movilens Case Study -  PYTHON

```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

# machine learning
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
```

In [2]:

```
dfMovies = pd.read_csv("/Users/viviankoneri/Desktop/movies.dat",sep="::",names=["MovieID","Title","Genres"],engine='python')
dfMovies.head()
```

Out[2]:

| | 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 |

In [3]:

```
# Import Ratings Dataset
dfRatings = pd.read_csv("/Users/viviankoneri/Desktop/ratings.dat",sep="::",
names=["UserID","MovieID","Rating","Timestamp"],engine='python')
dfRatings.head()
```

Out[3]:

| | UserID | MovieID | Rating | Timestamp |
|---|---|---|---|---|
| 0 | 1 | 1193 | 5 | 978300760 |

|   | UserID | MovieID | Rating | Timestamp |
|---|---|---|---|---|
| 1 | 1 | 661 | 3 | 978302109 |
| 2 | 1 | 914 | 3 | 978301968 |
| 3 | 1 | 3408 | 4 | 978300275 |
| 4 | 1 | 2355 | 5 | 978824291 |

In [4]:

```
# Import Ratings Dataset
dfUsers = pd.read_csv("/Users/viviankoneri/Desktop/users.dat",sep="::",names=["UserID","Gender","Age","Occupation","Zip-code"],engine='python')
dfUsers.head()
```

Out[4]:

|   | 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 |

In [5]:

```
dfMovies.shape
```

Out[5]:

```
(3883, 3)
```

In [6]:

```
dfRatings.shape
```

Out[6]:

```
(1000209, 4)
```

In [7]:

```
dfUsers.shape
```

Out[7]:

```
(6040, 5)
```

In [8]:

```
# Create a New Master Data Set

dfMovieRatings = dfMovies.merge(dfRatings,on='MovieID',how='inner')
```

```
dfMovieRatings.head()
```

Out[8]:

|  | 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 [9]:

```
# Check to see that merging does not change the shape of the dataset

dfMovieRatings.shape
```

Out[9]:

```
(1000209, 6)
```

In [10]:

```
dfMaster = dfMovieRatings.merge(dfUsers,on="UserID",how='inner')
dfMaster.head()
```

Out[10]:

|  | 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 (1995) | 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 |

|  | Movie ID | Title | Genres | User ID | Rating | Timestamp | Gender | Age | Occupation | Zip-code |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 527 | Schindler's List (1993) | Drama\|War | 1 | 5 | 978824195 | F | 1 | 10 | 48067 |

In [11]:

```
dfMaster.to_csv("Master.csv")
```

In [12]:

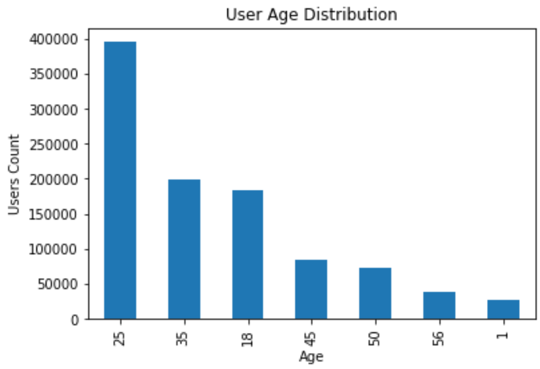
```
# Exploring the data with visual representations
#Users with Different Age Groups

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

Out[12]:

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

In [13]:

```
# Plot for users with different age groups

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



In [14]:

```
# User rating for Toy Story

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

Out[14]:

| | Movie ID | 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 |
| 50 | 3114 | Toy Story 2 (1999) | Animation\|Children's\|Comedy | 1 | 4 | 978302174 | 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 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 998988 | 3114 | Toy Story 2 (1999) | Animation\|Children's\|Comedy | 3023 | 4 | 970471948 | F | 25 | 7 | 92108 |
| 999027 | 3114 | Toy Story 2 (1999) | Animation\|Children's\|Comedy | 5800 | 5 | 958015250 | M | 35 | 18 | 90804 |
| 999486 | 3114 | Toy Story 2 | Animation\|Children's\|Comedy | 2189 | 4 | 974607816 | M | 1 | 10 | 60148 |

<table>
<tr><th></th><th>Movie ID</th><th>Title</th><th>Genres</th><th>UserID</th><th>Rating</th><th>Timestamp</th><th>Gender</th><th>Age</th><th>Occupation</th><th>Zip-code</th></tr>
<tr><td></td><td></td><td>(1999)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>999869</td><td>3114</td><td>Toy Story 2 (1999)</td><td>Animation|Children's|Comedy</td><td>159</td><td>4</td><td>989966944</td><td>F</td><td>45</td><td>0</td><td>37922</td></tr>
<tr><td>1000192</td><td>3114</td><td>Toy Story 2 (1999)</td><td>Animation|Children's|Comedy</td><td>5727</td><td>5</td><td>958492554</td><td>M</td><td>25</td><td>4</td><td>92843</td></tr>
</table>

3662 rows × 10 columns

In [15]:

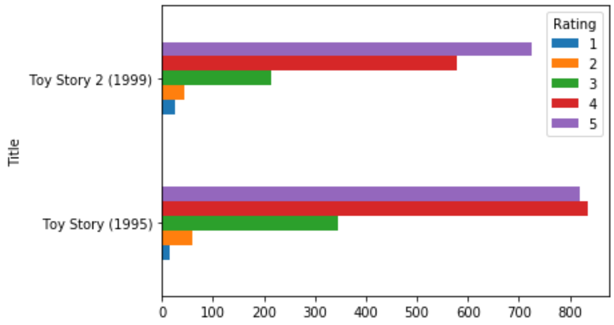
```
toystoryRating.groupby(["Title","Rating"]).size()
```

Out[15]:

```
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
```

In [16]:

```
toystoryRating.groupby(["Title","Rating"]).size().unstack().plot(kind='barh
',stacked=False,legend=True)
plt.show()
```



In [17]:

```
# Top 25 movies by viewrship rating
```

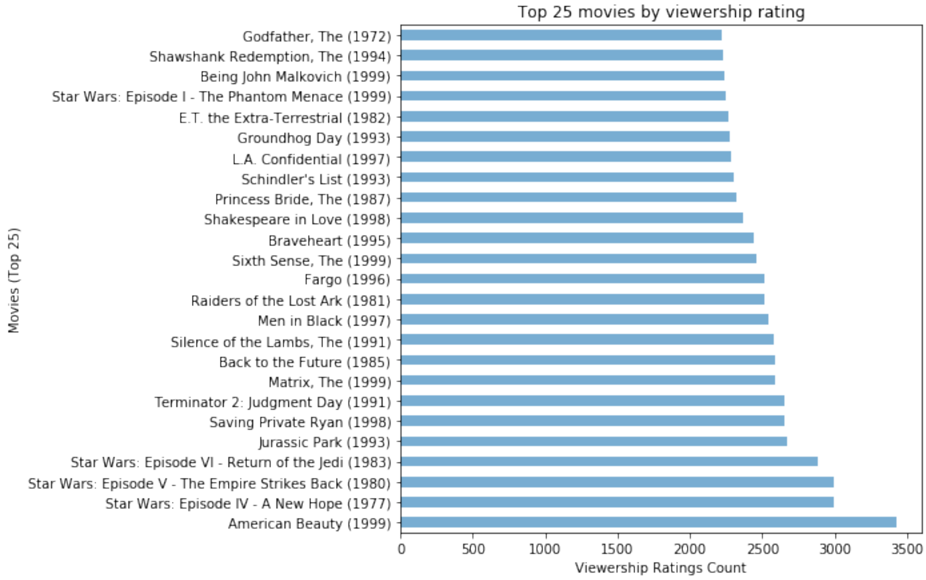
```
dfTop25 = dfMaster.groupby('Title').size().sort_values(ascending=False)[:25
]
dfTop25
```

Out[17]:

```
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
```

In [18]:

```
dfTop25.plot(kind='barh',alpha=0.6,figsize=(7,7))
plt.xlabel("Viewership Ratings Count")
plt.ylabel("Movies (Top 25)")
plt.title("Top 25 movies by viewership rating")
plt.show()
```



In [19]:

```
# Find the ratings for all the movies reviewed by user id = 2696

userId = 2696
userRatingById = dfMaster[dfMaster["UserID"] == userId]
userRatingById
```

Out[19]:

| | Movie ID | Title | Genres | User ID | 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 |

|  | Movie ID | Title | Genres | User ID | Rating | Timestamp | Gender | Age | Occupation | Zip-code |
|---|---|---|---|---|---|---|---|---|---|---|
| 991040 | 1270 | Back to the Future (1985) | Comedy\|Sci-Fi | 2696 | 2 | 973308676 | M | 25 | 7 | 24210 |
| 991041 | 1589 | Cop Land (1997) | Crime\|Drama\|Mystery | 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 (1998) | 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 | Perfect Murder, A (1998) | Mystery\|Thriller | 2696 | 4 | 973308904 | M | 25 | 7 | 24210 |

|  | Movie ID | Title | Genres | User ID | Rating | Timestamp | Gender | Age | Occupation | Zip-code |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 [20]:

```
# Feature Engineering

# Finding out all unique GENRES

dfGenres = dfMaster['Genres'].str.split("|")

dfGenres
```

Out[20]:

```
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: Genres, Length: 1000209, dtype: object
```

In [21]:

```
listGenres = set()
for genre in dfGenres:
    listGenres = listGenres.union(set(genre))
```

In [22]:

```
listGenres
```

Out[22]:

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

In [23]:

```
dfMaster.shape
```

Out[23]:

```
(1000209, 10)
```

In [24]:

```
# Had to dis-integrate the complete data in five smaller data frames as my
processor
# is not able to handle 1 million records in one go.

dfA = dfMaster.iloc[0:200000,:]

ratingsOneHotA = dfA['Genres'].str.get_dummies("|")

ratingsOneHotA
```

Out[24]:

| | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 199995 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 199996 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 199997 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 199998 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 199999 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

200000 rows × 18 columns

In [25]:

```
dfB = dfMaster.iloc[200000:400000,:]

ratingsOneHotB = dfB['Genres'].str.get_dummies("|")
```

```
ratingsOneHotB
```

Out[25]:

| | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 200000 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 200001 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 200002 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 200003 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 200004 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 399995 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 399996 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 399997 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 399998 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

| | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 399999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

200000 rows × 18 columns

In [26]:

```
dfC = dfMaster.iloc[400000:600000,:]

ratingsOneHotC = dfC['Genres'].str.get_dummies("|")

ratingsOneHotC
```

Out[26]:

| | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 400000 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 400001 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 400002 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 400003 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 400004 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

|  | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 599995 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 599996 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 599997 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 599998 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 599999 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

200000 rows × 18 columns

In [27]:

```
dfD = dfMaster.iloc[600000:800000,:]

ratingsOneHotD = dfD['Genres'].str.get_dummies("|")

ratingsOneHotD
```

Out[27]:

|  | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 600000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

|  | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 600001 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 600002 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 600003 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 600004 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 799995 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 799996 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 799997 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 799998 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 799999 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

200000 rows × 18 columns

In [28]:

```
dfE = dfMaster.iloc[800000:1000210,:]
```

```
ratingsOneHotE = dfE['Genres'].str.get_dummies("|")

ratingsOneHotE
```

Out[28]:

| | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 800000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 800001 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 800002 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 800003 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 800004 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1000204 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1000205 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1000206 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

|  | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1000207 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1000208 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

200209 rows × 18 columns

In [29]:

```
dfA = pd.concat([dfA,ratingsOneHotA],axis=1)
```

In [30]:

```
dfA.shape
```

Out[30]:

```
(200000, 28)
```

In [31]:

```
dfB = pd.concat([dfB,ratingsOneHotB],axis=1)
dfB.shape
```

Out[31]:

```
(200000, 28)
```

In [32]:

```
dfC = pd.concat([dfC,ratingsOneHotC],axis=1)
dfC.shape
```

Out[32]:

```
(200000, 28)
```

In [33]:

```
dfD = pd.concat([dfD,ratingsOneHotD],axis=1)
dfD.shape
```

Out[33]:

```
(200000, 28)
```

In [34]:

```
dfE = pd.concat([dfE,ratingsOneHotE],axis=1)
dfE.shape
```

Out[34]:

```
(200209, 28)
```

In [35]:

```
# Concatenation of all five dataframes to create one Master_final data frame.
```

```
frames = [dfA, dfB, dfC, dfD, dfE]

Master_final = pd.concat(frames)

Master_final.shape
```

Out[35]:

```
(1000209, 28)
```

In [36]:

```
Master_final.columns
```

Out[36]:

```
Index(['MovieID', 'Title', 'Genres', 'UserID', 'Rating', 'Timestamp', 'Gend
er',
       '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')
```

In [37]:

```
Master_final.to_csv("Final_Master.csv")
```

In [38]:

```
# Determining the features affecting the ratings of a particular movie.

Master_final[["title","Year"]] = Master_final.Title.str.extract("(.)\s\((.\
d+)",expand=True)
```

In [39]:

```
Master_final = Master_final.drop(columns=["title"])
Master_final.head()
```

Out[39]:

| | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Toy Story (1995) | Animation\|Children's\|Comedy | 1 | 5 | 978824268 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 |
| 1 | 48 | Pocahontas | Animation\|Children's\|Musical\|Romance | 1 | 5 | 978824351 | F | 1 | 10 | 48067 | ... | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1995 |

| | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | (1995) | | | | | | | | | | | | | | | | | | | |
| 2 | 150 | Apollo 13 (1995) | Drama | 1 | 5 | 978301777 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 |
| 3 | 260 | Star Wars: Episode IV - A New Hope (1977) | Action\|Adventure\|Fantasy\|Sci-Fi | 1 | 4 | 978300760 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1977 |
| 4 | 527 | Schindler's List (1993) | Drama\|War | 1 | 5 | 978824195 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1993 |

5 rows × 29 columns

In [40]:

```
Master_final.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1000209 entries, 0 to 1000208
Data columns (total 29 columns):
```

```
MovieID        1000209 non-null int64
Title          1000209 non-null object
Genres         1000209 non-null object
UserID         1000209 non-null int64
Rating         1000209 non-null int64
Timestamp      1000209 non-null int64
Gender         1000209 non-null object
Age            1000209 non-null int64
Occupation     1000209 non-null int64
Zip-code       1000209 non-null object
Action         1000209 non-null int64
Adventure      1000209 non-null int64
Animation      1000209 non-null int64
Children's     1000209 non-null int64
Comedy         1000209 non-null int64
Crime          1000209 non-null int64
Documentary    1000209 non-null int64
Drama          1000209 non-null int64
Fantasy        1000209 non-null int64
Film-Noir      1000209 non-null int64
Horror         1000209 non-null int64
Musical        1000209 non-null int64
Mystery        1000209 non-null int64
Romance        1000209 non-null int64
Sci-Fi         1000209 non-null int64
Thriller       1000209 non-null int64
War            1000209 non-null int64
Western        1000209 non-null int64
Year           1000209 non-null object
dtypes: int64(24), object(5)
memory usage: 228.9+ MB
```

In [41]:

```
# Converting the data type Column Year from object to Integer for ease of calculation and plotting purpose.

Master_final['Year'] = Master_final.Year.astype(int)
```

In [42]:

```
Master_final['Movie_Age'] = 2000 - Master_final.Year
Master_final.head()
```

Out[42]:

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Toy Story (1995) | Animation\|Children's\|Comedy | 1 | 5 | 978824268 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 1 | 48 | Pocahontas (1995) | Animation\|Children's\|Musical\|Romance | 1 | 5 | 978824351 | F | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 2 | 150 | Apollo 13 (1995) | Drama | 1 | 5 | 978301777 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 3 | 260 | Star Wars: Episode IV - A New Hope (1977) | Action\|Adventure\|Fantasy\|Sci-Fi | 1 | 4 | 978300760 | F | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1977 | 23 |
| 4 | 527 | Schind | Drama\|War | 1 | 5 | 978824 | F | 1 | 10 | 480 | .. | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 19 | 7 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | ler's List (1993) |  |  |  | 195 |  |  |  | 67 |  |  |  |  |  |  |  |  |  | 93 |  |

5 rows × 30 columns

In [43]:

```
# Changing the data type of Gender from Object to integer 1 for 'F' and integer 0 for 'M' for ease of plotting

Master_final['Gender'] = Master_final.Gender.str.replace('F','1')
```

In [44]:

```
Master_final['Gender'] = Master_final.Gender.str.replace('M','0')
```

In [45]:

```
Master_final['Gender'] = Master_final.Gender.astype(int)
```

In [46]:

```
Master_final.head()
```

Out[46]:

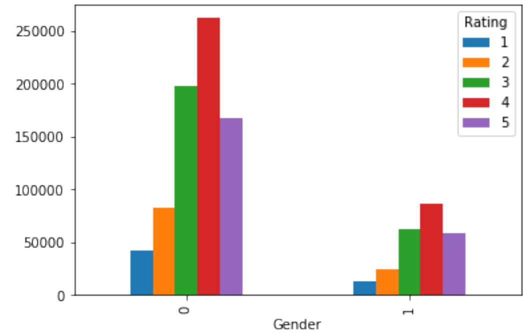
|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Toy Story (1995) | Animation\|Children's\|Comedy | 1 | 5 | 978824268 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 1 | 48 | Pocahontas (1 | Animation\|Children's\|Musical\|Romance | 1 | 5 | 978824351 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1995 | 5 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | 995) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 150 | Apollo 13 (1995) | Drama | 1 | 5 | 978301777 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 3 | 260 | Star Wars: Episode IV - A New Hope (1977) | Action\|Adventure\|Fantasy\|Sci-Fi | 1 | 4 | 978300760 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1977 | 23 |
| 4 | 527 | Schindler's List (1993) | Drama\|War | 1 | 5 | 978824195 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1993 | 7 |

5 rows × 30 columns

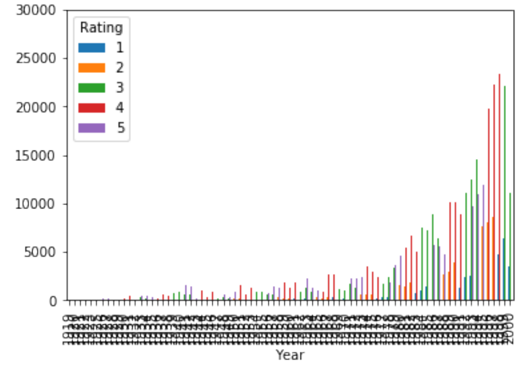
In [47]:

```
# Plot shows how ratings of movies based on Gender

Master_final.groupby(["Gender","Rating"]).size().unstack().plot(kind='bar',
stacked=False,legend=True)
plt.show()
```



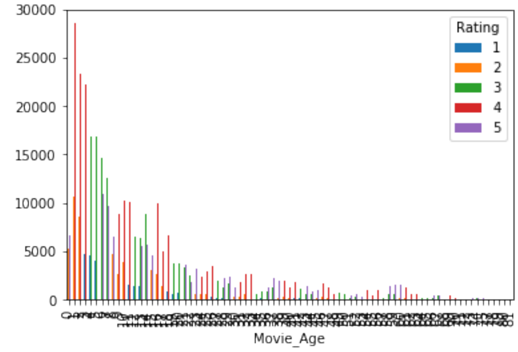
In [48]:

```
# Visual representation as to how the Year of release affectst the Ratinf of a movie

Master_final.groupby(["Year","Rating"]).size().unstack().plot(kind='bar',stacked=False,legend=True)
plt.show()
```



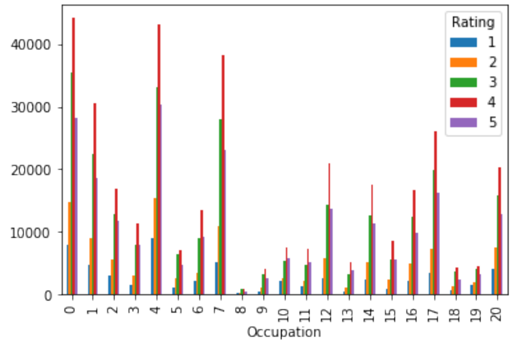
In [49]:

```
# Plot shows how Age of Movie affects the Rating of a movie

Master_final.groupby(["Movie_Age","Rating"]).size().unstack().plot(kind='bar',stacked=False,legend=True)
plt.show()
```



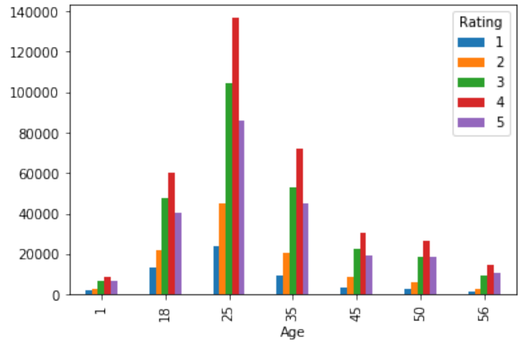
In [50]:

```
# Plot shows how Occupation of viewers affects the Rating of a movie

Master_final.groupby(["Occupation","Rating"]).size().unstack().plot(kind='bar',stacked=False,legend=True)
plt.show()
```



In [51]:

```
# Finally Age of a Viewer affecting the Rating of a Movie

Master_final.groupby(["Age","Rating"]).size().unstack().plot(kind='bar',stacked=False,legend=True)
plt.show()
```



In [52]:

```
# Developing an appropriate model to predict the movie ratings

# First 1000 extracted records

first_1000 = Master_final[0:1000]

first_1000.head(25)
```

Out[52]:

| | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | Toy Story (1995) | Animation\|Children's\|Comedy | 1 | 5 | 978824268 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 | 5 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 48 | Pocahontas (1995) | Animation\|Children's\|Musical\|Romance | 1 | 5 | 978824351 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 2 | 150 | Apollo 13 (1995) | Drama | 1 | 5 | 978301777 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1995 | 5 |
| 3 | 260 | Star Wars: Episode IV - A New Hope (1977) | Action\|Adventure\|Fantasy\|Sci-Fi | 1 | 4 | 978300760 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1977 | 23 |
| 4 | 527 | Schindler's List (1993) | Drama\|War | 1 | 5 | 978824195 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1993 | 7 |
| 5 | 531 | Secret Garden, The | Children's\|Drama | 1 | 4 | 978302149 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1993 | 7 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | (1993) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 588 | Aladdin (1992) | Animation\|Children's\|Comedy\|Musical | 1 | 4 | 978824268 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1992 | 8 |
| 7 | 594 | Snow White and the Seven Dwarfs (1937) | Animation\|Children's\|Musical | 1 | 4 | 978302268 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1937 | 63 |
| 8 | 595 | Beauty and the Beast (1991) | Animation\|Children's\|Musical | 1 | 5 | 978824268 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1991 | 9 |
| 9 | 608 | Fargo (1996) | Crime\|Drama\|Thriller | 1 | 4 | 978301398 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1996 | 4 |
| 10 | 661 | James and the | Animation\|Children's\|Musical | 1 | 3 | 978302109 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1996 | 4 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | Giant Peach (1996) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | 720 | Wallace & Gromit: The Best of Aardman Animatio... | Animation | 1 | 3 | 978300760 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1996 | 4 |
| 12 | 745 | Close Shave, A (1995) | Animation\|Comedy\|Thriller | 1 | 3 | 978824268 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1995 | 5 |
| 13 | 783 | Hunchback of Notre Dame, The | Animation\|Children's\|Musical | 1 | 4 | 978824291 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1996 | 4 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | (1996) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | 914 | My Fair Lady (1964) | Musical\|Romance | 1 | 3 | 978301968 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1964 | 36 |
| 15 | 919 | Wizard of Oz, The (1939) | Adventure\|Children's\|Drama\|Musical | 1 | 4 | 978301368 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1939 | 61 |
| 16 | 938 | Gigi (1958) | Musical | 1 | 4 | 978301752 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1958 | 42 |
| 17 | 1022 | Cinderella (1950) | Animation\|Children's\|Musical | 1 | 5 | 978300055 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1950 | 50 |
| 18 | 1028 | Mary Poppins (1964) | Children's\|Comedy\|Musical | 1 | 5 | 978301777 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1964 | 36 |
| 19 | 1029 | Dumbo | Animation\|Children's\|Musical | 1 | 5 | 978302 | 1 | 1 | 10 | 480 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1941 | 59 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | (1941) |  |  |  | 205 |  |  |  | 67 |  |  |  |  |  |  |  |  |  |  |  |
| 20 | 1035 | Sound of Music, The (1965) | Musical | 1 | 5 | 978301753 | 1 | 1 | 10 | 48067 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1965 | 35 |
| 21 | 1097 | E.T. the Extra-Terrestrial (1982) | Children's\|Drama\|Fantasy\|Sci-Fi | 1 | 4 | 978301953 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1982 | 18 |
| 22 | 1193 | One Flew Over the Cuckoo's Nest (1975) | Drama | 1 | 5 | 978300760 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1975 | 25 |
| 23 | 1197 | Princess Bride, Th | Action\|Adventure\|Comedy\|Romance | 1 | 3 | 978302268 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1987 | 13 |

|  | MovieID | Title | Genres | UserID | Rating | Timestamp | Gender | Age | Occupation | Zip-code | ... | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western | Year | Movie_Age |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | e (1987) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 24 | 1207 | To Kill a Mockingbird (1962) | Drama | 1 | 4 | 978300719 | 1 | 1 | 10 | 48067 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1962 | 38 |

25 rows × 30 columns

In [53]:

```
#Using the following features: Movie id, Occupation and Age

features = first_1000[['MovieID','Age','Occupation']].values
```

In [54]:

```
#Using Rating as label

labels = first_1000[['Rating']].values
```

In [55]:

```
features
```

Out[55]:

```
array([[   1,    1,   10],
       [  48,    1,   10],
       [ 150,    1,   10],
       ...,
       [2394,   18,    3],
       [2402,   18,    3],
       [2404,   18,    3]], dtype=int64)
```

In [56]:

```
labels
```

Out[56]:

```
array([[5],
       [5],
       [5],
       [4],
       [5],
       [4],
        [4],
        [4],
        [5],
        [4],
        [3],
        [3],
        [3],
        [4],
        [3],
        [4],
        [4],
        [5],
        [5],
        [5],
        [5],
        [4],
        [5],
        [3],
        [4],
        [4],
        [5],
        [5],
        [4],
        [4],
        [4],
        [5],
        [4],
        [5],
        [4],
        [4],
        [5],
        [4],
        [3],
        [3],
        [5],
        [4],
        [3],
        [4],
        [4],
        [4],
        [4],
        [5],
        [4],
        [5],
        [4],
        [4],
        [4],
[4],
[4],
[4],
[5],
[5],
[4],
[2],
[4],
[4],
[3],
[4],
[4],
[4],
[3],
[4],
[5],
[4],
[4],
[5],
[4],
[3],
[4],
[4],
[5],
[4],
[5],
[4],
[4],
[3],
[3],
[5],
[4],
[4],
[4],
[4],
[5],
[3],
[5],
[3],
[4],
[3],
[4],
[3],
[3],
[3],
[4],
[5],
[3],
[3],
[4],
[1],
[5],
[4],
[5],
[5],
[5],
[3],
[3],
[4],
[5],
[4],
[4],
[3],
[5],
[3],
[4],
[3],
[3],
[4],
[4],
[5],
[4],
[3],
[4],
[4],
[4],
[4],
[5],
[4],
[3],
[3],
[5],
[4],
[4],
[5],
[5],
[4],
[4],
[3],
[5],
[4],
[5],
[4],
[4],
        [4],
        [3],
        [5],
        [5],
        [5],
        [3],
        [4],
        [4],
        [2],
        [3],
        [5],
        [3],
        [5],
        [3],
        [3],
        [3],
        [5],
        [4],
        [3],
        [4],
        [5],
        [5],
        [5],
        [5],
        [3],
        [5],
        [5],
        [4],
        [3],
        [4],
        [4],
        [5],
        [5],
        [5],
        [3],
        [4],
        [5],
        [5],
        [4],
        [3],
        [3],
        [4],
        [4],
        [4],
        [3],
        [5],
        [5],
[3],
[5],
[5],
[4],
[3],
[4],
[5],
[3],
[5],
[4],
[4],
[3],
[2],
[4],
[4],
[5],
[3],
[3],
[5],
[3],
[3],
[5],
[3],
[3],
[2],
[3],
[5],
[3],
[5],
[5],
[2],
[4],
[2],
[3],
[5],
[2],
[4],
[3],
[5],
[5],
[3],
[3],
[5],
[3],
[5],
[3],
[4],
    [5],
    [5],
    [3],
    [3],
    [2],
    [4],
    [3],
    [4],
    [3],
    [3],
    [5],
    [3],
    [4],
    [3],
    [5],
    [5],
    [3],
    [3],
    [3],
    [4],
    [3],
    [4],
    [5],
    [4],
    [4],
    [5],
    [4],
    [3],
    [4],
    [4],
    [4],
    [5],
    [4],
    [3],
    [3],
    [3],
    [3],
    [5],
    [4],
    [3],
    [4],
    [5],
    [5],
    [5],
    [5],
    [3],
    [4],
    [4],
    [4],
    [5],
    [3],
    [3],
    [2],
    [4],
    [3],
    [4],
    [4],
    [4],
    [4],
    [4],
    [4],
    [4],
    [4],
    [3],
    [4],
    [4],
    [3],
    [3],
    [3],
    [4],
    [3],
    [3],
    [4],
    [4],
    [2],
    [5],
    [4],
    [4],
    [4],
    [3],
    [4],
    [4],
    [5],
    [4],
    [3],
    [3],
    [4],
    [5],
    [5],
    [5],
    [4],
    [4],
    [4],
    [4],
[4],
[5],
[4],
[5],
[2],
[4],
[4],
[4],
[5],
[4],
[3],
[4],
[4],
[3],
[3],
[3],
[4],
[4],
[3],
[4],
[2],
[4],
[2],
[2],
[3],
[3],
[4],
[3],
[4],
[2],
[4],
[2],
[3],
[3],
[5],
[5],
[4],
[3],
[5],
[4],
[5],
[3],
[4],
[3],
[5],
[3],
[2],
[3],
[4],
[4],
[2],
[5],
[5],
[5],
[4],
[3],
[5],
[4],
[5],
[4],
[4],
[5],
[3],
[5],
[4],
[4],
[5],
[5],
[4],
[4],
[4],
[3],
[4],
[4],
[4],
[4],
[5],
[5],
[4],
[4],
[3],
[3],
[5],
[4],
[4],
[5],
[4],
[5],
[5],
[4],
[4],
[4],
[4],
[3],
[5],
[3],
[5],
[4],
[4],
[5],
[3],
[5],
[5],
[3],
[4],
[4],
[5],
[4],
[3],
[3],
[5],
[5],
[5],
[5],
[5],
[5],
[3],
[3],
[4],
[2],
[4],
[4],
[5],
[5],
[5],
[3],
[5],
[5],
[3],
[3],
[5],
[4],
[3],
[5],
[5],
[4],
[5],
[3],
[4],
[4],
[5],
[5],
[5],
[5],
[5],
[5],
[4],
[5],
[3],
[5],
[4],
[4],
[5],
[4],
[4],
[4],
[4],
[4],
[5],
[5],
[5],
[5],
[5],
[2],
[3],
[4],
[4],
[4],
[4],
[4],
[3],
[5],
[4],
[4],
[4],
[4],
[5],
[5],
[3],
[3],
[3],
[5],
[3],
[3],
[5],
[3],
[5],
[5],
        [4],
        [5],
        [3],
        [5],
        [5],
        [3],
        [5],
        [3],
        [5],
        [5],
        [4],
        [4],
        [5],
        [3],
        [5],
        [3],
        [5],
        [3],
        [5],
        [5],
        [4],
        [4],
        [4],
        [4],
        [4],
        [3],
        [5],
        [3],
        [3],
        [4],
        [5],
        [4],
        [5],
        [3],
        [3],
        [4],
        [3],
        [5],
        [5],
        [4],
        [5],
        [4],
        [3],
        [4],
        [2],
        [5],
        [5],
[4],
[5],
[4],
[5],
[3],
[4],
[5],
[5],
[2],
[3],
[4],
[5],
[3],
[5],
[4],
[3],
[5],
[5],
[5],
[5],
[4],
[4],
[4],
[4],
[3],
[4],
[4],
[5],
[5],
[4],
[4],
[3],
[4],
[3],
[5],
[3],
[4],
[4],
[5],
[4],
[3],
[4],
[3],
[4],
[4],
[5],
[3],
        [5],
        [4],
        [3],
        [4],
        [4],
        [5],
        [5],
        [5],
        [5],
        [4],
        [4],
        [5],
        [3],
        [5],
        [4],
        [4],
        [4],
        [4],
        [3],
        [5],
        [5],
        [5],
        [5],
        [5],
        [4],
        [4],
        [5],
        [5],
        [5],
        [4],
        [4],
        [5],
        [5],
        [4],
        [4],
        [4],
        [3],
        [4],
        [5],
        [5],
        [4],
        [4],
        [5],
        [5],
        [5],
        [5],
        [5],
        [4],
        [5],
        [4],
        [4],
        [4],
        [5],
        [4],
        [3],
        [5],
        [5],
        [4],
        [5],
        [3],
        [5],
        [4],
        [4],
        [5],
        [4],
        [4],
        [5],
        [5],
        [4],
        [4],
        [3],
        [5],
        [5],
        [4],
        [3],
        [4],
        [4],
        [5],
        [5],
        [3],
        [4],
        [3],
        [4],
        [4],
        [4],
        [4],
        [3],
        [5],
        [4],
        [2],
        [4],
        [4],
        [5],
        [4],
       [5],
       [3],
       [3],
       [5],
       [4],
       [2],
       [5],
       [4],
       [5],
       [4],
       [4],
       [3],
       [4],
       [3],
       [3],
       [4],
       [5],
       [3],
       [4],
       [3],
       [5],
       [5],
       [3],
       [5],
       [4],
       [3],
       [4],
       [3],
       [2],
       [5],
       [5],
       [5],
       [5],
       [4],
       [4],
       [5],
       [5],
       [2],
       [5],
       [4],
       [3],
       [3],
       [4],
       [4],
       [5],
       [3],
       [5],
        [3],
        [3],
        [2],
        [2],
        [4],
        [5],
        [5],
        [4],
        [3],
        [3],
        [4],
        [4],
        [4],
        [2],
        [5],
        [4],
        [4],
        [3],
        [5],
        [4],
        [1],
        [4],
        [5],
        [3],
        [1],
        [5],
        [3],
        [4],
        [5],
        [1],
        [1],
        [3],
        [4],
        [4],
        [5],
        [5],
        [5],
        [3],
        [4],
        [1],
        [4],
        [5],
        [5],
        [4],
        [3],
        [5],
        [5],
        [4],
        [3],
        [2],
        [3],
        [1],
        [5],
        [4],
        [2],
        [3],
        [3],
        [4],
        [2],
        [3],
        [5],
        [5],
        [1],
        [2],
        [5],
        [2],
        [4],
        [4],
        [5],
        [5],
        [4],
        [4],
        [5],
        [5],
        [4],
        [5],
        [4],
        [3],
        [1],
        [1],
        [5],
        [5],
        [3],
        [3],
        [4],
        [1],
        [5],
        [5],
        [5],
        [4],
        [5],
        [1],
        [3],
        [5],
            [4],
            [5],
            [4],
            [4],
            [4],
            [3],
            [1],
            [2],
            [4],
            [5],
            [4],
            [5],
            [1],
            [5],
            [1],
            [4],
            [5],
            [5],
            [5],
            [4],
            [4],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [5],
            [3],
            [5],
            [1],
            [2],
            [4],
            [3],
            [4],
            [3],
            [4],
            [5],
            [5],
            [1],
            [4],
[4],
[3],
[1],
[1],
[5],
[4],
[1],
[2],
[5],
[1],
[4],
[5],
[4],
[3],
[5],
[4],
[5],
[4],
[1],
[4],
[4],
[4],
[3],
[1],
[4],
[1],
[3],
[4],
[5],
[3],
[1],
[5],
[3],
[3],
[4],
[5],
[5],
[5],
[5],
[4],
[3],
[4],
[5],
[3],
[2],
[4],
[5],
[5],
[4],
[2],
[5],
[1],
[3],
[3],
[3],
[4],
[5],
[5],
[4],
[5],
[4],
[4],
[4],
[5],
[3],
[5],
[4],
[4],
[5],
[4],
[3],
[3],
[5],
[5],
[4],
[5],
[3],
[5],
[3],
[3],
[5],
[4],
[5],
[4],
[4],
[1],
[4],
[3],
[5],
[4],
[5],
[3],
[5],
[5],
       [3],
       [4],
       [2],
       [1],
       [4],
       [2],
       [2]], dtype=int64)
```

In [57]:

```
#Createing training and test data set

train, test, train_labels, test_labels = train_test_split(features,labels,test_size=0.33,random_state=42)
```

In [58]:

```
train
```

Out[58]:

```
array([[3035,   35,    1],
       [1393,   25,   17],
       [3198,   35,    1],
       ...,
       [1073,   18,    3],
       [ 784,   35,    1],
       [2802,   50,    9]], dtype=int64)
```

In [59]:

```
test
```

Out[59]:

```
array([[1265,   35,    1],
       [3408,   35,    1],
       [3447,   35,    1],
       [2423,   35,    1],
       [ 539,   35,    1],
       [2657,   35,    1],
       [2109,   35,    1],
       [1247,   35,    1],
       [1064,   18,    3],
       [ 105,   25,   12],
       [ 421,   18,    3],
       [1035,   50,    9],
       [2140,   35,    1],
       [2140,   18,    3],
       [2000,   18,    3],
       [1527,   18,    3],
       [ 508,   25,   17],
       [1275,   18,    3],
       [3704,   35,    1],
       [1653,   25,   17],
       [1380,   35,    1],
       [1027,   25,   12],
[   7,   35,    1],
[1282,   35,    1],
[2006,   25,   12],
[2712,   25,   12],
[2506,   50,    9],
[2278,   18,    3],
[1562,   18,    3],
[2021,   18,    3],
[3114,   25,   17],
[ 150,   25,   12],
[2094,   35,    1],
[1203,   35,    1],
[   2,   35,    1],
[1693,   25,   12],
[2795,   35,    1],
[1948,   35,    1],
[1552,   18,    3],
[ 296,   50,    9],
[1923,   25,   17],
[2100,   50,    9],
[1446,   25,   17],
[2141,   18,    3],
[1148,   25,   17],
[ 428,   25,   17],
[1801,   18,    3],
[2033,   35,    1],
[ 898,   35,    1],
[ 737,   18,    3],
[1676,   35,    1],
[1246,   18,    3],
[3500,   25,   12],
[1884,   35,    1],
[1197,    1,   10],
[1721,    1,   10],
[2087,   35,    1],
[ 661,    1,   10],
[2336,   25,   12],
[ 532,   18,    3],
[ 994,   25,   17],
[  17,   50,    9],
[1357,   35,    1],
[1916,   25,   12],
[2043,   35,    1],
[2863,   35,    1],
[2431,   35,    1],
[1186,   18,    3],
[ 912,   50,    9],
[1371,    35,     1],
[3408,    50,     9],
[1148,    35,     1],
[2001,    35,     1],
[3451,    35,     1],
[ 838,    25,    17],
[ 745,    25,    17],
[2375,    35,     1],
[2340,     1,    10],
[2000,    35,     1],
[1310,    25,    17],
[2662,    35,     1],
[ 590,    50,     9],
[ 377,    25,    17],
[ 595,    50,     9],
[1639,    25,    17],
[1377,    35,     1],
[2402,    18,     3],
[3108,    35,     1],
[3668,    35,     1],
[1921,    25,    17],
[ 186,    35,     1],
[ 923,    35,     1],
[1269,    35,     1],
[2320,    25,    12],
[ 168,    18,     3],
[ 802,    35,     1],
[3809,    35,     1],
[1688,    50,     9],
[ 383,    50,     9],
[ 588,    18,     3],
[3153,    35,     1],
[3253,    25,    17],
[1967,    18,     3],
[3267,    25,    12],
[2138,    35,     1],
[2302,    35,     1],
[1411,    35,     1],
[2085,    18,     3],
[ 589,    25,    12],
[1690,    18,     3],
[1283,    35,     1],
[ 552,    18,     3],
[2268,    18,     3],
[ 485,    18,     3],
[1569,    50,     9],
[ 750,    35,     1],
[1411,   25,   12],
[2153,   18,    3],
[1294,   35,    1],
[ 912,   25,   17],
[1387,   18,    3],
[ 597,   35,    1],
[3100,   35,    1],
[3481,   25,   12],
[2908,   25,   12],
[1566,   35,    1],
[1947,   35,    1],
[ 914,   35,    1],
[1193,   18,    3],
[ 302,   18,    3],
[2018,   35,    1],
[1005,   18,    3],
[  25,   25,   17],
[2384,   18,    3],
[1286,   35,    1],
[  34,   50,    9],
[3685,   50,    9],
[2291,   25,   12],
[1246,    1,   10],
[ 920,   50,    9],
[2762,    1,   10],
[3213,   25,   12],
[3194,   35,    1],
[ 524,   25,   17],
[1391,   18,    3],
[1682,   18,    3],
[ 412,   18,    3],
[3155,   25,   12],
[ 551,   18,    3],
[1682,   25,   17],
[2300,   35,    1],
[ 476,   25,   12],
[2145,   18,    3],
[ 720,   35,    1],
[2005,   18,    3],
[1009,   35,    1],
[1356,   25,   17],
[2948,   35,    1],
[ 364,   50,    9],
[1282,   18,    3],
[2009,   35,    1],
[3869,   35,    1],
[2336,   35,    1],
[2355,   35,    1],
[1185,   18,    3],
[2498,   35,    1],
[1089,   25,   17],
[2804,   35,    1],
[1500,   25,   17],
[1358,   25,   17],
[3623,   25,   17],
[1079,   35,    1],
[3524,   50,    9],
[2336,   18,    3],
[1104,   35,    1],
[3751,   25,   17],
[3260,   25,   12],
[3425,   25,   12],
[ 327,   18,    3],
[2640,   35,    1],
[1196,   35,    1],
[2529,   35,    1],
[ 110,   35,    1],
[1278,   35,    1],
[ 919,   18,    3],
[ 110,   25,   12],
[3301,   25,   17],
[3793,   25,   17],
[2006,   18,    3],
[3591,   35,    1],
[2025,   18,    3],
[ 592,   18,    3],
[2398,   35,    1],
[1701,   25,   12],
[2268,   25,   12],
[ 480,   35,    1],
[2268,   25,   17],
[1840,   25,   12],
[2078,   35,    1],
[1088,   50,    9],
[1566,    1,   10],
[1302,   35,    1],
[2918,   35,    1],
[1517,   35,    1],
[ 736,   18,    3],
[2011,   35,    1],
[ 538,   25,   12],
[1702,   18,    3],
[1197,   35,    1],
[ 555,   18,    3],
[ 588,   50,    9],
[ 161,   25,   12],
[1804,   18,    3],
[ 589,   18,    3],
[2396,   35,    1],
[2166,   25,   17],
[2453,   35,    1],
[3250,   25,   12],
[1617,   18,    3],
[3086,   35,    1],
[1265,   25,   17],
[3363,   35,    1],
[2142,   18,    3],
[3105,    1,   10],
[1411,   18,    3],
[ 150,    1,   10],
[1372,   35,    1],
[3178,   25,   17],
[1587,   18,    3],
[1304,   35,    1],
[2959,   25,   17],
[2375,   18,    3],
[1084,   35,    1],
[1756,   35,    1],
[1961,    1,   10],
[1836,    1,   10],
[2688,   25,   12],
[2093,   18,    3],
[1639,   25,   12],
[2278,   25,   17],
[ 260,    1,   10],
[1735,   25,   12],
[1250,   35,    1],
[ 296,   18,    3],
[1307,   25,   17],
[3508,   50,    9],
[ 743,   35,    1],
[1043,   50,    9],
[1441,   50,    9],
[ 597,   25,   17],
[1296,   50,    9],
[2355,   18,    3],
[ 222,   18,    3],
[1396,   18,    3],
[ 339,   35,    1],
[1721,   25,   17],
[1257,   35,    1],
[1688,   18,    3],
[1739,   18,    3],
[ 531,    1,   10],
[3347,   35,    1],
[1016,   35,    1],
[2321,   50,    9],
[  47,   25,   17],
[3452,   25,   17],
[1196,   18,    3],
[ 292,   18,    3],
[ 153,   35,    1],
[2133,   35,    1],
[ 180,   35,    1],
[1129,   35,    1],
[3624,   50,    9],
[1544,   18,    3],
[3252,   25,   12],
[1270,   35,    1],
[ 608,    1,   10],
[1678,   25,   12],
[2041,   35,    1],
[1210,   50,    9],
[ 110,   18,    3],
[1961,   35,    1],
[ 291,   18,    3],
[3006,   25,   12],
[ 778,   25,   17],
[2023,   25,   12],
[3155,   35,    1],
[3510,   25,   17],
[ 919,   35,    1],
[2541,   25,   12],
[2042,   18,    3],
[2020,   18,    3],
[ 288,   18,    3],
[1210,   35,    1],
[1015,   35,    1],
[3717,   25,   17],
[1022,   35,    1],
[1962,   18,    3],
[ 671,   35,    1],
[ 594,    1,   10],
[ 393,   25,   12],
[1356,   35,    1],
[ 899,   35,    1],
[1088,   35,    1],
[ 594,   35,    1],
       [  47,   18,    3],
       [2028,   18,    3],
       [2294,   25,   17],
       [1784,   35,    1],
       [1960,   18,    3],
       [1639,   35,    1],
       [2006,   35,    1],
       [1287,   35,    1],
       [1923,   35,    1],
       [3072,   35,    1],
       [ 918,   35,    1],
       [ 926,   35,    1],
       [2571,   25,   12],
       [3255,   25,   17],
       [3264,   35,    1],
       [2028,   25,   12],
       [1101,   50,    9],
       [ 282,   25,   12],
       [1221,   25,   17],
       [2115,   35,    1],
       [  44,   18,    3],
       [1223,   35,    1],
       [2858,   25,   17],
       [1101,   35,    1],
       [3097,   35,    1],
       [  42,   25,   12]], dtype=int64)
```

In [60]:

```
train_labels
```

Out[60]:

```
array([[3],
       [3],
       [3],
       [3],
       [5],
       [5],
       [2],
       [4],
       [4],
       [5],
       [5],
       [4],
       [5],
       [3],
       [4],
       [5],
       [4],
       [5],
[5],
[3],
[3],
[3],
[5],
[4],
[3],
[4],
[4],
[4],
[3],
[5],
[5],
[4],
[4],
[5],
[4],
[4],
[4],
[5],
[5],
[4],
[5],
[4],
[4],
[5],
[4],
[4],
[5],
[3],
[5],
[3],
[5],
[4],
[5],
[3],
[5],
[4],
[4],
[3],
[3],
[4],
[4],
[4],
[3],
[5],
[5],
[5],
[4],
[4],
[5],
[4],
[4],
[3],
[5],
[5],
[4],
[5],
[4],
[5],
[4],
[4],
[5],
[4],
[3],
[4],
[3],
[4],
[4],
[4],
[3],
[5],
[5],
[4],
[3],
[5],
[4],
[3],
[5],
[4],
[5],
[5],
[5],
[3],
[4],
[4],
[5],
[3],
[4],
[3],
[5],
[5],
[5],
[4],
[3],
[3],
[4],
[3],
[4],
[4],
[5],
[5],
[3],
[5],
[5],
[5],
[4],
[4],
[5],
[5],
[5],
[5],
[4],
[4],
[5],
[4],
[3],
[5],
[4],
[5],
[3],
[5],
[4],
[4],
[5],
[3],
[5],
[4],
[3],
[4],
[3],
[4],
[4],
[4],
[3],
[5],
[4],
[4],
[3],
[5],
[5],
            [5],
            [5],
            [5],
            [3],
            [3],
            [4],
            [5],
            [3],
            [3],
            [5],
            [5],
            [5],
            [4],
            [4],
            [4],
            [5],
            [4],
            [5],
            [4],
            [5],
            [3],
            [3],
            [3],
            [4],
            [4],
            [4],
            [3],
            [5],
            [5],
            [4],
            [5],
            [4],
            [4],
            [5],
            [4],
            [5],
            [3],
            [5],
            [4],
            [5],
            [3],
            [4],
            [5],
            [5],
            [5],
            [5],
            [5],
[4],
[4],
[3],
[5],
[4],
[4],
[5],
[5],
[3],
[2],
[5],
[4],
[4],
[4],
[5],
[4],
[4],
[5],
[5],
[3],
[5],
[3],
[4],
[3],
[5],
[2],
[4],
[3],
[5],
[3],
[5],
[3],
[4],
[4],
[5],
[4],
[5],
[4],
[3],
[5],
[5],
[4],
[4],
[5],
[3],
[5],
[1],
[4],
[3],
[4],
[5],
[4],
[3],
[3],
[2],
[4],
[4],
[4],
[1],
[4],
[4],
[4],
[5],
[5],
[5],
[4],
[4],
[5],
[4],
[5],
[4],
[3],
[3],
[4],
[5],
[4],
[3],
[3],
[4],
[4],
[5],
[3],
[1],
[4],
[4],
[5],
[5],
[3],
[4],
[5],
[4],
[5],
[4],
[3],
        [3],
        [5],
        [5],
        [4],
        [5],
        [4],
        [5],
        [4],
        [5],
        [3],
        [2],
        [2],
        [4],
        [4],
        [4],
        [5],
        [5],
        [3],
        [5],
        [3],
        [3],
        [4],
        [1],
        [4],
        [3],
        [5],
        [3],
        [4],
        [3],
        [1],
        [3],
        [4],
        [3],
        [5],
        [4],
        [2],
        [5],
        [1],
        [4],
        [4],
        [4],
        [5],
        [5],
        [5],
        [3],
        [4],
        [5],
[4],
[5],
[4],
[4],
[1],
[4],
[5],
[2],
[5],
[3],
[4],
[4],
[4],
[5],
[5],
[5],
[5],
[3],
[4],
[3],
[4],
[4],
[5],
[1],
[5],
[2],
[3],
[3],
[4],
[4],
[4],
[4],
[3],
[2],
[5],
[3],
[3],
[3],
[3],
[5],
[5],
[5],
[4],
[5],
[4],
[4],
[4],
    [2],
    [4],
    [4],
    [5],
    [5],
    [3],
    [4],
    [3],
    [3],
    [4],
    [2],
    [3],
    [5],
    [4],
    [5],
    [5],
    [3],
    [4],
    [4],
    [5],
    [3],
    [4],
    [5],
    [3],
    [3],
    [4],
    [5],
    [5],
    [2],
    [5],
    [5],
    [4],
    [4],
    [5],
    [5],
    [5],
    [5],
    [3],
    [5],
    [5],
    [4],
    [3],
    [5],
    [5],
    [4],
    [5],
    [4],
        [3],
        [4],
        [5],
        [4],
        [4],
        [5],
        [2],
        [3],
        [4],
        [4],
        [4],
        [5],
        [3],
        [4],
        [2],
        [3],
        [3],
        [3],
        [4],
        [3],
        [5],
        [5],
        [3],
        [1],
        [3],
        [4],
        [5],
        [5],
        [5],
        [3],
        [4],
        [5],
        [3],
        [4],
        [5],
        [3],
        [5],
        [5],
        [4],
        [3],
        [5],
        [4],
        [4],
        [3],
        [4],
        [4],
        [3],
[3],
[5],
[4],
[5],
[3],
[4],
[3],
[5],
[1],
[2],
[4],
[5],
[3],
[3],
[5],
[3],
[5],
[4],
[4],
[1],
[3],
[3],
[4],
[3],
[5],
[4],
[5],
[5],
[5],
[3],
[4],
[5],
[4],
[4],
[5],
[5],
[5],
[5],
[5],
[4],
[4],
[5],
[3],
[4],
[4],
[4],
[5],
[4],
[2],
[5],
[3],
[5],
[4],
[4],
[5],
[3],
[5],
[5],
[4],
[2],
[3],
[3],
[5],
[4],
[5],
[3],
[5],
[3],
[4],
[4],
[5],
[4],
[2],
[5],
[3],
[3],
[4],
[5],
[4],
[5],
[4],
[1],
[4],
[4],
[4],
[3],
[3],
[5],
[4],
[5],
[4],
[5],
[2],
[2],
[4],
[3],
[4],
[5],
[5],
[4],
[5],
[4],
[3],
[4],
[3],
[4],
[4],
[2],
[2],
[3],
[2],
[1],
[4],
[3],
[5],
[5],
[4],
[2],
[4],
[5],
[4],
[4],
[3],
[5],
[4],
[5],
[5],
[4],
[4],
[5],
[3],
[4],
[5],
[4],
[4],
[4],
[1],
[4],
[5],
[4],
[4],
       [4],
       [4],
       [5],
       [4],
       [5],
       [3],
       [4],
       [4],
       [5],
       [3],
       [3],
       [5],
       [3],
       [2],
       [4],
       [4],
       [4],
       [5],
       [4],
       [3],
       [5],
       [5],
       [5],
       [4],
       [5],
       [3],
       [4],
       [5],
       [5],
       [3],
       [4],
       [4],
       [4],
       [5],
       [4],
       [5],
       [5],
       [4],
       [4],
       [3],
       [4]], dtype=int64)
```

In [61]:

```
test_labels
```

Out[61]:

```
array([[5],
       [4],
       [5],
[5],
[5],
[4],
[4],
[3],
[2],
[4],
[4],
[5],
[5],
[4],
[4],
[4],
[3],
[5],
[2],
[4],
[5],
[4],
[4],
[5],
[3],
[3],
[3],
[3],
[1],
[4],
[4],
[4],
[4],
[3],
[5],
[3],
[3],
[4],
[1],
[2],
[5],
[3],
[3],
[5],
[4],
[3],
[1],
[3],
[4],
[1],
[4],
[5],
[3],
[2],
[3],
[4],
[5],
[3],
[3],
[1],
[4],
[4],
[5],
[5],
[5],
[4],
[5],
[1],
[4],
[4],
[5],
[5],
[4],
[5],
[3],
[4],
[4],
[3],
[5],
[3],
[4],
[3],
[3],
[4],
[4],
[3],
[2],
[5],
[4],
[4],
[3],
[3],
[5],
[2],
[1],
[5],
[5],
[5],
[4],
[5],
[2],
[4],
[5],
[5],
[3],
[5],
[4],
[4],
[5],
[1],
[3],
[2],
[4],
[2],
[4],
[4],
[5],
[3],
[4],
[4],
[4],
[4],
[3],
[4],
[3],
[3],
[5],
[5],
[4],
[5],
[4],
[1],
[4],
[2],
[5],
[4],
[3],
[5],
[4],
[4],
[4],
[3],
[4],
[4],
    [3],
    [4],
    [5],
    [3],
    [5],
    [4],
    [5],
    [3],
    [3],
    [5],
    [5],
    [5],
    [3],
    [4],
    [4],
    [5],
    [4],
    [3],
    [5],
    [4],
    [5],
    [4],
    [4],
    [5],
    [4],
    [4],
    [4],
    [5],
    [3],
    [5],
    [4],
    [4],
    [3],
    [3],
    [3],
    [5],
    [5],
    [4],
    [4],
    [5],
    [5],
    [5],
    [2],
    [4],
    [3],
    [5],
    [2],
[4],
[5],
[4],
[3],
[4],
[4],
[4],
[4],
[5],
[4],
[5],
[5],
[3],
[1],
[4],
[3],
[4],
[5],
[4],
[4],
[3],
[3],
[5],
[5],
[4],
[4],
[3],
[5],
[4],
[4],
[5],
[3],
[5],
[5],
[5],
[4],
[3],
[4],
[3],
[4],
[4],
[4],
[4],
[5],
[5],
[3],
[3],
[5],
[4],
[4],
[4],
[3],
[5],
[4],
[3],
[3],
[4],
[4],
[3],
[3],
[5],
[3],
[4],
[5],
[5],
[5],
[4],
[1],
[4],
[5],
[5],
[3],
[5],
[2],
[5],
[4],
[3],
[4],
[2],
[4],
[4],
[3],
[3],
[4],
[4],
[5],
[3],
[3],
[5],
[5],
[1],
[3],
[5],
[3],
       [5],
       [3],
       [5],
       [3],
       [1],
       [5],
       [4],
       [4],
       [3],
       [3],
       [5],
       [5],
       [3],
       [4],
       [2],
       [5],
       [4],
       [5],
       [5],
       [1],
       [5],
       [4],
       [5],
       [5],
       [2],
       [3],
       [3],
       [5],
       [4],
       [5],
       [4],
       [5],
       [4],
       [3],
       [5],
       [4],
       [3],
       [4],
       [5],
       [4],
       [5],
       [4],
       [4],
       [5],
       [3]], dtype=int64)
```

In [62]:

```
# Logistic Regression
```

```
logreg = LogisticRegression()
logreg.fit(train, train_labels)
Y_pred = logreg.predict(test)
acc_log = round(logreg.score(train, train_labels) * 100, 2)
acc_log
```

```
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/sklearn/linea
r_model/logistic.py:432: FutureWarning: Default solver will be changed to '
lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/sklearn/utils
/validation.py:724: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), f
or example using ravel().
  y = column_or_1d(y, warn=True)
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/sklearn/linea
r_model/logistic.py:469: FutureWarning: Default multi_class will be changed
to 'auto' in 0.22. Specify the multi_class option to silence this warning.
  "this warning.", FutureWarning)
```

Out[62]:

```
39.25
```

In [63]:

```
# Support Vector Machines

svc = SVC()
svc.fit(train, train_labels)
Y_pred = svc.predict(test)
acc_svc = round(svc.score(train, train_labels) * 100, 2)
acc_svc
```

```
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/sklearn/utils
/validation.py:724: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), f
or example using ravel().
  y = column_or_1d(y, warn=True)
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/sklearn/svm/b
ase.py:193: FutureWarning: The default value of gamma will change from 'aut
o' to 'scale' in version 0.22 to account better for unscaled features. Set
gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[63]:

```
95.82
```

In [64]:

```
# Gaussian Naive Bayes

gaussian = GaussianNB()
gaussian.fit(train, train_labels)
Y_pred = gaussian.predict(test)
```

```
acc_gaussian = round(gaussian.score(train, train_labels) * 100, 2)
acc_gaussian
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/sklearn/utils
/validation.py:724: DataConversionWarning: A column-vector y was passed whe
n a 1d array was expected. Please change the shape of y to (n_samples, ), f
or example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[64]:

```
39.55
```

In [65]:

```
# Decision Tree

decision_tree = DecisionTreeClassifier()
decision_tree.fit(train, train_labels)
Y_pred = decision_tree.predict(test)
acc_decision_tree = round(decision_tree.score(train, train_labels) * 100, 2
)
acc_decision_tree
```

Out[65]:

```
100.0
```

In [66]:

```
# K Nearest Neighbors Classifier

knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(train, train_labels)
Y_pred = knn.predict(test)
acc_knn = round(knn.score(train, train_labels) * 100, 2)
acc_knn
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:4: DataConversionWarning: A column-vector y was passed when a 1d a
rray was expected. Please change the shape of y to (n_samples, ), for examp
le using ravel().
  after removing the cwd from sys.path.
```

Out[66]:

```
59.7
```

In [67]:

```
# Random Forest

random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(train, train_labels)
Y_pred = random_forest.predict(test)
random_forest.score(train, train_labels)
acc_random_forest = round(random_forest.score(train, train_labels) * 100, 2
)
acc_random_forest
/Users/viviankoneri/opt/anaconda3/lib/python3.7/site-packages/ipykernel_lau
ncher.py:4: DataConversionWarning: A column-vector y was passed when a 1d a
```

```
rray was expected. Please change the shape of y to (n_samples,), for exampl
e using ravel().
  after removing the cwd from sys.path.
```

Out[67]:

```
100.0
```

In [68]:

```
models = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression',
              'Random Forest', 'Naive Bayes',
              'Decision Tree'],
    'Score': [acc_svc, acc_knn, acc_log,
              acc_random_forest, acc_gaussian,
              acc_decision_tree]})
models.sort_values(by='Score', ascending=False)
```

Out[68]:

| | Model | Score |
|---|---|---|
| 3 | Random Forest | 100.00 |
| 5 | Decision Tree | 100.00 |
| 0 | Support Vector Machines | 95.82 |
| 1 | KNN | 59.70 |
| 4 | Naive Bayes | 39.55 |
| 2 | Logistic Regression | 39.25 |

In [69]:

```
# Random Forest, Decision Tree and Support Vector Machines are the Predicti
on Models recommended for this Project.
```