MostRatedGenres

April 1, 2021

Most Rated Genres

0.1 IMDB Movie Dataset

The data are contained in six files `links.csv`, `movies.csv`, `ratings.csv` and `tags.csv` etc.In this project we will use only two given data set which is **movies.csv** and **ratings.csv**. Our research question is **What types of movies genres user viewed and rated most than other movies genres ?**

DataSet can be get from this site : http://grouplens.org/datasets/

Importing Pandas

```
[1]: import pandas as pd
     %matplotlib inline
```

Importing or Acquiring movies.csv and ratings.csv data sets

```
[2]: movies = pd.read_csv('/home/roshan/Documents/datascience/edX/
      ↪PythonForDataScience/ml-25m/movies.csv',sep = ',')
     ratings = pd.read_csv('/home/roshan/Documents/datascience/edX/
      ↪PythonForDataScience/ml-25m/ratings.csv',sep = ',')
```

```
[3]: movies.head() # showing the first 15 items in csv file
```

```
[3]:    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)

                                              genres
     0  Adventure|Animation|Children|Comedy|Fantasy
     1                   Adventure|Children|Fantasy
     2                               Comedy|Romance
     3                         Comedy|Drama|Romance
     4                                       Comedy
```

```
[4]: ratings.head(5) # showing the first 15 items in csv file
```

```
[4]:    userId  movieId  rating   timestamp
     0       1      296     5.0  1147880044
     1       1      306     3.5  1147868817
     2       1      307     5.0  1147868828
     3       1      665     5.0  1147878820
     4       1      899     3.5  1147868510
```

```
[5]: # deleting the timestamp and userId coloumns
     del ratings['timestamp']
     del ratings['userId']
```

```
[6]: ratings.head() # after
```

```
[6]:    movieId  rating
     0      296     5.0
     1      306     3.5
     2      307     5.0
     3      665     5.0
     4      899     3.5
```

Merge Dataframes

```
[7]: # take the average ratings value and group them by concern MovieId ....
     avg_ratings = ratings.groupby('movieId', as_index=False).mean()
     avg_ratings.head()
```

```
[7]:    movieId    rating
     0        1  3.893708
     1        2  3.251527
     2        3  3.142028
     3        4  2.853547
     4        5  3.058434
```

```
[8]: # we can visualize ratings values by box ploting
     avg_ratings.boxplot(column='rating',figsize=(10,5))
```

```
[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3cec33bdd0>
```

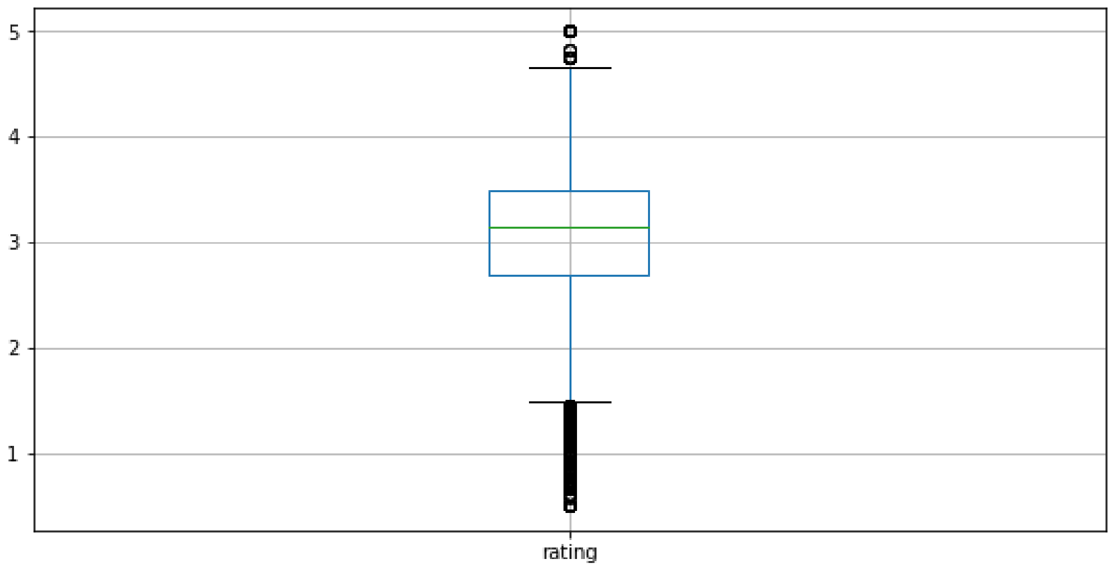


Fig : Visualize Rating in Box Plot

[9]:

```
# extract the launching year of each movies and make a new columns named year
movies['year'] = movies['title'].str.extract('.*\((.*)\).*', expand=True)
movies.tail()
```

[9]:

```
       movieId                          title                genres  year
62418   209157                      We (2018)                 Drama  2018
62419   209159         Window of the Soul (2001)           Documentary  2001
62420   209163                Bad Poems (2018)          Comedy|Drama  2018
62421   209169             A Girl Thing (2001)    (no genres listed)  2001
62422   209171  Women of Devil's Island (1962)  Action|Adventure|Drama  1962
```

Merge the previous avg_ratings and movies data set

[10]:

```
#  merging....
join_datasets = movies.merge(avg_ratings, on='movieId', how='inner')
join_datasets.tail()
```

[10]:

```
       movieId                          title                genres  year  \
59042   209157                      We (2018)                 Drama  2018
59043   209159         Window of the Soul (2001)           Documentary  2001
59044   209163                Bad Poems (2018)          Comedy|Drama  2018
59045   209169             A Girl Thing (2001)    (no genres listed)  2001
59046   209171  Women of Devil's Island (1962)  Action|Adventure|Drama  1962

       rating
59042     1.5
```

```
59043    3.0
59044    4.5
59045    3.0
59046    3.0
```

```
[11]: join_datasets.columns # coloumn in new data set
```

```
[11]: Index(['movieId', 'title', 'genres', 'year', 'rating'], dtype='object')
```

```
[12]: # get rid of the title column
      del join_datasets['title']
```

```
[13]: join_datasets.head()
```

```
[13]:    movieId                                        genres  year    rating
      0        1  Adventure|Animation|Children|Comedy|Fantasy  1995  3.893708
      1        2                   Adventure|Children|Fantasy  1995  3.251527
      2        3                               Comedy|Romance  1995  3.142028
      3        4                         Comedy|Drama|Romance  1995  2.853547
      4        5                                       Comedy  1995  3.058434
```

Data Cleaning: Handling Missing Data

```
[14]: # Find the shape of the join data set
      join_datasets.shape
```

```
[14]: (59047, 4)
```

```
[15]: # check is there any null value , if so , true boolean value will be return
      join_datasets.isnull().any()
```

```
[15]: movieId    False
      genres     False
      year        True
      rating     False
      dtype: bool
```

Hmm , there're some null value , we have to drop them out.

```
[16]: # dropna () is used to drop out the null values
      join_datasets = join_datasets.dropna()
```

```
[17]: # again check the shape of the data sets , these time row number decrease
      # indcating some rows are erased as they hold null values
      join_datasets.shape
```

```
[17]: (58678, 4)
```

```
[18]: # again check is their any null values
      join_datasets.isnull().any()
```

```
[18]: movieId     False
      genres      False
      year        False
      rating      False
      dtype: bool
```

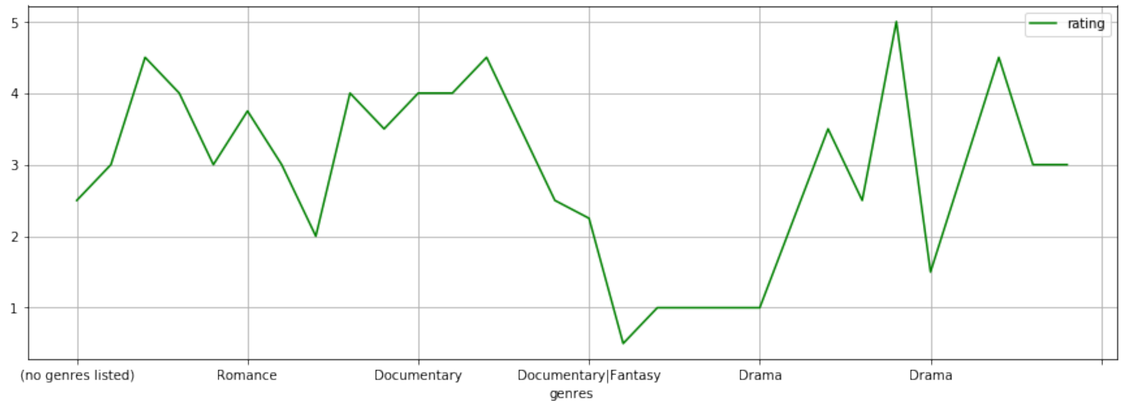
hmm , all null values are gone.It’s Ok now.

Data Visualization

0.1.1 Comparing Genres VS Ratings value , to see the correlation plot in following. We will use general ploting diagram to visualize it ,where genres is alone X axes and ratings is along Y axes.

```
[19]: join_datasets[-30:].plot(x='genres', y='rating', figsize=(15,5), grid=True ,␣
       ↪color ='g')
```

```
[19]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3cebb37a10>
```



`Fig : Ploting ratings VS genres . Drama genres tend to high than other movie genres`

Comments On Plot

Here we can see , ploting **genres** and **ratings** values shows us that Drama type movies tends to rate more high than other movies genres.Other genres has average ratinsgs scale though comedy genres is following Drama genres.

0.1.2 For making to visualize more convineint , let's use pie plot.

[20]:

```
# using value_counts() on our join_datasets , we can also see Drama movies are␣
 ↪majority in numbers>
gen_count = join_datasets['genres'].value_counts()
gen_count[:10]
```

[20]:

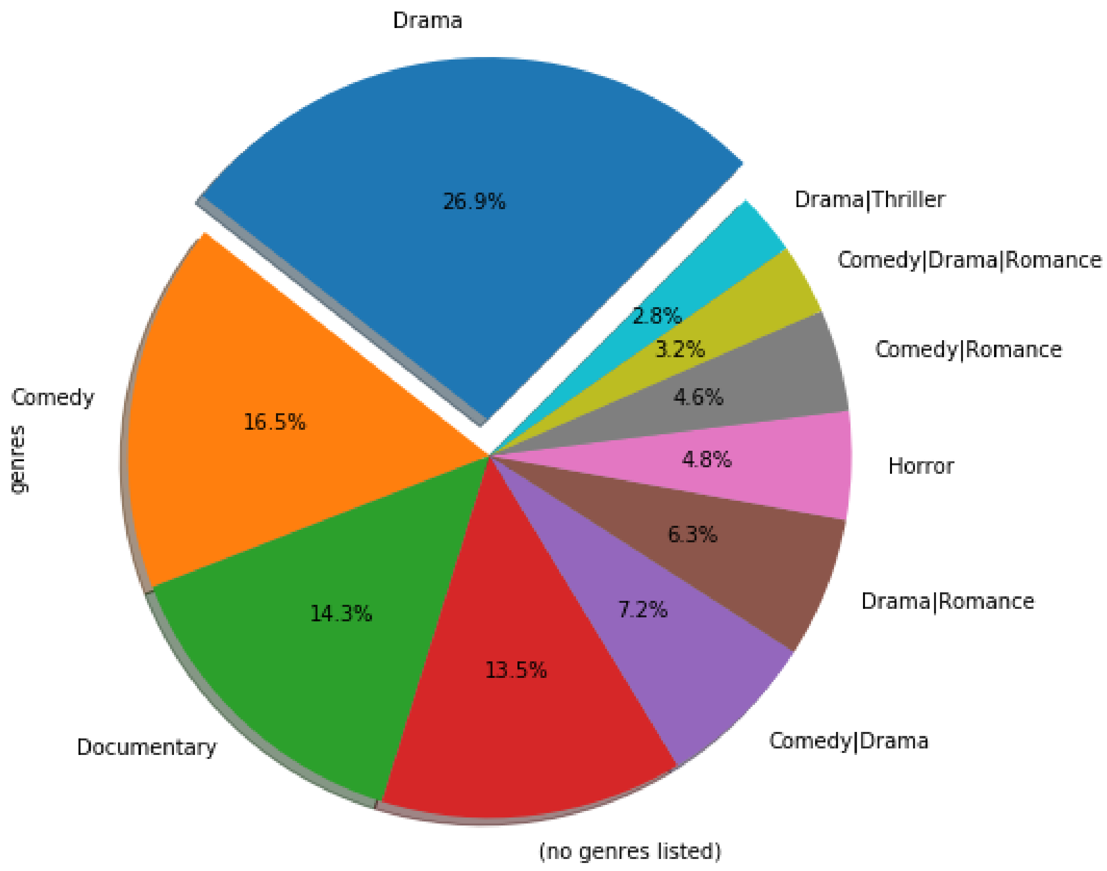
```
Drama                     8621
Comedy                    5283
Documentary               4571
(no genres listed)        4325
Comedy|Drama              2308
Drama|Romance             2004
Horror                    1549
Comedy|Romance            1460
Comedy|Drama|Romance      1014
Drama|Thriller             893
Name: genres, dtype: int64
```

[21]:

```
# plot the most frequent genres
gen_count[:10].plot(
                    kind = 'pie', figsize=(10,8) , shadow = True,
                    explode =(0.1,0,0,0,0,0,0,0,0,0),
                    autopct = '%1.1f%%' , startangle = 45
                   )
```

[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3ceba0c2d0>
```

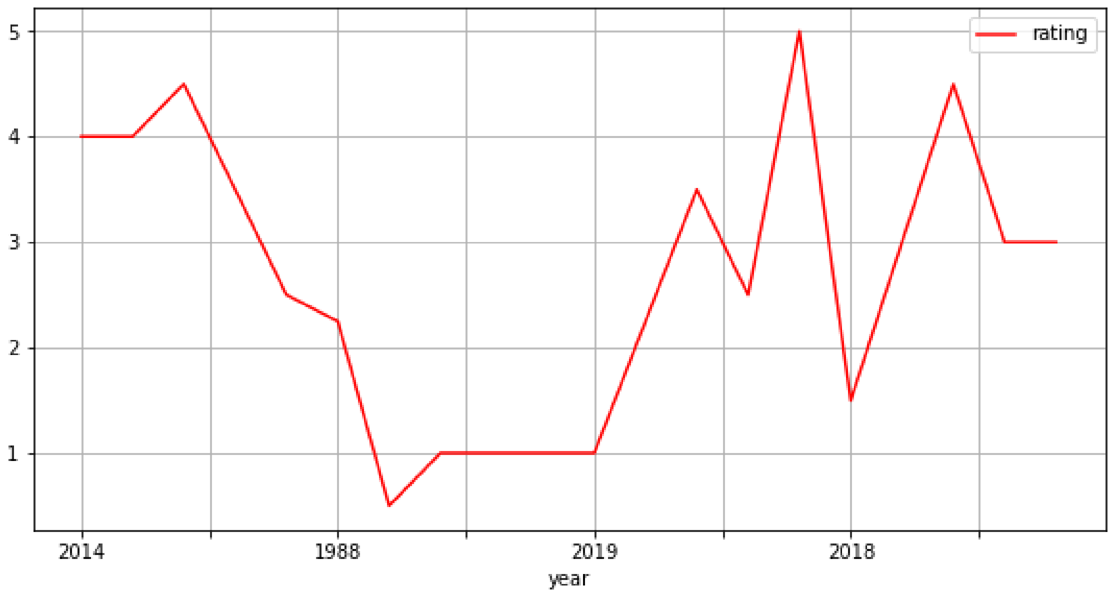


```
Fig : Ploting the most frequent genres , here which is Drama
```

0.1.3 We can also find is movie ratings are related of its concern launch year

```
[22]: # takings our whole data set
      join_datasets[-20:].plot(x='year', y='rating', figsize=(10,5), grid=True ,␣
       ↪color = 'r')
```

```
[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3ceb487e90>
```



```
[23]: # taking average of the year
      average_year = join_datasets[['year','rating']].groupby('year', as_index=False).
       ↪mean()
```

```
[24]: average_year[-20:].plot(x='year', y='rating', figsize=(10,5), grid=True ,␣
       ↪color='DarkBlue')
```

```
[24]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3ceb05c850>
```

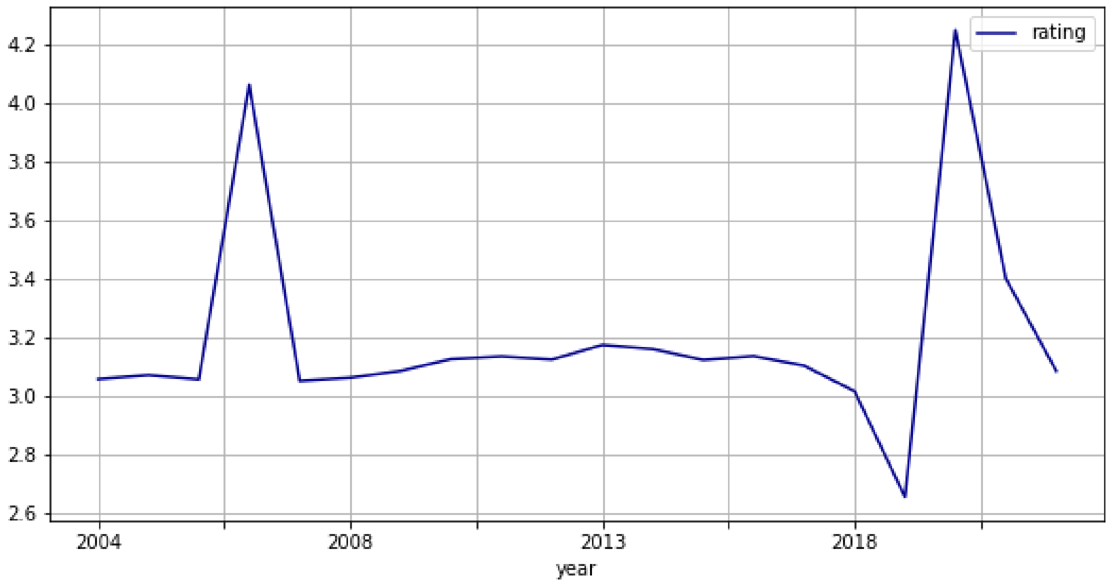


Fig : Average Movie Ratings over Time