IMDB MOVIE ANALYSIS

July 9, 2023

Importing pyton libararies pandas matplotlib and seaborn for analysis of imdb dataset

```
[2]: import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
```

Now reading the csv imdb dataset

```
[3]: dataim=pd.read_csv("Desktop/IMDB-Movie-Data.csv")
```

Now ensuring that the data is there or not and reading the head of data (10 head rows)

```
[4]: dataim.head(10)
```

```
[4]:   Rank                    Title                       Genre  \
     0     1  Guardians of the Galaxy     Action,Adventure,Sci-Fi
     1     2               Prometheus    Adventure,Mystery,Sci-Fi
     2     3                    Split             Horror,Thriller
     3     4                     Sing     Animation,Comedy,Family
     4     5            Suicide Squad    Action,Adventure,Fantasy
     5     6           The Great Wall    Action,Adventure,Fantasy
     6     7               La La Land          Comedy,Drama,Music
     7     8                 Mindhorn                      Comedy
     8     9       The Lost City of Z  Action,Adventure,Biography
     9    10               Passengers     Adventure,Drama,Romance

                                            Description              Director  \
     0  A group of intergalactic criminals are forced …            James Gunn
     1  Following clues to the origin of mankind, a te…          Ridley Scott
     2  Three girls are kidnapped by a man with a diag…    M. Night Shyamalan
     3  In a city of humanoid animals, a hustling thea…  Christophe Lourdelet
     4  A secret government agency recruits some of th…            David Ayer
     5  European mercenaries searching for black powde…           Yimou Zhang
     6  A jazz pianist falls for an aspiring actress i…       Damien Chazelle
     7  A has-been actor best known for playing the ti…            Sean Foley
     8  A true-life drama, centering on British explor…            James Gray
     9  A spacecraft traveling to a distant colony pla…         Morten Tyldum

                                                 Actors  Year  Runtime (Minutes)  \
     0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014                121
     1  Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012                124
     2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016                117
     3  Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016                108
     4  Will Smith, Jared Leto, Margot Robbie, Viola D…  2016                123
     5      Matt Damon, Tian Jing, Willem Dafoe, Andy Lau  2016                103
     6  Ryan Gosling, Emma Stone, Rosemarie DeWitt, J…  2016                128
     7  Essie Davis, Andrea Riseborough, Julian Barrat…  2016                 89
     8  Charlie Hunnam, Robert Pattinson, Sienna Mille…  2016                141
     9  Jennifer Lawrence, Chris Pratt, Michael Sheen,…  2016                116

        Rating   Votes  Revenue (Millions)  Metascore
     0     8.1  757074              333.13       76.0
     1     7.0  485820              126.46       65.0
     2     7.3  157606              138.12       62.0
     3     7.2   60545              270.32       59.0
     4     6.2  393727              325.02       40.0
     5     6.1   56036               45.13       42.0
     6     8.3  258682              151.06       93.0
     7     6.4    2490                 NaN       71.0
     8     7.1    7188                8.01       78.0
     9     7.0  192177              100.01       41.0
```

Now we should know the shape of the data rows and columns

[12]:
```python
print("total no of row in dataset is ",dataim.shape[0])
print("total no. of columns in the dataset is ",dataim.shape[1])
```

```
total no of row in dataset is  1000
total no. of columns in the dataset is  12
```

1 Converting Type Of Data

Converting data type of year to date time to have better analysis of data according to date at the time analysis

In the data we have year column which should be of datetime data type

[19]:
```python
dataim["Year"]=pd.to_datetime(dataim["Year"])
```

Info() will get the datatype of all column and memeory usage of data is 93.9+ KB

[20]:
```python
dataim.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Rank                1000 non-null   int64
 1   Title               1000 non-null   object
 2   Genre               1000 non-null   object
 3   Description         1000 non-null   object
 4   Director            1000 non-null   object
 5   Actors              1000 non-null   object
 6   Year                1000 non-null   datetime64[ns]
 7   Runtime (Minutes)   1000 non-null   int64
 8   Rating              1000 non-null   float64
 9   Votes               1000 non-null   int64
 10  Revenue (Millions)  872 non-null    float64
 11  Metascore           936 non-null    float64
dtypes: datetime64[ns](1), float64(3), int64(3), object(5)
memory usage: 93.9+ KB
```

2 Checking Missing Data

Now we are checking the missing details to dataset by this step we are going to know the quality of data we are getting and how easy is this data can be analyse or not

```
[4]: dataim.isna().sum()
```

```
[4]: Rank                  0
     Title                 0
     Genre                 0
     Description           0
     Director              0
     Actors                0
     Year                  0
     Runtime (Minutes)     0
     Rating                0
     Votes                 0
     Revenue (Millions)  128
     Metascore            64
     dtype: int64
```

```
[5]: dataim["Revenue (Millions)"]=dataim["Revenue (Millions)"].fillna(0)
```

```
[6]: dataim["Metascore"]=dataim["Metascore"].fillna(0)
```

```
[7]: dataim["Metascore"].isna().value_counts()
```

```
[7]: False    1000
     Name: Metascore, dtype: int64
```

```
[15]: dataim.isna().sum()
```

```
[15]: Rank                  0
      Title                 0
      Genre                 0
      Description           0
      Director              0
      Actors                0
      Year                  0
      Runtime (Minutes)     0
      Rating                0
      Votes                 0
      Revenue (Millions)    0
      Metascore             0
      dtype: int64
```

this upper list is showing that there is no NaN (means null value) in our data

3 Overall Statistics of data

```
[16]: dataim.describe()
```

```
[16]:             Rank         Year  Runtime (Minutes)       Rating         Votes  \
      count  1000.000000  1000.000000        1000.000000  1000.000000  1.000000e+03
      mean    500.500000  2012.783000         113.172000     6.723200  1.698083e+05
      std     288.819436     3.205962          18.810908     0.945429  1.887626e+05
      min       1.000000  2006.000000          66.000000     1.900000  6.100000e+01
      25%     250.750000  2010.000000         100.000000     6.200000  3.630900e+04
      50%     500.500000  2014.000000         111.000000     6.800000  1.107990e+05
      75%     750.250000  2016.000000         123.000000     7.400000  2.399098e+05
      max    1000.000000  2016.000000         191.000000     9.000000  1.791916e+06

             Revenue (Millions)    Metascore
      count         1000.000000  1000.000000
      mean            72.337960    55.210000
      std            100.320314    22.030598
      min              0.000000     0.000000
      25%              3.352500    43.000000
      50%             37.145000    58.000000
      75%             99.177500    71.000000
      max            936.630000   100.000000
```

Through this upper set of data will show us the maximum , minimum , total count , and average a lot more statistical data of our dataset

4 Displaying title of the movies having runtime>=180

[17]:

```
dataim.head()
```

[17]:

```
   Rank                    Title                     Genre  \
0     1  Guardians of the Galaxy   Action,Adventure,Sci-Fi
1     2               Prometheus  Adventure,Mystery,Sci-Fi
2     3                    Split           Horror,Thriller
3     4                     Sing   Animation,Comedy,Family
4     5            Suicide Squad  Action,Adventure,Fantasy

                                         Description              Director  \
0  A group of intergalactic criminals are forced …            James Gunn
1  Following clues to the origin of mankind, a te…          Ridley Scott
2  Three girls are kidnapped by a man with a diag…    M. Night Shyamalan
3  In a city of humanoid animals, a hustling thea…  Christophe Lourdelet
4  A secret government agency recruits some of th…            David Ayer

                                              Actors  Year  Runtime (Minutes)  \
0  Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014                121
1  Noomi Rapace, Logan Marshall-Green, Michael Fa…  2012                124
2  James McAvoy, Anya Taylor-Joy, Haley Lu Richar…  2016                117
3  Matthew McConaughey,Reese Witherspoon, Seth Ma…  2016                108
4  Will Smith, Jared Leto, Margot Robbie, Viola D…  2016                123

   Rating   Votes  Revenue (Millions)  Metascore
0     8.1  757074              333.13       76.0
1     7.0  485820              126.46       65.0
2     7.3  157606              138.12       62.0
3     7.2   60545              270.32       59.0
4     6.2  393727              325.02       40.0
```

[21]:

```
dataim[dataim["Runtime (Minutes)"]>=180].Title
```

[21]:

```
82     The Wolf of Wall Street
88           The Hateful Eight
311             La vie d'Adèle
828                 Grindhouse
965              Inland Empire
Name: Title, dtype: object
```

This upper list showing the 5 films which are having runtime more than 180 minutes

5 In which year there was the highest Average profit ?

I want to change the style of the graphs we want so using the style available I have used GGPLOT

[8]:

```
plt.style.use('ggplot')
```

[9]:

```
dataim.groupby("Year")["Revenue (Millions)"].mean().sort_values(ascending=False)
```

[9]:

```
Year
2012    107.973281
2009    103.769804
2010     99.827500
2008     97.177308
2011     86.221587
2013     84.249670
2006     82.374091
2014     81.606122
2007     81.249623
2015     69.717480
2016     37.749663
Name: Revenue (Millions), dtype: float64
```

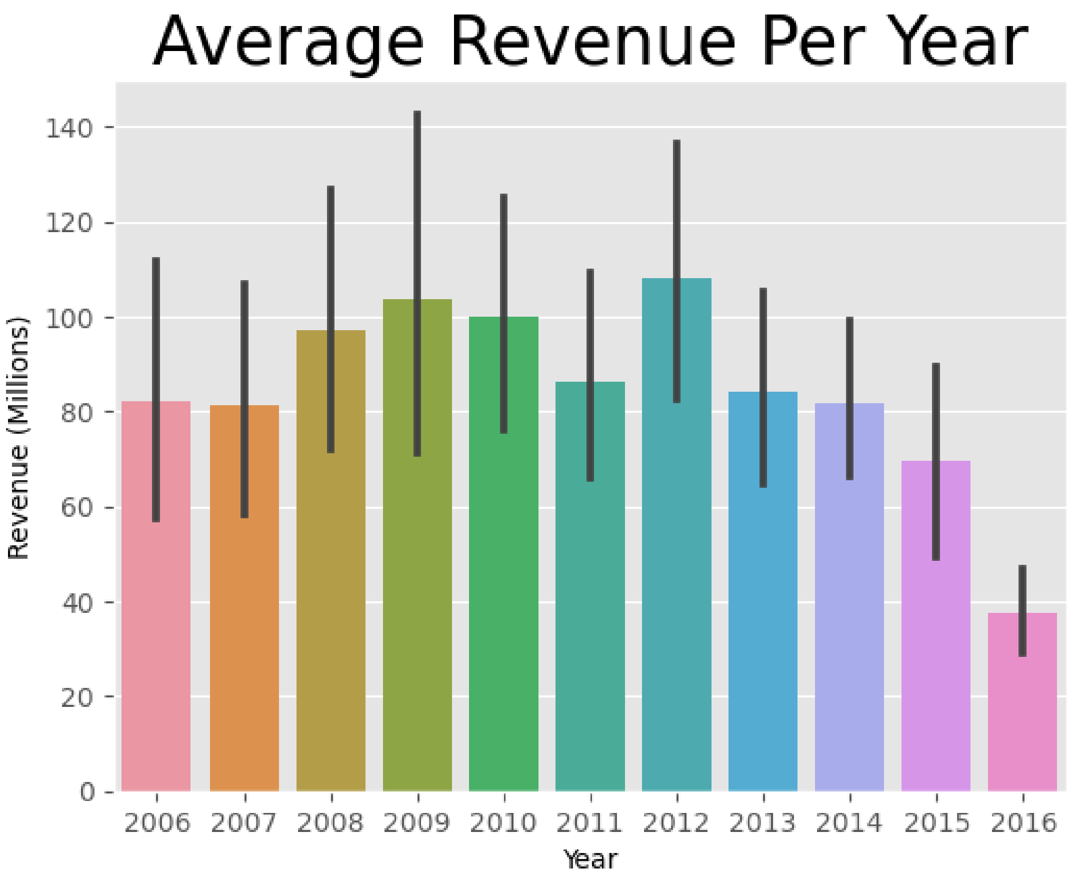
This series type of data is showing 2 columns to year and average of highest revenue of films per year

[86]:

```
plt.style.use('ggplot')
```

[10]:

```
sns.barplot(data=dataim,x="Year",y="Revenue (Millions)")
plt.title("Average Revenue Per Year",size=25)
plt.show()
```



This bar graph is showing that 2012 is the year in highest average revenue of the year

6 Average Rating Of Each Director

```
[111]: dataim.groupby(["Director"])["Rating"].mean().sort_values(ascending=False).
         ↪head(5)
```

```
[111]: Director
       Nitesh Tiwari        8.80
       Christopher Nolan    8.68
       Olivier Nakache      8.60
       Makoto Shinkai       8.60
       Aamir Khan           8.50
       Name: Rating, dtype: float64
```

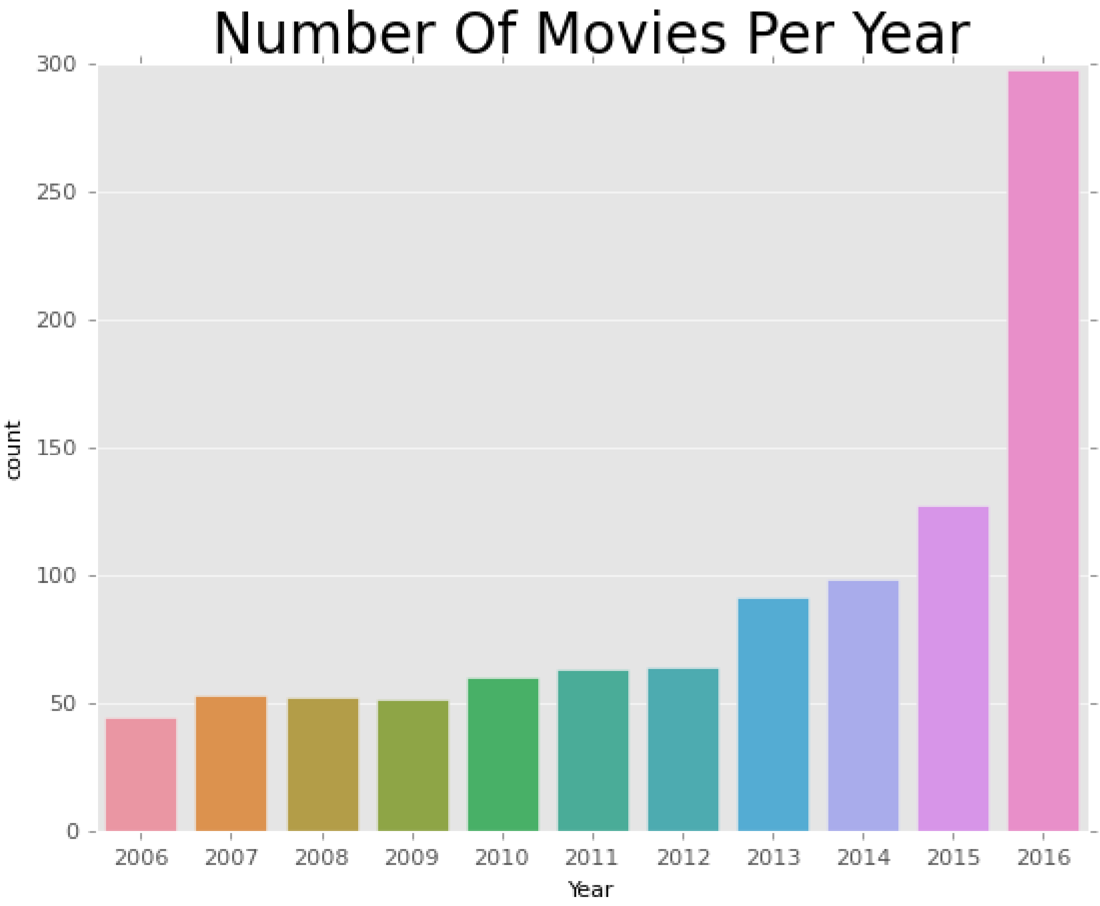
NITESH TIWARI is the Director who have the average rating with 8.80 in the dataset

7 Number of Movies Per Year

[118]:
```
dataim["Year"].value_counts()
```

[118]:
```
2016    297
2015    127
2014     98
2013     91
2012     64
2011     63
2010     60
2007     53
2008     52
2009     51
2006     44
Name: Year, dtype: int64
```

[126]:
```
sns.countplot(data=dataim,x="Year")
plt.title("Number Of Movies Per Year",size=25)
plt.show()
```



This bar graph is showing us that 2016 is the year in which we have the highest numbner of movies and the bar showing number of movies in all years .

8 Most Popular Movie (Highest Revenue)

```
[132]: dataim["Revenue (Millions)"].max()
```

```
[132]: 936.63
```

The max revenue of the movie is 936.63 million .

```
[135]: dataim[dataim["Revenue (Millions)"]==936.63]
```

```
[135]:     Rank                                          Title  \
       50    51  Star Wars: Episode VII - The Force Awakens

                               Genre  \
       50  Action,Adventure,Fantasy

                                               Description     Director  \
       50  Three decades after the defeat of the Galactic…  J.J. Abrams

                                                    Actors  Year  \
       50  Daisy Ridley, John Boyega, Oscar Isaac, Domhna…  2015

           Runtime (Minutes)  Rating   Votes  Revenue (Millions)  Metascore
       50                136     8.1  661608              936.63       81.0
```

Now the above set of data is the data of highest revenue which is 936.63 whose director is J.J. Abrams and title is Star Wars: Episode VII - The Force Awakens etc.

9 Top 10 Highest Rated Movie With Directors

```
[163]: fortop10=dataim[["Title","Director","Rating"]]
```

```
[164]: top10=d1.nlargest(10,"Rating")
       top10
```

```
[164]:                   Title            Director  Rating
       54       The Dark Knight  Christopher Nolan     9.0
       80             Inception  Christopher Nolan     8.8
       117               Dangal      Nitesh Tiwari     8.8
       36          Interstellar  Christopher Nolan     8.6
       96          Kimi no na wa     Makoto Shinkai     8.6
       249     The Intouchables    Olivier Nakache     8.6
       64           The Prestige  Christopher Nolan     8.5
```

```
99            The Departed    Martin Scorsese     8.5
124  The Dark Knight Rises  Christopher Nolan     8.5
133               Whiplash    Damien Chazelle     8.5
```

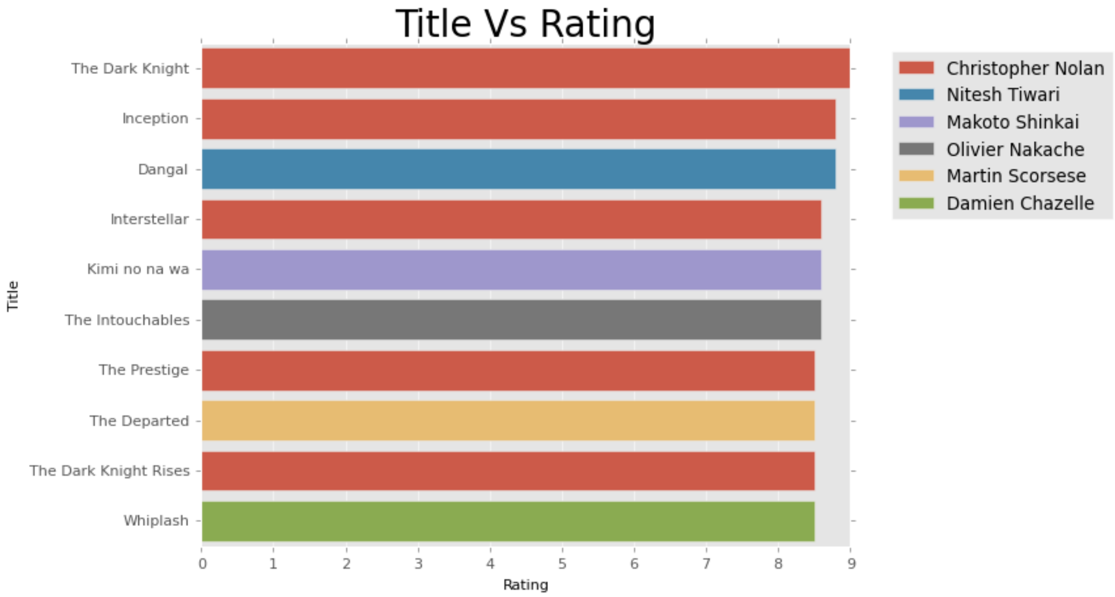
This list of data showing the top 10 highest rated movies data which include title , director , rating.

[181]:

```
sns.barplot(data=top10,y="Title",x="Rating",hue="Director",dodge=False)
plt.title("Title Vs Rating",size=25)
plt.legend(bbox_to_anchor=(1.05,1),loc=2)
```

[181]:

```
<matplotlib.legend.Legend at 0x1e069af7940>
```



10 Average Rating Of Movies Year Wise

[194]:

```
dataim.groupby(["Year"])["Rating"].mean().sort_values(ascending=False)
```

[194]:

```
Year
2007    7.133962
2006    7.125000
2009    6.960784
2012    6.925000
2011    6.838095
2014    6.837755
2010    6.826667
2013    6.812088
2008    6.784615
2015    6.602362
```

```
2016    6.436700
Name: Rating, dtype: float64
```

This series having year and average of rating per year. 2007 is the year in which average rating is highest of 7.133962 and 2016 is the year having least average of rating.

11 Classification of movie on Good Average and Excellent

Now we are using the function of classify which is having EXCELLENT , GOOD & AVERAGE category of films

[197]:

```
def classify(Rating):
    if Rating>7.0:
        return "EXCELLENT"
    elif Rating>6.0:
        return "GOOD"
    elif Rating>5.0:
        return "AVERAGE"
```

[201]:

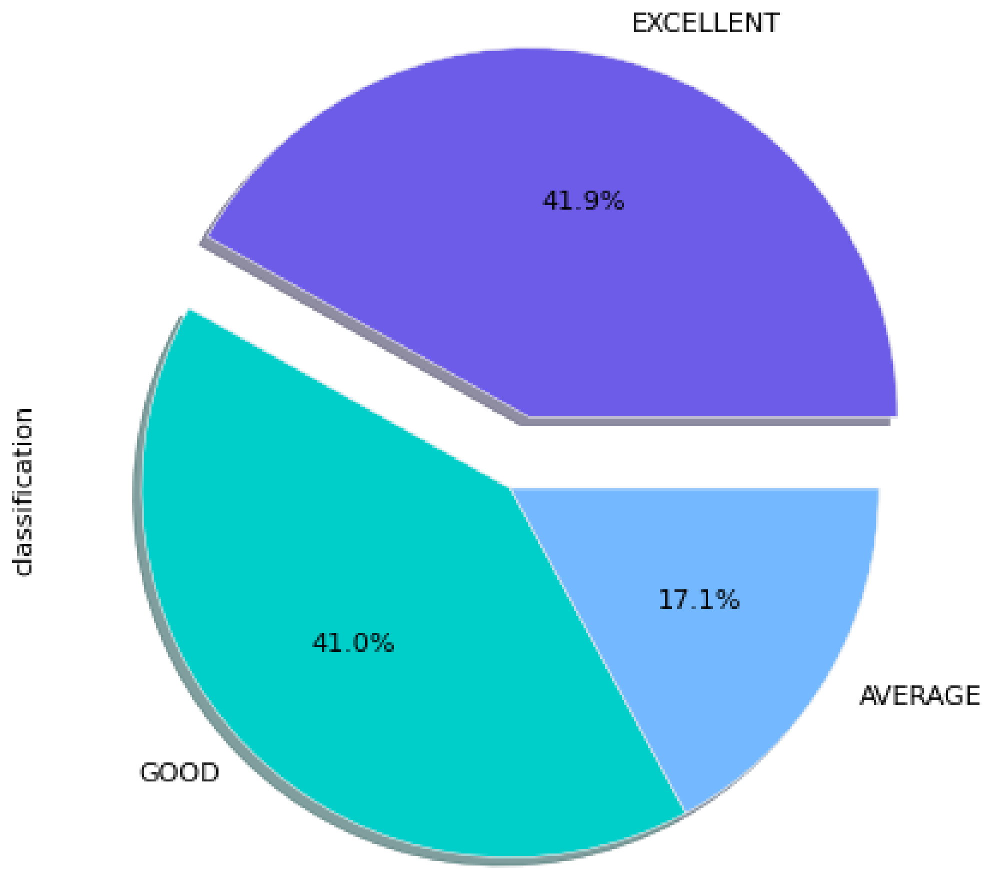
```
dataim["classification"]=dataim["Rating"].apply(classify)
```

[241]:

```
dataim["classification"].value_counts().plot(kind="pie", autopct="%1.
 ↪1f%%",colors=["#6c5ce7","#00cec9", "#74b9ff"],shadow=True,explode=(0.2,0,0))
```

[241]:

```
<Axes: ylabel='classification'>
```



This pie chart is showing that in our dataset excellent films are 41.9% , good movies are 41.0% and average movie is 17.1%

12 No. of action movies

Now we want the movies which are having action genre so we have to see that in the genre section we have see action genre in the column

```
[12]: dataim[dataim["Genre"].str.contains("Action",case=False)]
```

```
[12]:      Rank                           Title                        Genre  \
      0       1         Guardians of the Galaxy      Action,Adventure,Sci-Fi
      4       5                   Suicide Squad     Action,Adventure,Fantasy
      5       6                  The Great Wall     Action,Adventure,Fantasy
      8       9              The Lost City of Z   Action,Adventure,Biography
      12     13                       Rogue One      Action,Adventure,Sci-Fi
      ..    …                              …                         …
```

```
958   959                  3 Days to Kill       Action,Drama,Thriller
968   969                         Wrecker      Action,Horror,Thriller
969   970                 The Lone Ranger    Action,Adventure,Western
990   991  Underworld: Rise of the Lycans    Action,Adventure,Fantasy
993   994        Resident Evil: Afterlife     Action,Adventure,Horror

                                        Description            Director  \
0    A group of intergalactic criminals are forced …          James Gunn
4    A secret government agency recruits some of th…          David Ayer
5    European mercenaries searching for black powde…         Yimou Zhang
8    A true-life drama, centering on British explor…          James Gray
12   The Rebel Alliance makes a risky move to steal…      Gareth Edwards
..                                                …                   …
958  A dying CIA agent trying to reconnect with his…                 McG
968  Best friends Emily and Lesley go on a road tri…      Micheal Bafaro
969  Native American warrior Tonto recounts the unt…      Gore Verbinski
990  An origins story centered on the centuries-old…  Patrick Tatopoulos
993  While still out to destroy the evil Umbrella C…  Paul W.S. Anderson

                                             Actors  Year  \
0    Chris Pratt, Vin Diesel, Bradley Cooper, Zoe S…  2014
4    Will Smith, Jared Leto, Margot Robbie, Viola D…  2016
5        Matt Damon, Tian Jing, Willem Dafoe, Andy Lau  2016
8    Charlie Hunnam, Robert Pattinson, Sienna Mille…  2016
12   Felicity Jones, Diego Luna, Alan Tudyk, Donnie…  2016
..                                                …   …
958  Kevin Costner, Hailee Steinfeld, Connie Nielse…  2014
968  Anna Hutchison, Andrea Whitburn, Jennifer Koen…  2015
969  Johnny Depp, Armie Hammer, William Fichtner,To…  2013
990  Rhona Mitra, Michael Sheen, Bill Nighy, Steven…  2009
993  Milla Jovovich, Ali Larter, Wentworth Miller,K…  2010

     Runtime (Minutes)  Rating   Votes  Revenue (Millions)  Metascore
0                  121     8.1  757074              333.13       76.0
4                  123     6.2  393727              325.02       40.0
5                  103     6.1   56036               45.13       42.0
8                  141     7.1    7188                8.01       78.0
12                 133     7.9  323118              532.17       65.0
..                  …     …     …                 …        …
958                117     6.2   73567               30.69       40.0
968                 83     3.5    1210                0.00       37.0
969                150     6.5  190855               89.29        0.0
990                 92     6.6  129708               45.80       44.0
993                 97     5.9  140900               60.13       37.0

[303 rows x 12 columns]
```

```
[16]: dataim["Genre"].str.contains("Action",case=False).value_counts()
```

```
[16]: False    697
      True     303
      Name: Genre, dtype: int64
```

Now these true values are showing that 303 is the action movie count in our dataset

13 Classify the movies on the basis of Genre

```
[218]: split=dataim["Genre"].str.split(",",expand=True)
```

```
[220]: dataim["genre new"]=split[0]
```

```
[222]: dataim["genre new"].value_counts()
```

```
[222]: Action       293
       Drama        195
       Comedy       175
       Adventure     75
       Crime         71
       Biography     64
       Animation     49
       Horror        46
       Mystery       13
       Thriller      10
       Fantasy        4
       Sci-Fi         3
       Romance        2
       Name: genre new, dtype: int64
```

These are different genre we have in our dataset.

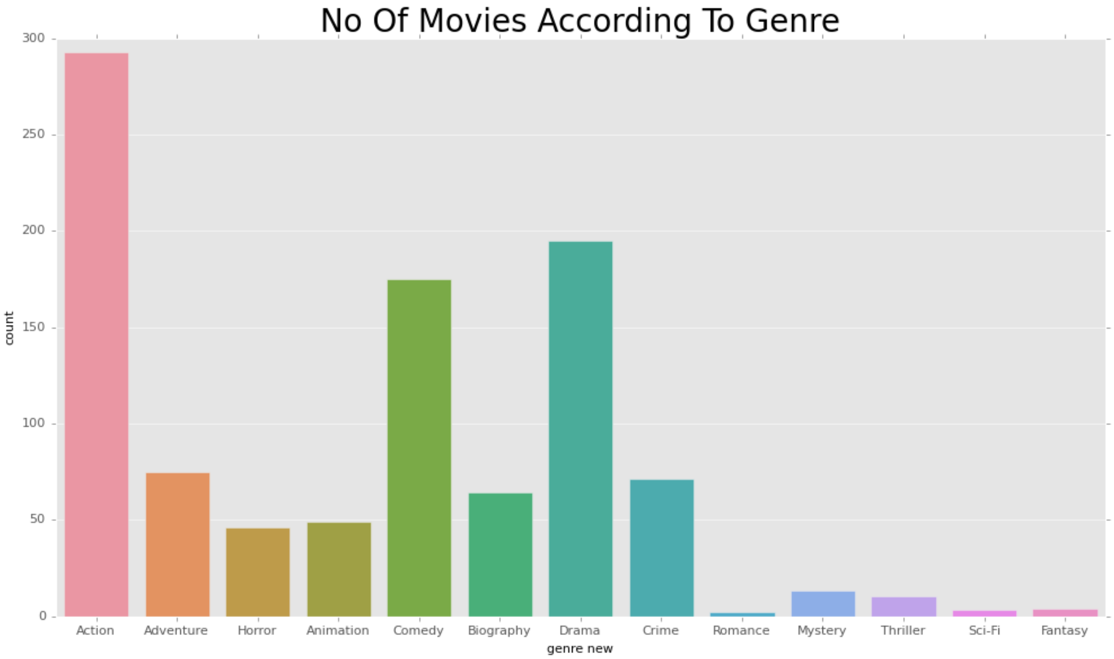
```
[248]: plt.figure(figsize=(10,20))
```

```
[248]: <Figure size 800x1600 with 0 Axes>

      <Figure size 800x1600 with 0 Axes>
```

```
[254]: plt.figure(figsize=(15,8))
       sns.countplot(data=dataim,x="genre new")
       plt.title("No Of Movies According To Genre",size=25)
```

```
[254]: Text(0.5, 1.0, 'No Of Movies According To Genre')
```



Now we can see the Action movies are having the number of 293 and this series is showing the other genre's number of movies too.

```
[ ]:
```