Internet Movie Database - Exploratory Data Analysis

The dataset consists of meta details about the movies such as director_name num_critic_for_reviews, duration,genres, actor_1_name, movie_title num_voted_users,language, country content_rating,budget,title_year,imdb_score etc. As the first step, let's load the dataset. In this kernel, I have analysed this dataset to find top insights and findings.

In [ ]:

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

In [ ]:

```
movie_df=pd.read_csv('/content/movie_metadata.csv')


#movie_df= pd.read_csv("/content/movie_metadata.csv",index_col=0)#Data Munging:convert .csv files to html

#movie_df.to_html('movie_df.html')
```

In [ ]:

```
#pd.read_html('/content/movie_df.html')
```

In [ ]:

```
from IPython.display import display
#show all columns
pd.options.display.max_columns = None
display(movie_df)
```

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	76050
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	30940
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20007
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	
...	...	...	...	...	...	...	...	...	
5038	Color	Scott Smith	1.0	87.0	2.0	318.0	Daphne Zuniga	637.0	
5039	Color	NaN	43.0	43.0	NaN	319.0	Valorie Curry	841.0	
5040	Color	Benjamin Roberds	13.0	76.0	0.0	0.0	Maxwell Moody	0.0	
5041	Color	Daniel Hsia	14.0	100.0	0.0	489.0	Daniel Henney	946.0	1
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	8

5043 rows × 28 columns

In [ ]:

```
movie_df.shape
```

Out[5]:

```
(5043, 28)
```

In [ ]:

```
print('The no.of rows are', movie_df.shape[0])
print('The no.of columns are', movie_df.shape[1])
```

```
The no.of rows are 5043
The no.of columns are 28
```

In [ ]:

```
movie_df.head(10)
```

Out[7]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gr
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	76050584
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	30940415
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20007417
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813064
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	N
5	Color	Andrew Stanton	462.0	132.0	475.0	530.0	Samantha Morton	640.0	7305867
6	Color	Sam Raimi	392.0	156.0	0.0	4000.0	James Franco	24000.0	33653030
7	Color	Nathan Greno	324.0	100.0	15.0	284.0	Donna Murphy	799.0	20080726
8	Color	Joss Whedon	635.0	141.0	0.0	19000.0	Robert Downey Jr.	26000.0	45899159
9	Color	David Yates	375.0	153.0	282.0	10000.0	Daniel Radcliffe	25000.0	30195698

In [ ]:

```
movie_df.tail(10)
```

Out[8]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	g
5033	Color	Shane Carruth	143.0	77.0	291.0	8.0	David Sullivan	291.0	4247
5034	Color	Neill Dela Llana	35.0	80.0	0.0	0.0	Edgar Tancangco	0.0	700
5035	Color	Robert Rodriguez	56.0	81.0	0.0	6.0	Peter Marquardt	121.0	20409
5036	Color	Anthony Vallone	NaN	84.0	2.0	2.0	John Considine	45.0	
5037	Color	Edward Burns	14.0	95.0	0.0	133.0	Caitlin FitzGerald	296.0	45
5038	Color	Scott Smith	1.0	87.0	2.0	318.0	Daphne Zuniga	637.0	
5039	Color	NaN	43.0	43.0	NaN	319.0	Valorie Curry	841.0	
5040	Color	Benjamin Roberds	13.0	76.0	0.0	0.0	Maxwell Moody	0.0	
5041	Color	Daniel Hsia	14.0	100.0	0.0	489.0	Daniel Henney	946.0	104
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	852

In [ ]:

```
movie_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5043 entries, 0 to 5042
Data columns (total 28 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   color                      5024 non-null   object 
 1   director_name              4939 non-null   object 
 2   num_critic_for_reviews     4993 non-null   float64
 3   duration                   5028 non-null   float64
 4   director_facebook_likes    4939 non-null   float64
 5   actor_3_facebook_likes     5020 non-null   float64
 6   actor_2_name               5030 non-null   object 
 7   actor_1_facebook_likes     5036 non-null   float64
 8   gross                      4159 non-null   float64
 9   genres                     5043 non-null   object 
 10  actor_1_name               5036 non-null   object 
 11  movie_title                5043 non-null   object 
 12  num_voted_users            5043 non-null   int64  
 13  cast_total_facebook_likes  5043 non-null   int64  
 14  actor_3_name               5020 non-null   object 
 15  facenumber_in_poster       5030 non-null   float64
 16  plot_keywords              4890 non-null   object 
 17  movie_imdb_link            5043 non-null   object 
 18  num_user_for_reviews       5022 non-null   float64
 19  language                   5031 non-null   object 
 20  country                    5038 non-null   object 
 21  content_rating             4740 non-null   object 
 22  budget                     4551 non-null   float64
 23  title_year                 4935 non-null   float64
 24  actor_2_facebook_likes     5030 non-null   float64
 25  imdb_score                 5043 non-null   float64
 26  aspect_ratio               4714 non-null   float64
 27  movie_facebook_likes       5043 non-null   int64  
dtypes: float64(13), int64(3), object(12)
memory usage: 1.1+ MB
```

In [ ]:

```
# gives statistical data
movie_df.describe()
```

Out[10]:

	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_1_facebook_likes	gross	num_voted_users	cast_
count	4993.000000	5028.000000	4939.000000	5020.000000	5036.000000	4.159000e+03	5.043000e+03	
mean	140.194272	107.201074	686.509212	645.009761	6560.047061	4.846841e+07	8.366816e+04	
std	121.601675	25.197441	2813.328607	1665.041728	15020.759120	6.845299e+07	1.384853e+05	
min	1.000000	7.000000	0.000000	0.000000	0.000000	1.620000e+02	5.000000e+00	
25%	50.000000	93.000000	7.000000	133.000000	614.000000	5.340988e+06	8.593500e+03	
50%	110.000000	103.000000	49.000000	371.500000	988.000000	2.551750e+07	3.435900e+04	
75%	195.000000	118.000000	194.500000	636.000000	11000.000000	6.230944e+07	9.630900e+04	
max	813.000000	511.000000	23000.000000	23000.000000	640000.000000	7.605058e+08	1.689764e+06	

In [ ]:

```
#gives statistical data
movie_df.describe(include='all')
```

Out[12]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes
count	5024	4939	4993.000000	5028.000000	4939.000000	5020.000000	5030	5036.000000
unique	2	2398	NaN	NaN	NaN	NaN	3032	NaN
top	Color	Steven Spielberg	NaN	NaN	NaN	NaN	Morgan Freeman	NaN
freq	4815	26	NaN	NaN	NaN	NaN	20	NaN
mean	NaN	NaN	140.194272	107.201074	686.509212	645.009761	NaN	6560.047061
std	NaN	NaN	121.601675	25.197441	2813.328607	1665.041728	NaN	15020.759120
min	NaN	NaN	1.000000	7.000000	0.000000	0.000000	NaN	0.000000
25%	NaN	NaN	50.000000	93.000000	7.000000	133.000000	NaN	614.000000
50%	NaN	NaN	110.000000	103.000000	49.000000	371.500000	NaN	988.000000
75%	NaN	NaN	195.000000	118.000000	194.500000	636.000000	NaN	11000.000000
max	NaN	NaN	813.000000	511.000000	23000.000000	23000.000000	NaN	640000.000000

In [ ]:

```
print('is any null value in movie_df? ', movie_df.isnull().values.any())
```

```
is any null value in movie_df?  True
```

In [ ]:

```
movie_df.isnull().sum()
```

Out[14]:

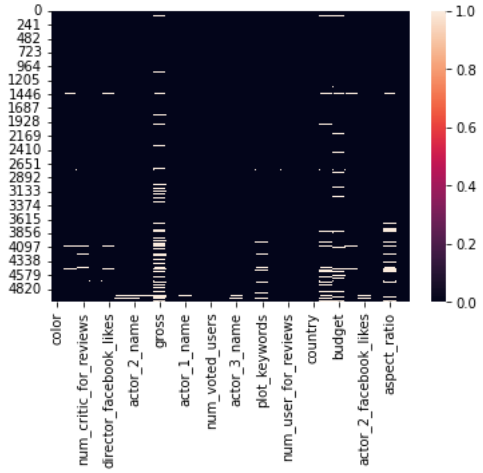
```
color                       19
director_name              104
num_critic_for_reviews      50
duration                    15
director_facebook_likes    104
actor_3_facebook_likes      23
actor_2_name                13
actor_1_facebook_likes       7
gross                      884
genres                       0
actor_1_name                 7
movie_title                  0
num_voted_users              0
cast_total_facebook_likes    0
actor_3_name                23
facenumber_in_poster        13
plot_keywords              153
movie_imdb_link              0
num_user_for_reviews        21
language                    12
country                      5
content_rating             303
budget                     492
title_year                 108
actor_2_facebook_likes      13
imdb_score                   0
aspect_ratio               329
movie_facebook_likes         0
dtype: int64
```

In [ ]:

```
sns.heatmap(movie_df.isnull())
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7faf88c77d30>
```



In [ ]:

```
miss_perc= movie_df.isnull().sum()*100/len(movie_df)       #gives percent of missing values
miss_perc
```

Out[16]:

```
color                      0.376760
director_name              2.062265
num_critic_for_reviews     0.991473
duration                   0.297442
director_facebook_likes    2.062265
actor_3_facebook_likes     0.456078
actor_2_name               0.257783
actor_1_facebook_likes     0.138806
gross                     17.529248
genres                     0.000000
actor_1_name               0.138806
movie_title                0.000000
num_voted_users            0.000000
cast_total_facebook_likes  0.000000
actor_3_name               0.456078
facenumber_in_poster       0.257783
plot_keywords              3.033908
movie_imdb_link            0.000000
num_user_for_reviews       0.416419
language                   0.237954
country                    0.099147
content_rating             6.008328
budget                     9.756098
title_year                 2.141582
actor_2_facebook_likes     0.257783
imdb_score                 0.000000
aspect_ratio               6.523895
movie_facebook_likes       0.000000
dtype: float64
```

In [ ]:

```
#REMOVE all missing values by default axis=0 (rows), inplace=True
movie_df.dropna(axis=0)
```

Out[18]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	76050
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	30940
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20007
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813
5	Color	Andrew Stanton	462.0	132.0	475.0	530.0	Samantha Morton	640.0	7305
...	...	...	...	...	...	...	...	...	
5026	Color	Olivier Assayas	81.0	110.0	107.0	45.0	Béatrice Dalle	576.0	13
5027	Color	Jafar Panahi	64.0	90.0	397.0	0.0	Nargess Mamizadeh	5.0	67
5033	Color	Shane Carruth	143.0	77.0	291.0	8.0	David Sullivan	291.0	42
5035	Color	Robert Rodriguez	56.0	81.0	0.0	6.0	Peter Marquardt	121.0	204
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	8

3756 rows × 28 columns

In [ ]:

```
dup_data= movie_df.duplicated().any()
dup_data
```

Out[19]:

```
True
```

In [ ]:

```
movie_df.drop_duplicates()
```

Out[20]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	76050
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	30940
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20007
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	
...	...	...	...	...	...	...	...	...	
5038	Color	Scott Smith	1.0	87.0	2.0	318.0	Daphne Zuniga	637.0	
5039	Color	NaN	43.0	43.0	NaN	319.0	Valorie Curry	841.0	
5040	Color	Benjamin Roberds	13.0	76.0	0.0	0.0	Maxwell Moody	0.0	
5041	Color	Daniel Hsia	14.0	100.0	0.0	489.0	Daniel Henney	946.0	1
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	8

4998 rows × 28 columns

In [ ]:

```
# movies with duration > 3hr
movie_df[movie_df[ 'duration']>=240]['movie_title']
```

Out[22]:

```
308         The Wolf of Wall Street
883               Gods and Generals
1144                  Heaven's Gate
1160                      Cleopatra
1501             Blood In, Blood Out
1571                  Apocalypse Now
1710              Trapped
1714    Once Upon a Time in America
1980                     Gettysburg
2088                         Gandhi
2466               Carlos
2561         Arn: The Knight Templar
2727         The Company
2970                        Das Boot
3311       The Legend of Suriyothai
3650                  Emma
Name: movie_title, dtype: object
```

In [ ]:

```
movie_df[movie_df['num_critic_for_reviews']>800]
```

Out[23]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gr
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813064

In [ ]:

```
movie_df[movie_df['num_voted_users']>1500000]['movie_title']
```

Out[24]:

```
66                The Dark Knight
1937    The Shawshank Redemption
Name: movie_title, dtype: object
```

In [ ]:

```
movie_df.groupby('director_name')['num_voted_users'].mean()[0:10].sort_values(ascending=False)
```

Out[25]:

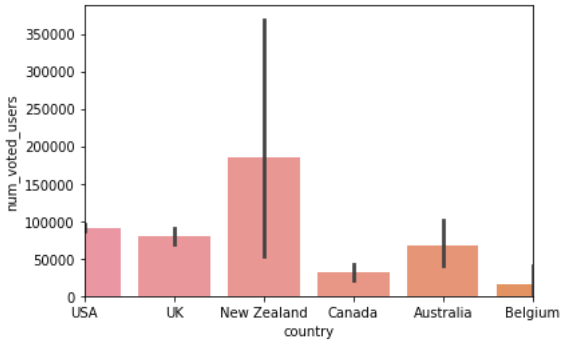
```
director_name
Adam Brooks         127760.0
Aaron Seltzer        50415.0
Adam Green           23349.0
Aaron Schneider      19147.0
Aaron Hann           13279.0
Adam Jay Epstein      9560.0
Abel Ferrara          6921.0
Adam Goldberg         1618.0
Adam Carolla          1351.0
A. Raven Cruz          534.0
Name: num_voted_users, dtype: float64
```

In [ ]:

```
sns.barplot(x='country', y='num_voted_users', data=movie_df)
plt.xlim(0,5)
```

Out[26]:

```
(0.0, 5.0)
```

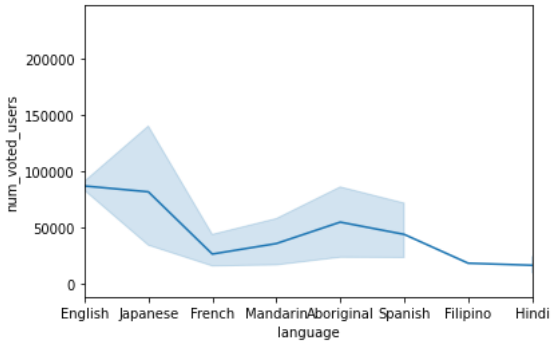


In [ ]:

```
sns.lineplot(x='language', y='num_voted_users', data=movie_df)
plt.xlim(0,7)
```

Out[27]:

```
(0.0, 7.0)
```

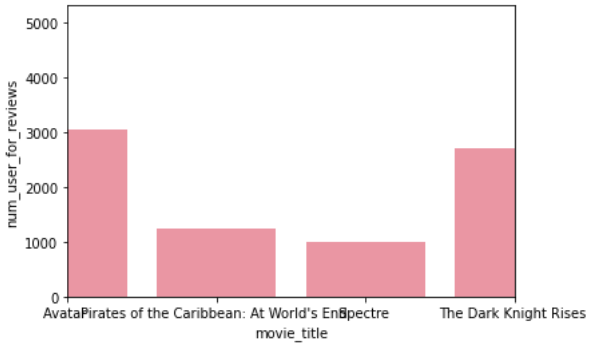


In [ ]:

```
sns.barplot(x='movie_title' , y='num_user_for_reviews', data=movie_df)
plt.xlim(0,3)
```

Out[40]:

```
(0.0, 3.0)
```



In [ ]:

```
movie_df.groupby('director_name')['imdb_score'].mean().head().sort_values(ascending=False)
```

Out[33]:

```
director_name
Aaron Schneider    7.1
Abel Ferrara       6.6
Aaron Hann         6.0
Aaron Seltzer      2.7
A. Raven Cruz      1.9
Name: imdb_score, dtype: float64
```

In [ ]:

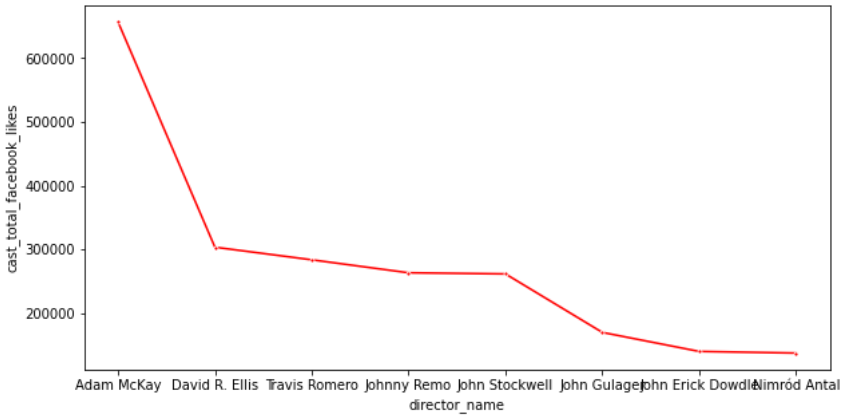
```
#Pandas nlargest() method** is used to get n largest values from a data frame or a series.
#DataFrame.nlargest(n, columns_name)[[col1,col2,...]]
DF=movie_df.nlargest(8,'cast_total_facebook_likes')[['cast_total_facebook_likes','director_name']]
```

In [ ]:

```
plt.figure(figsize=(10,5))
sns.lineplot(x='director_name', y='cast_total_facebook_likes', data=DF, color='r', marker ='.')
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7faf77498970>
```



In [ ]:

```
movie_df['title_year'].value_counts()
```

Out[43]:

```
2009.0    260
2014.0    252
2006.0    239
2013.0    237
2010.0    230
         ...
1932.0      1
1916.0      1
1934.0      1
1925.0      1
1920.0      1
Name: title_year, Length: 91, dtype: int64
```

In [ ]:

```
movie_df['title_year'].value_counts().shape
```

Out[44]:

```
(91,)
```

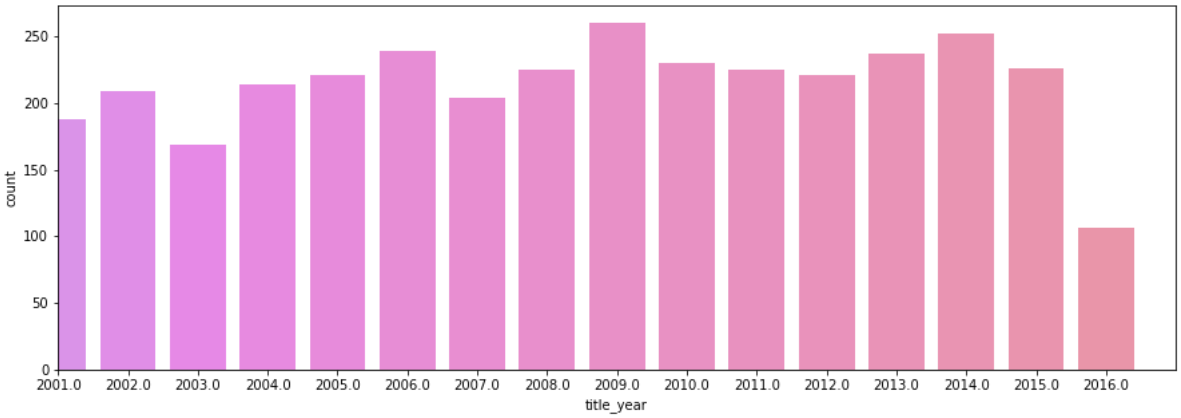
In [ ]:

```
plt.figure(figsize=(15,5))

sns.countplot(x='title_year', data=movie_df)
plt.xlim(75,91)
```

Out[45]:

```
(75.0, 91.0)
```



In [ ]:

```
movie_df[movie_df['budget'].max()==movie_df['budget']][['movie_title','budget' ]]   #[['movie_title','budget' ]] gives only movie_title,bu
```

Out[46]:

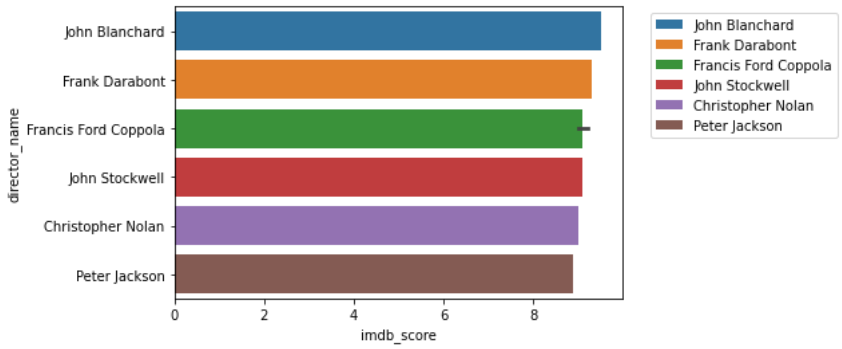
	movie_title	budget
2988	The Host	1.221550e+10

In [ ]:

```
DF_1=movie_df.nlargest(10,'imdb_score')[['director_name','imdb_score']]
```

In [ ]:

```
sns.barplot(x='imdb_score', y='director_name', data=DF_1, hue='director_name', dodge=False)
plt.legend(bbox_to_anchor=(1.05,1), loc=2)
plt.show()
```



In [ ]:

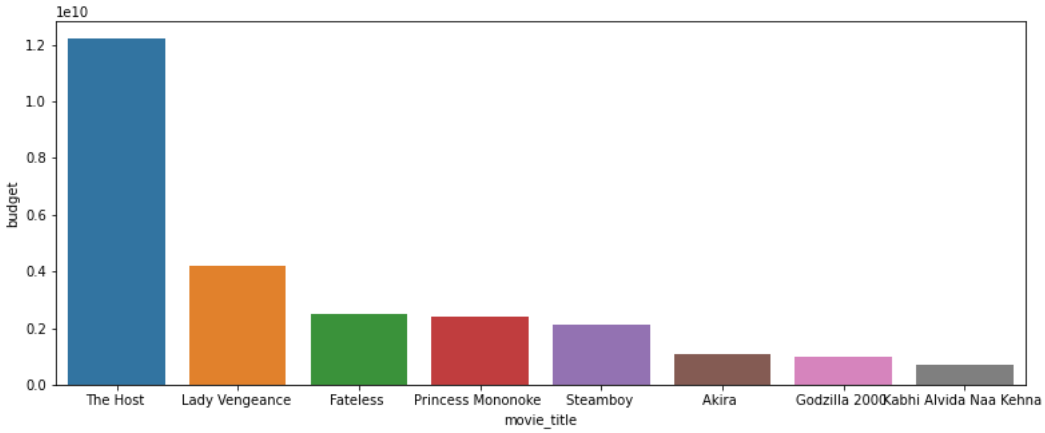
```
DF_2=movie_df.nlargest(8,'budget')[['movie_title','budget']]
```

In [ ]:

```
plt.figure(figsize=(13,5))
sns.barplot(x='movie_title', y='budget', data=DF_2)
```

Out[50]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7faf77458f70>
```



In [ ]:

```
DF_3=movie_df.nlargest(200,'budget')[['budget','imdb_score']]
```

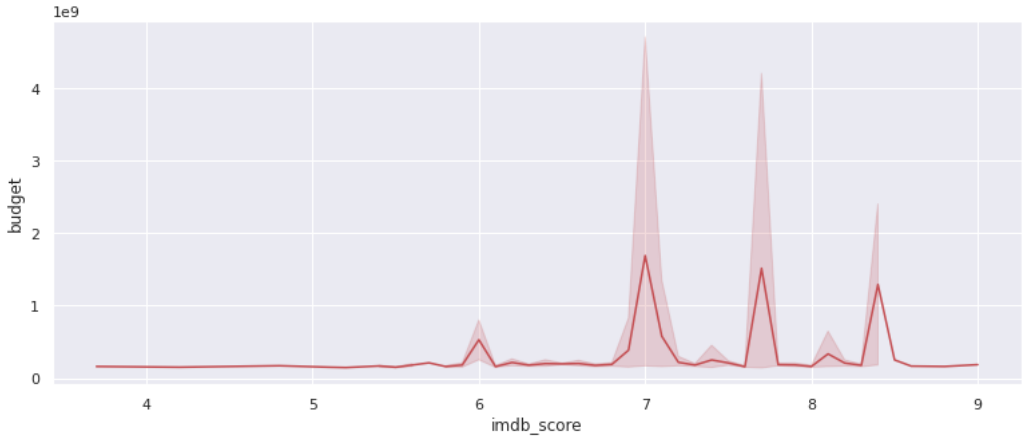
In [ ]:

```
plt.figure(figsize=(13,5))
sns.set_theme(style="darkgrid")

sns.lineplot(y='budget', x='imdb_score', data=DF_3,color='r')
```

Out[52]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7faf77489ee0>
```



In [ ]:

```
def Rating(imdb_score):
  if imdb_score >= 7.0:
    return 'Excellent'
  elif imdb_score >=5:
    return "medium"
  else:
    return 'low'
```

In [ ]:

```
movie_df['catagory']=movie_df['imdb_score'].apply(Rating)
```

In [ ]:

```
movie_df.head(10)
```

Out[55]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gr
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	76050584
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	30940415
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20007417
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813064
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	N
5	Color	Andrew Stanton	462.0	132.0	475.0	530.0	Samantha Morton	640.0	7305867
6	Color	Sam Raimi	392.0	156.0	0.0	4000.0	James Franco	24000.0	33653030
7	Color	Nathan Greno	324.0	100.0	15.0	284.0	Donna Murphy	799.0	20080726
8	Color	Joss Whedon	635.0	141.0	0.0	19000.0	Robert Downey Jr.	26000.0	45899159
9	Color	David Yates	375.0	153.0	282.0	10000.0	Daniel Radcliffe	25000.0	30195698

In [ ]:

```
#movie_df.drop('cat', axis=1)
```

In [ ]:

```
len(movie_df[movie_df['genres'].str.contains('action', case=False)])
```

Out[58]:

1153

In [ ]:

```
len(movie_df[movie_df['genres'].str.contains('comedy', case=False)])
```

Out[59]:

1872

In [ ]:

```
len(movie_df[movie_df['genres'].str.contains('Documentary', case=False)])
```

Out[60]:

121

In [ ]:

```
len(movie_df[movie_df['genres'].str.contains('crime', case=False)])
```

Out[61]:

889

In [ ]:

```
len(movie_df[movie_df['genres'].str.contains('Drama', case=False)])
```

Out[62]:

2594

In [ ]:

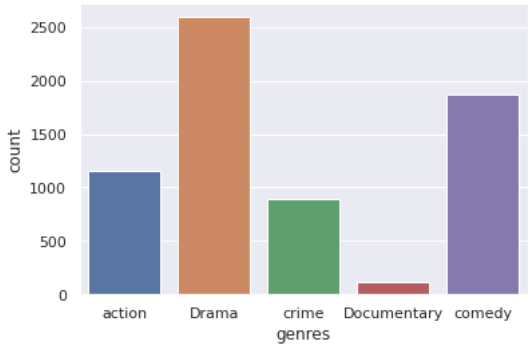
```
movie_type=pd.DataFrame({'genres':['action','Drama','crime','Documentary','comedy'],
                         'count': [1153, 2594, 889,121,1872]})
movie_type
```

Out[63]:

	genres	count
0	action	1153
1	Drama	2594
2	crime	889
3	Documentary	121
4	comedy	1872

In [ ]:

```
sns.barplot(x='genres', y='count', data=movie_type)
sns.set_theme(style='darkgrid')
plt.show()
```



In [ ]:

```
movie_type['genres'].unique()
```

Out[65]:

```
array(['action', 'Drama', 'crime', 'Documentary', 'comedy'], dtype=object)
```

In [ ]:

```
def custom_rating(genres,imdb_score):
  if 'Documentary' in 'genres':
    return min(10,imdb_score+1)
  elif 'Drama':
    return max(0,imdb_score-1)
  else:
    return 'imdb_score'
```

In [ ]:

```
movie_df['custom_rating']= movie_df.apply(lambda x:custom_rating(x['genres'], x['imdb_score']), axis=1 )
movie_df              #check last column
```

Out[67]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	76050
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	30940
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	20007
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	44813
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	
...	...	...	...	...	...	...	...	...	
5038	Color	Scott Smith	1.0	87.0	2.0	318.0	Daphne Zuniga	637.0	
5039	Color	NaN	43.0	43.0	NaN	319.0	Valorie Curry	841.0	
5040	Color	Benjamin Roberds	13.0	76.0	0.0	0.0	Maxwell Moody	0.0	
5041	Color	Daniel Hsia	14.0	100.0	0.0	489.0	Daniel Henney	946.0	1
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	8

5043 rows × 30 columns

In [ ]:

```
#movie_df[(movie_df['actor_1_name']=='Johnny Depp') & ( movie_df['budget']>=250000000.0)]
#movie_df[(movie_df['actor_1_name']=='Johnny Depp') & ( movie_df['imdb_score']>=8.0)]
movie_df[(movie_df['actor_1_name']=='salman khan') | ( movie_df['imdb_score']>=9.2)]   #or
```

Out[68]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	
1937	Color	Frank Darabont	199.0	142.0	0.0	461.0	Jeffrey DeMunn	11000.0	2834
2765	Color	John Blanchard	NaN	65.0	0.0	176.0	Andrea Martin	770.0	
3466	Color	Francis Ford Coppola	208.0	175.0	0.0	3000.0	Marlon Brando	14000.0	13482

Key Insights

At the end of our EDA on the IMDB,we have found out the following infereneces:

- John Blanchard is the director with movies having highest imdb ratings
- Higest number of movies are released on year 2009 between 2001 to 2016
- The Host is the blockblaster movie made with highest budget
- Highest number of drammatical movies are available on imdb following Comedy and documentry movies having lowest count
- Ratings have an effect on our target variable. The higher the rating, is the more budget movie.