IMDB Movie Analysis

By Santosh Shinde

Description

For your Final Project, we are providing you with dataset having various columns of different IMDB Movies. You are required to Frame the problem. For this task, you will need to define a problem you want to shed some light on.

We can do this by asking 'What?' This is where you frame the problem i.e. What is the problem?

Once you have framed the problem and gathered initial insights from the data, you can ask the following questions as you dig deeper into your analysis.

- What do you see happening?
- What are the specific symptoms of the problem?
- What is your hypothesis for the cause of the problem?

Approach

ASK

Nothing down key business questions asked by team.

PREPARE

IMDB_Movies data such as director name, actor name movie title, gross, budget, profit etc. is needed.

PROCESS

Data is collected in the format of csv file, then missing and duplicate data is cleaned.

ANALYZE

Data analysis using Python Jupyter to answer the business questions.

SHARE

Bringing data to life with visuals and sharing final report.

ACT

Deriving meaningful insights from analysis.

Tech-Stack Used

Data is in the format of .csv files from which the data will be processed into Jupyter Notebook

GOOGLE SHEETS

PYTHON JUPYTER

All the Business Questions will be answered in this platform.

GOOGLE SLIDES

All the analyzed data will be visualise in the form PPT.

Insights

1.Cleaning the Data : One of the most important step to perform before moving further into the analysis. Cleaning process such as dropping the columns, removing duplicates, removing null values etc.

>> The Dataset has 5043 Rows and 28 Columns.

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.0
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	NaN
...	...	...	...	...	...	...	...	...	...
5038	Color	Scott Smith	1.0	87.0	2.0	318.0	Daphne Zuniga	637.0	NaN
5039	Color	NaN	43.0	43.0	NaN	319.0	Valorie Curry	841.0	NaN
5040	Color	Benjamin Roberds	13.0	76.0	0.0	0.0	Maxwell Moody	0.0	NaN
5041	Color	Daniel Hsia	14.0	100.0	0.0	489.0	Daniel Henney	946.0	10443.0
5042	Color	Jon Gunn	43.0	90.0	16.0	16.0	Brian Herzlinger	86.0	85222.0

5043 rows × 28 columns

>> After Cleaning the Dataset, We now has 3767 Rows and 27 Columns.

>> Deleted ‘ IMDB_movie_link’ column name as it only contains links of the movies.

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross
0	Color	James Cameron	723	178	0	855	Joel David Moore	1000	760505847
1	Color	Gore Verbinski	302	169	563	1000	Orlando Bloom	40000	309404152
2	Color	Sam Mendes	602	148	0	161	Rory Kinnear	11000	200074175
3	Color	Christopher Nolan	813	164	22000	23000	Christian Bale	27000	448130642
5	Color	Andrew Stanton	462	132	475	530	Samantha Morton	640	73058679
...	...	...	...	...	...	...	...	...	...
5027	Color	Jafar Panahi	64	90	397	0	Nargess Mamizadeh	5	673780
5029	Color	Kiyoshi Kurosawa	78	111	62	6	Anna Nakagawa	89	94596
5033	Color	Shane Carruth	143	77	291	8	David Sullivan	291	424760
5035	Color	Robert Rodriguez	56	81	0	6	Peter Marquardt	121	2040920
5042	Color	Jon Gunn	43	90	16	16	Brian Herzlinger	86	85222

3767 rows × 27 columns

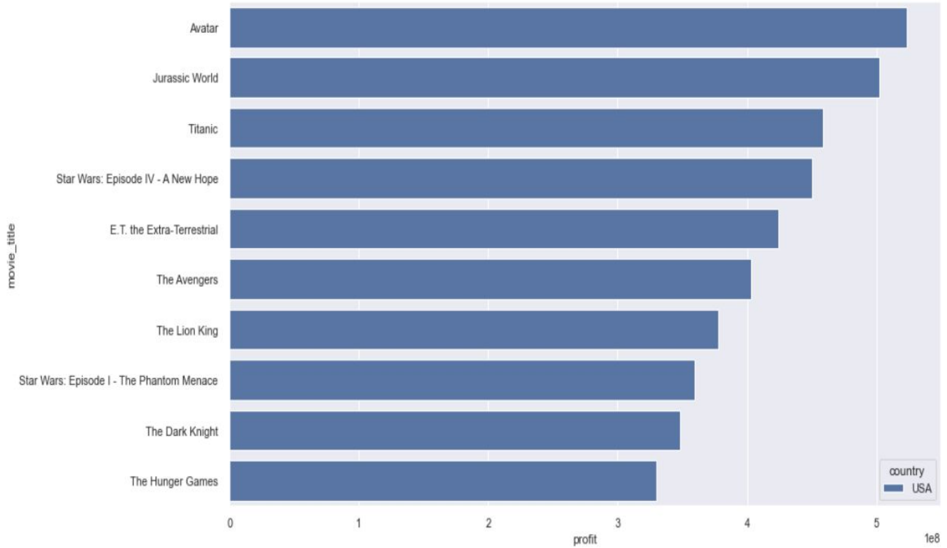
Insights

2.Movies with highest profits: Added a new column called profit that contains the difference of the gross and budget columns.
>>Plot profit (y-axis) vs budget (x- axis) and observe the outliers using the appropriate chart type.
>> Here is the top 10 highest profits of the movies.

```
#Retriving top 10 Highest proift movies
top_10 = Highest_profit.iloc[:10]
top_10[['movie_title','country','budget','profit']]
```

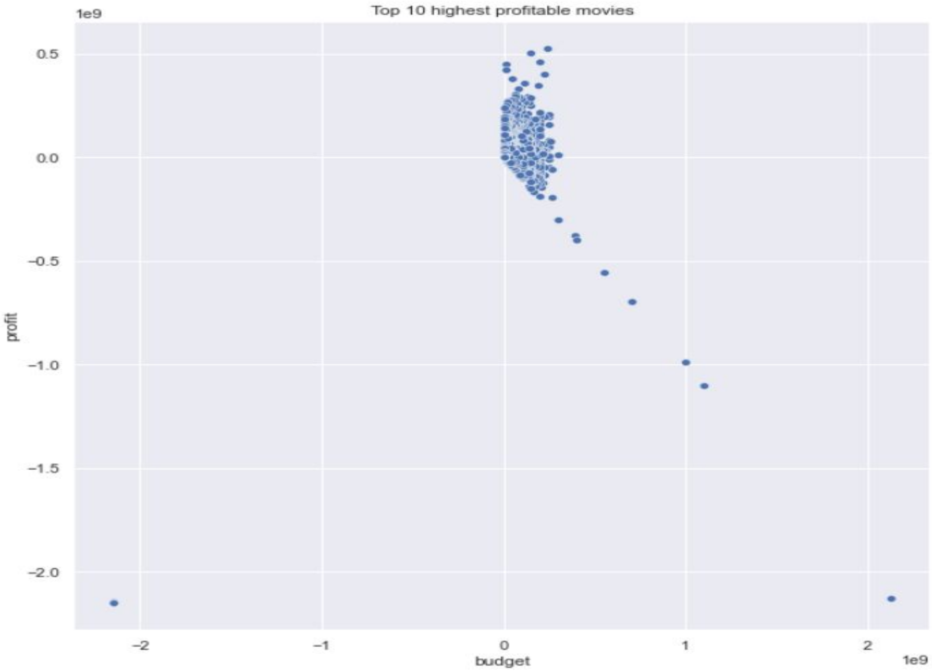
	movie_title	country	budget	profit
0	Avatar	USA	237000000	523505847
29	Jurassic World	USA	150000000	502177271
26	Titanic	USA	200000000	458672302
3024	Star Wars: Episode IV - A New Hope	USA	11000000	449935665
3080	E.T. the Extra-Terrestrial	USA	10500000	424449459
17	The Avengers	USA	220000000	403279547
509	The Lion King	USA	45000000	377783777
240	Star Wars: Episode I - The Phantom Menace	USA	115000000	359544677
66	The Dark Knight	USA	185000000	348316061
439	The Hunger Games	USA	78000000	329999255

```
sns.set(style='darkgrid')
plt.figure(figsize=(14,8))
sns.barplot(data = top_10, y = top_10['movie_title'], x = top_10['profit'],hue = top_10['country'])
plt.show()
```



>> We can see 5 to 6 Outliers based on the ‘profit ’ column.

```
plt.figure(figsize = (10,10))
sns.scatterplot(data['budget'],data['profit'])
plt.title("Top 10 highest profitable movies")
plt.show()
```



Insights

3. Top 250: Create a new column IMDb_Top_250 and store the top 250 movies with the highest IMDb Rating(corresponding to the column: imdb_score). Also make sure that for all of these movies, the num_voted_users is greater than 25,000. Also add a Rank column containing the values 1 to 250 indicating the ranks of the corresponding films.

Extract all the movies in the IMDb_Top_250 column which are not in the English language and store them in a new column named Top_Foreign_Lang_Film.

>> These are the Top 250 movies

Rank	IMDb_Top_250	language	num_voted_users	imdb_score
1.0	The Shawshank Redemption	English	1689764	9.3
2.0	The Godfather	English	1155770	9.2
3.0	The Godfather: Part II	English	790926	9.0
4.0	The Dark Knight	English	1676169	9.0
5.0	The Good, the Bad and the Ugly	Italian	503509	8.9
6.0	The Lord of the Rings: The Return of the King	English	1215718	8.9
7.0	Pulp Fiction	English	1324680	8.9
8.0	Schindler's List	English	865020	8.9
9.0	Forrest Gump	English	1251222	8.8
10.0	Inception	English	1468200	8.8
238.0	The Untouchables	English	219008	7.9
239.0	Moon	English	260607	7.9
240.0	Taken	English	483756	7.9
241.0	The Right Stuff	English	45271	7.9
242.0	The Fighter	English	275869	7.9
243.0	Straight Outta Compton	English	119928	7.9
244.0	Walk the Line	English	188637	7.9
245.0	Glory	English	101888	7.9
246.0	The Notebook	English	396396	7.9
247.0	Before Midnight	English	95362	7.9
248.0	Hero	Mandarin	149414	7.9
249.0	The Remains of the Day	English	45703	7.9
250.0	Avatar	English	886204	7.9

Insights

3.1. Extract all the movies in the IMDb_Top_250 column which are not in the English language and store them in a new column named Top_Foreign_Lang_Film.

>> Here are the Top 250 movies which are not in English language. We only have 91 movies out of 250 Rows.

	IMDb_Top_250	language	num_voted_users	imdb_score
Rank				
1.0	The Good, the Bad and the Ugly	Italian	503509	8.9
2.0	Seven Samurai	Japanese	229012	8.7
3.0	City of God	Portuguese	533200	8.7
4.0	Spirited Away	Japanese	417971	8.6
5.0	Children of Heaven	Persian	27882	8.5
6.0	The Lives of Others	German	259379	8.5
7.0	Princess Mononoke	Japanese	221552	8.4
8.0	Das Boot	German	168203	8.4
9.0	Baahubali: The Beginning	Telugu	62756	8.4
10.0	Oldboy	Korean	356181	8.4
79.0	The Host	Korean	68883	7.0
80.0	El Mariachi	Spanish	52055	6.9
81.0	Jab Tak Hai Jaan	Hindi	42296	6.9
82.0	High Tension	French	55040	6.8
83.0	Coco Before Chanel	French	32003	6.7
84.0	Rumble in the Bronx	Cantonese	29843	6.7
85.0	[Rec] 2	Spanish	55597	6.6
86.0	Wasabi	French	29392	6.6
87.0	Night Watch	Russian	47097	6.5
88.0	The Interpreter	Aboriginal	86152	6.4
89.0	Dead Snow	Norwegian	54601	6.4
90.0	The Legend of Zorro	Spanish	71574	5.9
91.0	In the Land of Blood and Honey	Bosnian	31414	4.3

Insights

4.Best Directors: Group the column using the director_name column. Find out the top 10 directors for whom the mean of imdb_score is the highest and store them in a new column top 10 director. In case of a tie in IMDb score between two directors, sort them alphabetically.

>> So according to the findings 'Charles Chaplin' has the highest IMDB score of 8.6 and S.S Rajamouli is lowest in the top 10 directors who has 8.4 IMDB score.

```
# Write your code for extracting the top 10 directors here
best=data.groupby('director_name')

top10director=pd.DataFrame(best['imdb_score'].mean().sort_values(ascending=False))
top10director=top10director.head(10)

top10director=top10director.sort_values(['imdb_score','director_name'],ascending=(False,True))
top10director
```

director_name	imdb_score
Charles Chaplin	8.600000
Tony Kaye	8.600000
Alfred Hitchcock	8.500000
Damien Chazelle	8.500000
Majid Majidi	8.500000
Ron Fricke	8.500000
Sergio Leone	8.433333
Christopher Nolan	8.425000
Asghar Farhadi	8.400000
S.S. Rajamouli	8.400000

Insights

5.Popular Genres: Perform this step using the knowledge gained while performing previous steps.
>> Most of the people like genres such as 'Adventure|Animation|Drama|Family|Musical' which as imdb score of 8.50.

```
popular=data.groupby('genres')

pop=pd.DataFrame(popular['imdb_score'].mean().sort_values(ascending=False))
pop=pop.head(10)

pop=pop.sort_values(['imdb_score','genres'],ascending=(False,True))
pop
```

	imdb_score
genres	
Adventure|Animation|Drama|Family|Musical	8.50
Crime|Drama|Fantasy|Mystery	8.50
Action|Adventure|Drama|Fantasy|War	8.40
Adventure|Animation|Fantasy	8.40
Adventure|Drama|Thriller|War	8.40
Adventure|Animation|Comedy|Drama|Family|Fantasy	8.30
Biography|Drama|History|Music	8.30
Documentary|Drama|Sport	8.30
Documentary|War	8.30
Adventure|Drama|War	8.25

Insights

6.Charts :

- 1. Create three new DataFrame namely, `Meryl_Streep`, `Leo_Caprio`, and `Brad_Pitt` which contain the movies in which the actors: 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' are the lead actors. Use only the `actor_1_name` column for extraction. Also, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.

- 2. Append the rows of all these DataFrame and store them in a new dataframe named `Combined`.

- 3. Group the combined dataframe using the `actor_1_name` column.

- 4. Find the mean of the `num_critic_for_reviews` and `num_users_for_review` and identify the actors which have the highest mean.

- 5. Observe the change in number of voted users over decades using a bar chart. Create a column called `decade` which represents the decade to which every movie belongs to. For example, the `title_year` year 1923, 1925 should be stored as 1920s. Sort the DataFrame based on the column `decade`, group it by `decade` and find the sum of users voted in each decade. Store this in a new data frame called `df_by_decade`.

1. Create three new DataFrame namely, `Meryl_Streep`, `Leo_Caprio`, and `Brad_Pitt` which contain the movies in which the actors: 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' are the lead actors. Use only the `actor_1_name` column for extraction. Also, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.

>> These are the FILMS in which these three Actors were in lead role.

```
# Write your code for creating three new dataframes here
Meryl_Streep=data[['actor_1_name','movie_title','num_critic_for_reviews','num_user_for_reviews']]
Leo_Caprio=data[['actor_1_name','movie_title','num_critic_for_reviews','num_user_for_reviews']]
Brad_Pitt=data[['actor_1_name','movie_title','num_critic_for_reviews','num_user_for_reviews']]

# Include all movies in which Meryl_Streep is the lead
Meryl_Streep=Meryl_Streep.loc[Meryl_Streep['actor_1_name']=='Meryl Streep']
Meryl_Streep
```

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
410	Meryl Streep	It's Complicated	187	214
1106	Meryl Streep	The River Wild	42	69
1204	Meryl Streep	Julie & Julia	252	277
1408	Meryl Streep	The Devil Wears Prada	208	631
1483	Meryl Streep	Lions for Lambs	227	298
1575	Meryl Streep	Out of Africa	66	200
1618	Meryl Streep	Hope Springs	234	178
1674	Meryl Streep	One True Thing	64	112
1925	Meryl Streep	The Hours	174	660
2781	Meryl Streep	The Iron Lady	331	350
3135	Meryl Streep	A Prairie Home Companion	211	280

```
# Include all movies in which Leo_Caprio is the lead
Leo_Caprio=Leo_Caprio.loc[Leo_Caprio['actor_1_name']=='Leonardo DiCaprio']
Leo_Caprio
```

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
26	Leonardo DiCaprio	Titanic	315	2528
50	Leonardo DiCaprio	The Great Gatsby	490	753
97	Leonardo DiCaprio	Inception	642	2803
179	Leonardo DiCaprio	The Revenant	556	1188
257	Leonardo DiCaprio	The Aviator	267	799
296	Leonardo DiCaprio	Django Unchained	765	1193
307	Leonardo DiCaprio	Blood Diamond	166	657
308	Leonardo DiCaprio	The Wolf of Wall Street	606	1138
326	Leonardo DiCaprio	Gangs of New York	233	1166
361	Leonardo DiCaprio	The Departed	352	2054
452	Leonardo DiCaprio	Shutter Island	490	964
641	Leonardo DiCaprio	Body of Lies	238	263
911	Leonardo DiCaprio	Catch Me If You Can	194	667
990	Leonardo DiCaprio	The Beach	118	548
1114	Leonardo DiCaprio	Revolutionary Road	323	414
1422	Leonardo DiCaprio	The Man in the Iron Mask	83	244
1453	Leonardo DiCaprio	J. Edgar	392	279
1560	Leonardo DiCaprio	The Quick and the Dead	63	216
2067	Leonardo DiCaprio	Marvin's Room	45	71
2757	Leonardo DiCaprio	Romeo + Juliet	106	506
3476	Leonardo DiCaprio	The Great Gatsby	490	753

```python
# Include all movies in which Brad pitt is the lead
Brad_Pitt=Brad_Pitt.loc[Brad_Pitt['actor_1_name']=='Brad Pitt']
Brad_Pitt
```

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
101	Brad Pitt	The Curious Case of Benjamin Button	362	822
147	Brad Pitt	Troy	220	1694
254	Brad Pitt	Ocean's Twelve	198	627
255	Brad Pitt	Mr. & Mrs. Smith	233	798
382	Brad Pitt	Spy Game	142	361
400	Brad Pitt	Ocean's Eleven	186	845
470	Brad Pitt	Fury	406	701
611	Brad Pitt	Seven Years in Tibet	76	119
683	Brad Pitt	Fight Club	315	2968
792	Brad Pitt	Sinbad: Legend of the Seven Seas	98	91
940	Brad Pitt	Interview with the Vampire: The Vampire Chroni...	120	406
1490	Brad Pitt	The Tree of Life	584	975
1722	Brad Pitt	The Assassination of Jesse James by the Coward...	273	415
2204	Brad Pitt	Babel	285	908
2333	Brad Pitt	By the Sea	131	61
2682	Brad Pitt	Killing Them Softly	414	369
2898	Brad Pitt	True Romance	122	460

>>These are overall combined FILMS whose actors are Meryl Streep, Brad Pitt and leonardo Dicaprio.

```
#Combinig all the movies of these leading actors
Combined=Meryl_Streep.append(Leo_Caprio).append(Brad_Pitt)

Combined
```

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
410	Meryl Streep	It's Complicated	187	214
1106	Meryl Streep	The River Wild	42	69
1204	Meryl Streep	Julie & Julia	252	277
1408	Meryl Streep	The Devil Wears Prada	208	631
1483	Meryl Streep	Lions for Lambs	227	298
1575	Meryl Streep	Out of Africa	66	200
1618	Meryl Streep	Hope Springs	234	178
1674	Meryl Streep	One True Thing	64	112
1925	Meryl Streep	The Hours	174	660
2781	Meryl Streep	The Iron Lady	331	350
3135	Meryl Streep	A Prairie Home Companion	211	280
26	Leonardo DiCaprio	Titanic	315	2528
50	Leonardo DiCaprio	The Great Gatsby	490	753
97	Leonardo DiCaprio	Inception	642	2803
179	Leonardo DiCaprio	The Revenant	556	1188
257	Leonardo DiCaprio	The Aviator	267	799
296	Leonardo DiCaprio	Django Unchained	765	1193
307	Leonardo DiCaprio	Blood Diamond	166	657
308	Leonardo DiCaprio	The Wolf of Wall Street	606	1138
326	Leonardo DiCaprio	Gangs of New York	233	1166
361	Leonardo DiCaprio	The Departed	352	2054
452	Leonardo DiCaprio	Shutter Island	490	964
911	Leonardo DiCaprio	Catch Me If You Can	194	667
990	Leonardo DiCaprio	The Beach	118	548
1114	Leonardo DiCaprio	Revolutionary Road	323	414
1422	Leonardo DiCaprio	The Man in the Iron Mask	83	244
1453	Leonardo DiCaprio	J. Edgar	392	279
1560	Leonardo DiCaprio	The Quick and the Dead	63	216
2067	Leonardo DiCaprio	Marvin's Room	45	71
2757	Leonardo DiCaprio	Romeo + Juliet	106	506
3476	Leonardo DiCaprio	The Great Gatsby	490	753
101	Brad Pitt	The Curious Case of Benjamin Button	362	822
147	Brad Pitt	Troy	220	1694
254	Brad Pitt	Ocean's Twelve	198	627
255	Brad Pitt	Mr. & Mrs. Smith	233	798
382	Brad Pitt	Spy Game	142	361
400	Brad Pitt	Ocean's Eleven	186	845
470	Brad Pitt	Fury	406	701
611	Brad Pitt	Seven Years in Tibet	76	119
683	Brad Pitt	Fight Club	315	2968
792	Brad Pitt	Sinbad: Legend of the Seven Seas	98	91
940	Brad Pitt	Interview with the Vampire: The Vampire Chroni...	120	406
1490	Brad Pitt	The Tree of Life	584	975
1722	Brad Pitt	The Assassination of Jesse James by the Coward...	273	415
2204	Brad Pitt	Babel	285	908
2333	Brad Pitt	By the Sea	131	61
2682	Brad Pitt	Killing Them Softly	414	369
2898	Brad Pitt	True Romance	122	460

2. Append the rows of all these DataFrame and store them in a new dataframe named `Combined`.
3.Group the combined dataframe using the `actor_1_name` column.

>> These are overall mean of ‘num_critic_for_reviews’ and ‘num_user_for_reviews’ against the ‘actor_1_name’ column.
>> As from the findings, actor name ‘Leonardo DiCaprio’ has Aced against both the actors.

```
# Write your code for grouping the combined dataframe here
Actor_name=Combined.groupby('actor_1_name')
Actor_name
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000002B2246920D0>
```

```
# Grouping actor_1_name with num_critic_for_reviews
Combined.groupby(['actor_1_name'])['num_critic_for_reviews'].mean().reset_index()
```

	actor_1_name	num_critic_for_reviews
0	Brad Pitt	245.000000
1	Leonardo DiCaprio	330.190476
2	Meryl Streep	181.454545

```
# Grouping actor_1_name with num_user_for_reviews
Combined.groupby(['actor_1_name'])['num_user_for_reviews'].mean().reset_index()
```

	actor_1_name	num_user_for_reviews
0	Brad Pitt	742.352941
1	Leonardo DiCaprio	914.476190
2	Meryl Streep	297.181818

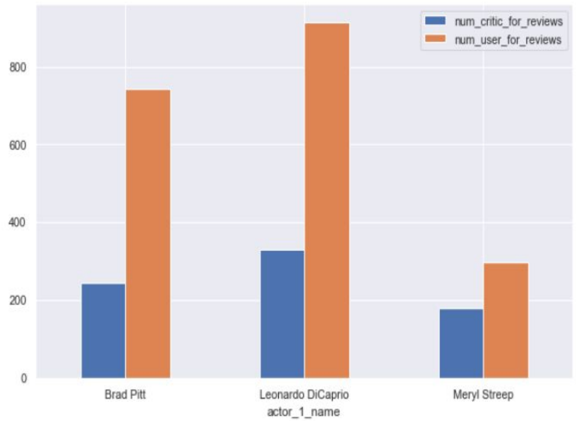
4. Find the mean of the `num_critic_for_reviews` and `num_users_for_review` and identify the actors which have the highest mean.

```
# Grouping all the three variables and finding their mean
Combined.groupby(['actor_1_name'])['num_critic_for_reviews','num_user_for_reviews'].mean()
```

actor_1_name	num_critic_for_reviews	num_user_for_reviews
Brad Pitt	245.000000	742.352941
Leonardo DiCaprio	330.190476	914.476190
Meryl Streep	181.454545	297.181818

```
# Ploting Graph for better visualization.
Combined.groupby(['actor_1_name'])['num_critic_for_reviews','num_user_for_reviews'].mean().plot(kind = 'bar',figsize = (10,
plt.xticks(rotation = 360)
```

```
(array([0, 1, 2]),
 [Text(0, 0, 'Brad Pitt'),
  Text(1, 0, 'Leonardo DiCaprio'),
  Text(2, 0, 'Meryl Streep')])
```



5. Observe the change in number of voted users over decades using a bar chart. Create a column called `decade` which represents the decade to which every movie belongs to. For example, the `title_year` year 1923, 1925 should be stored as 1920s. Sort the DataFrame based on the column `decade`, group it by `decade` and find the sum of users voted in each decade. Store this in a new data frame called `df_by_decade`.

```
# Write the code for calculating decade here
data['decade']=data['title_year'].apply(lambda x: (x//10) *10).astype(np.int64)
data['decade']=data['decade'].astype(str)+'s'
data=data.sort_values(['decade'])
data
```

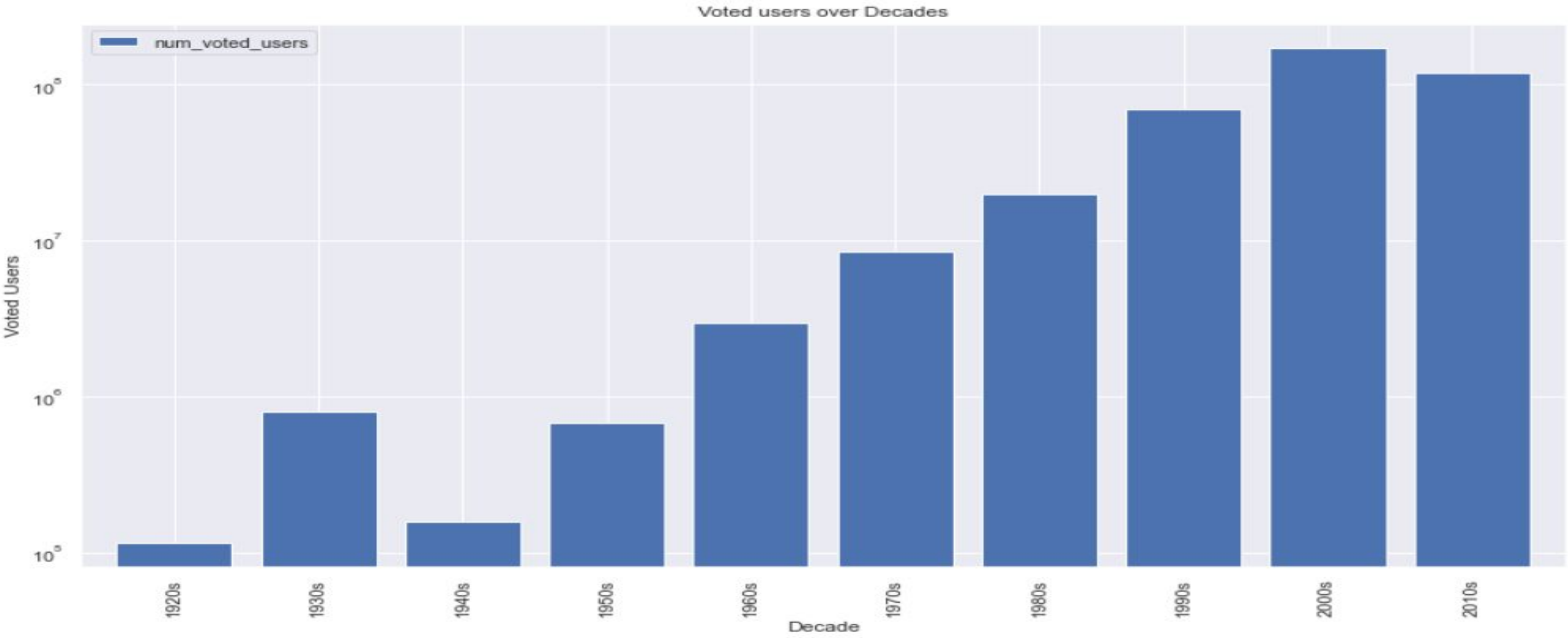
...	content_rating	budget	title_year	actor_2_facebook_likes	imdb_score	aspect_ratio	movie_facebook_likes	profit	IMDb_Top_250	decade
...	Not Rated	6000000	1927	23	8.3	1.33	12000	-5973565	Metropolis	1920s
...	Passed	379000	1929	28	6.3	1.37	167	2429000	The Broadway Melody	1920s
...	R	100000	1920	2	4.8	1.33	0	2900000	Over the Hill to the Poorhouse	1920s
...	Passed	2800000	1939	421	8.1	1.37	14000	19402612	The Wizard of Oz	1930s
...	G	3977000	1939	384	8.2	1.37	16000	194678278	Gone with the Wind	1930s
...	...	...	...	...	...	...	...	...	...	...
...	PG	30000000	2010	60	6.5	2.35	13000	23021560	Letters to Juliet	2010s
...	PG-13	18000000	2016	820	6.7	2.35	0	-14292206	Midnight Special	2010s
...	R	8000000	2013	12	7.4	1.85	43000	8168741	Begin Again	2010s
...	R	8000000	2014	63	5.3	2.35	675	-7947039	Aloft	2010s
...	R	24000000	2010	854	6.5	2.35	12000	-774089	Faster	2010s

```
# Write your code for creating the data frame df_by_decade
df_by_decade=data.groupby('decade')
df_by_decade['num_voted_users'].sum()
df_by_decade=pd.DataFrame(df_by_decade['num_voted_users'].sum())
df_by_decade
```

decade	num_voted_users
1920s	116392
1930s	804839
1940s	159517
1950s	678336
1960s	2982551
1970s	8523299
1980s	19987476
1990s	69720305
2000s	170859021
2010s	119536703

5. Observe the change in number of voted users over decades using a bar chart. Create a column called `decade` which represents the decade to which every movie belongs to. For example, the `title_year` year 1923, 1925 should be stored as 1920s. Sort the DataFrame based on the column `decade`, group it by `decade` and find the sum of users voted in each decade. Store this in a new data frame called `df_by_decade`.

```
# Write your code for plotting number of voted users vs decade
df_by_decade.plot.bar(figsize=(15,8),width=0.8)
plt.xlabel("Decade")
plt.ylabel("Voted Users")
plt.title("Voted users over Decades")
plt.yscale('log')
plt.show()
```



Results

GitHub Link
https://github.com/santy1586/IMDB-Movie-Analysis

1.There are 5044 rows and 28 columns. After cleaning the data and the duplicates. The rows are now 3767 and 27 columns.

2. Dropped the movie_IMDB_link Column as it does not show any intuition.

3. Movies such as ‘Avatar’, ‘Jurassic World’, ‘Titanic’ has gained more profits against their budgets.

4.’The Shawshank Redemption’,’The GodFather’, and ‘The GodFather part 2’ has the highest IMDB rating of 9.3 to 9 as compared to other movies.

5. We can see 5 to 6 Outliers based on the ‘profit ’ column.

6. ‘The good, the bad and the ugly’,’Seven Samurai’ and ‘City of God’ has the highest IMDB rating of 8.9 to 8.7.

7. ‘Christopher Nolan’, ‘Tony Kaye’, ‘Alfred Hitchcock’ these are the Directors who has the highest IMDB rating of all time.

8. 'Adventure|Animation|Drama|Family|Musical' these are the popular genres people/Audience like the most.

9. 'Leonardo DiCaprio' is the critic-favorite as well as the audience-favorite actor.

10. The most users voted in the decade 2000s and the least in the decade 1940s.

Thank You