IMDB Movie Analysis

BY- HIMANSHU KUMAR

Project Description

The Internet Movie Database (IMDB) is a website that serves as an online database of world cinema containing a large number of public data on films such as the title of the film, the year of release of the film, the genre of the film, the audience, budget, revenue, the rating of critics, the duration of the film, the summary of the film, actors, directors and much more.

Tasks Done:-

- **Cleaning the data**
- **Movies with highest profit**
- **Top 250**
- **Best Directors**
- **Popular Genres**
- **Charts**

Approach

The project is done by extracting the csv database to jupiter notebooks for further insight of the data. Various formulas and pandas, numpy, matplotlib functions are used in getting insights of the data respectively.

I've observed the every dataset columns to have the better understanding of the each column.

Tech-Stack Used

I have used Jupiter notebook to perform statistical analysis on this dataset because it allows users to edit, organize, and analyze different types of information. The datasheet was opened in the Jupiter notebook and opened using pandas to achieve the particular tasks.
Also I've used different basic and numpy formulas to gain insights from the dataset.
Also, the matplotlib library for the graphical insights.

Insights

The Internet Movie Database (IMDb) is a website that serves as an online database of world cinema containing a large number of public data on films such as the title of the film, the year of release of the film, the genre of the film, the audience, budget, revenue, the rating of critics, the duration of the film, the summary of the film, actors, directors and much more.
The insights of the data of the movies for the review.

Cleaning the data:

As we calculate the total number of null values in each column we get to know that the column gross has a much number of null values in it.
Then the column budget is the second column with much number of null values in it.
We fill it using numpy feature engineering.

```
#Counting number of null values
movies.isnull().sum()
```

```
#Rounding up to 100th to all columns for null values
movies=movies.drop(['color','director_facebook_likes','actor_1_facebook_likes','actor_2_facebook_likes','actor_3_facebook_likes',
round(100*(movies.isnull().sum()/len(movies.index)),2)
```

```
#Dropping the most Null values column
movies=movies[~np.isnan(movies['gross'])]
movies=movies[~np.isnan(movies['budget'])]

round(100*(movies.isnull().sum()/len(movies.index)),2)
```

```
#Filling the null values column with the features
movies['budget']=movies['budget'].apply(lambda x: round(x/1000000,1))
movies['gross']=movies['gross'].apply(lambda x: round(x/1000000,1))
movies.head()
```

Out[73]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_revie
0	James Cameron	723.0	0.0	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	30
1	Gore Verbinski	302.0	0.0	Action|Adventure|Fantasy	Johnny Depp	Pirates of the Caribbean: At World's End	471220	12
2	Sam Mendes	602.0	0.0	Action|Adventure|Thriller	Christoph Waltz	Spectre	275868	9
3	Christopher Nolan	813.0	0.0	Action|Thriller	Tom Hardy	The Dark Knight Rises	1144337	27
5	Andrew Stanton	462.0	0.0	Action|Adventure|Sci-Fi	Daryl Sabara	John Carter	212204	7
6	Sam Raimi	392.0	0.0	Action|Adventure|Romance	J.K. Simmons	Spider-Man 3	383056	19
7	Nathan Greno	324.0	0.0	Adventure|Animation|Comedy|Family|Fantasy|Musi...	Brad Garrett	Tangled	294810	3
8	Joss Whedon	635.0	0.0	Action|Adventure|Sci-Fi	Chris Hemsworth	Avengers: Age of Ultron	462669	11
9	David Yates	375.0	0.0	Adventure|Family|Fantasy|Mystery	Alan Rickman	Harry Potter and the Half-Blood Prince	321795	9
0	Zack Snyder	673.0	0.0	Action|Adventure|Sci-Fi	Henry Cavill	Batman v Superman: Dawn of Justice	371639	30

Movies with highest profit

As by calculating the difference between the gross and budget, we get to know the profit as a new column. From which we can plot a graph to see the outliers in them. The outliers present in the sheet

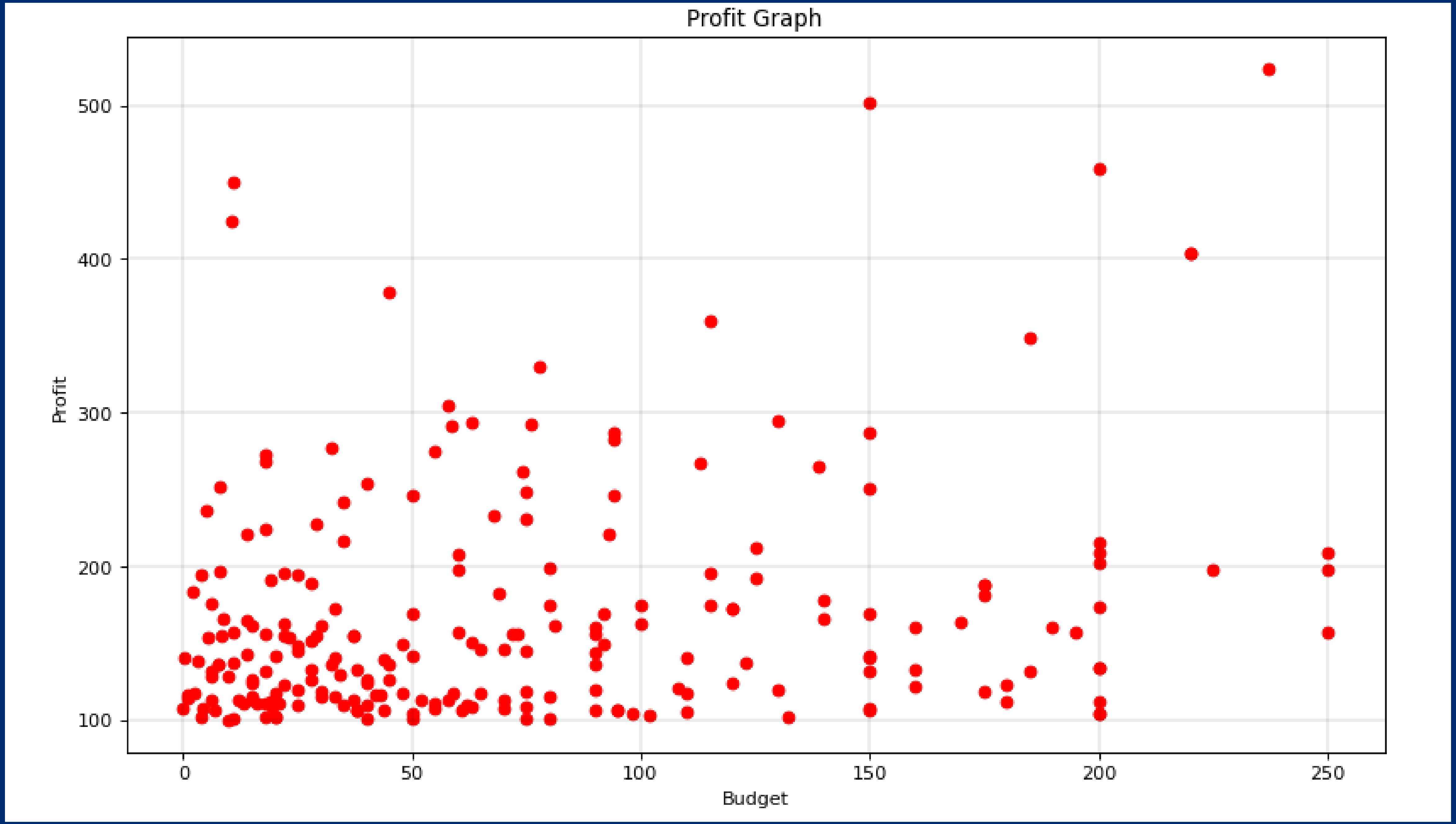
```
In [ ]: #Movies with high profit
        movies['profit']=movies['gross']-movies['budget']
        movie=movies.sort_values(by=['profit'],ascending=False)

In [ ]: movie.head(10)

In [ ]: plt.figure(num=None, figsize=(12,7), dpi=80)
        movie=movie[movie.profit>100]
        plt.scatter(movie['budget'], movie['profit'],marker ="o",facecolor='red')
        plt.xlabel("Budget")
        plt.ylabel("Profit")
        plt.title("Profit Graph")
        plt.grid(color='black', linestyle='-', linewidth=0.25, alpha=0.5)
        plt.show()
```

Out[82]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews
0	James Cameron	723.0	760.5	Action|Adventure|Fantasy|Sci-Fi	CCH Pounder	Avatar	886204	3054
29	Colin Trevorrow	644.0	652.2	Action|Adventure|Sci-Fi|Thriller	Bryce Dallas Howard	Jurassic World	418214	1290
26	James Cameron	315.0	658.7	Drama|Romance	Leonardo DiCaprio	Titanic	793059	2528
3024	George Lucas	282.0	460.9	Action|Adventure|Fantasy|Sci-Fi	Harrison Ford	Star Wars: Episode IV - A New Hope	911097	1470
3080	Steven Spielberg	215.0	434.9	Family|Sci-Fi	Henry Thomas	E.T. the Extra-Terrestrial	281842	515
794	Joss Whedon	703.0	623.3	Action|Adventure|Sci-Fi	Chris Hemsworth	The Avengers	995415	1722
17	Joss Whedon	703.0	623.3	Action|Adventure|Sci-Fi	Chris Hemsworth	The Avengers	995415	1722
509	Roger Allers	186.0	422.8	Adventure|Animation|Drama|Family|Musical	Matthew Broderick	The Lion King	644348	656
240	George Lucas	320.0	474.5	Action|Adventure|Fantasy|Sci-Fi	Natalie Portman	Star Wars: Episode I - The Phantom Menace	534658	3597
66	Christopher Nolan	645.0	533.3	Action|Crime|Drama|Thriller	Christian Bale	The Dark Knight	1676169	4667


Profit Graph
Profit
Budget
100
200
300
400
500
0
50
100
150
200
250

Top 250

The top 250 movies with the highest IMDb Rating with respect to their IMDB scores and the number of voted_users is greater than 25,000.

Also, the movies in the IMDb Top 250 column which are not in the English language.

```
In [ ]: #Checking out the top 250 imdb movies with the highest IMDb Rating and the num_voted_users is greater than 25,000 with
        #their ranking
        IMDb_Top_250=movies[['imdb_score','num_voted_users','movie_title','language']]

        IMDb_sort= IMDb_Top_250.sort_values(by=['imdb_score'],ascending=False)
        IMDb_Top_250=IMDb_sort[IMDb_Top_250.num_voted_users>25000]
        IMDb_Top_250["Rank"]=IMDb_Top_250['movie_title'].rank()
        IMDb_Top_250['Rank']=IMDb_Top_250['Rank'].sort_values(ascending=True).values
        IMDb_Top_250.head(250)
```

```
In [ ]: # the movies in the IMDb_Top_250 column which are not in the English language
        Top_Foreign_Lang_Film = IMDb_Top_250[(IMDb_Top_250.language !='English') ]
        Top_Foreign_Lang_Film[0:250]
```

Top 250 movies

Out[118]:

	imdb_score	num_voted_users	movie_title	language	Rank
1937	9.3	1689764	The Shawshank Redemption	English	1.0
3466	9.2	1155770	The Godfather	English	2.0
2837	9.0	790926	The Godfather: Part II	English	3.0
66	9.0	1676169	The Dark Knight	English	4.0
339	8.9	1215718	The Lord of the Rings: The Return of the King	English	5.0
...	...	...	...	...	...
1871	7.9	483756	Taken	English	246.0
23	7.9	483540	The Hobbit: The Desolation of Smaug	English	247.0
1884	7.9	219008	The Untouchables	English	248.0
4640	7.9	44763	4 Months, 3 Weeks and 2 Days	Romanian	249.0
4931	7.9	90827	Once	English	250.0

250 rows × 5 columns

Top 250 movies with foreign language

Out[119]:

	imdb_score	num_voted_users	movie_title	language	Rank
4498	8.9	503509	The Good, the Bad and the Ugly	Italian	8.0
4747	8.7	229012	Seven Samurai	Japanese	15.0
4029	8.7	533200	City of God	Portuguese	16.0
2373	8.6	417971	Spirited Away	Japanese	28.0
4259	8.5	259379	The Lives of Others	German	34.0
...	...	...	...	...	...
3883	6.5	47097	Night Watch	Russian	1564.5
377	6.4	86152	The Interpreter	Aboriginal	1619.0
4671	6.4	54601	Dead Snow	Norwegian	1702.0
484	5.9	71574	The Legend of Zorro	Spanish	2120.0
2890	4.3	31414	In the Land of Blood and Honey	Bosnian	2565.0

91 rows × 5 columns

Best Directors

From the database we extract the best 10 directors with highest IMDB_Score.

```
In [133]: #Best 10 directors with good imdb score
          mov=movies.groupby('director_name')
          top10director=pd.DataFrame(mov['imdb_score'].mean().sort_values(ascending=False))
          top10director=top10director.head(10)
          top10director=top10director.sort_values(['imdb_score','director_name'],ascending=(False,True))
          top10director
```

Out[133]:

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
Marius A. Markevicius	8.400000
S.S. Rajamouli	8.400000

Popular Genres

The most popular genres o the IMDB according to the IMDB_scores.

```
In [134]: movies['genres']=movies.genres.str.split('|')
          movies['genre_1']=movies['genres'].apply(lambda x: x[0])
          movies['genre_2']=movies['genres'].apply(lambda x: x[1] if len(x)>1 else x[0])
          movies.head()
```

		gross
genre_1	**genre_2**	
Family	**Sci-Fi**	434.900000
Adventure	**Sci-Fi**	228.637500
	Family	118.929412
	Animation	116.997436
Action	**Adventure**	109.597087
...	**...**	...
Horror	**Musical**	0.100000
Romance	**Romance**	0.100000
Thriller	**Thriller**	0.033333
Adventure	**War**	0.000000
Sci-Fi	**Sci-Fi**	0.000000

103 rows × 1 columns

Out[150]:

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_year	i
0	James Cameron	723.0	760.5	[Action, Adventure, Fantasy, Sci-Fi]	CCH Pounder	Avatar	886204	3054	English	237.0	2009.0	
1	Gore Verbinski	302.0	309.4	[Action, Adventure, Fantasy]	Johnny Depp	Pirates of the Caribbean: At World's End	471220	1238	English	300.0	2007.0	
2	Sam Mendes	602.0	200.1	[Action, Adventure, Thriller]	Christoph Waltz	Spectre	275868	994	English	245.0	2015.0	
3	Christopher Nolan	813.0	448.1	[Action, Thriller]	Tom Hardy	The Dark Knight Rises	1144337	2701	English	250.0	2012.0	
5	Andrew Stanton	462.0	73.1	[Action, Adventure, Sci-Fi]	Daryl Sabara	John Carter	212204	738	English	263.7	2012.0	
6	Sam Raimi	392.0	336.5	[Action, Adventure, Romance]	J.K. Simmons	Spider-Man 3	383056	1902	English	258.0	2007.0	
7	Nathan Greno	324.0	200.8	[Adventure, Animation, Comedy, Family, Fantasy...	Brad Garrett	Tangled	294810	387	English	260.0	2010.0	
8	Joss Whedon	635.0	459.0	[Action, Adventure, Sci-Fi]	Chris Hemsworth	Avengers: Age of Ultron	462669	1117	English	250.0	2015.0	
9	David Yates	375.0	302.0	[Adventure, Family, Fantasy, Mystery]	Alan Rickman	Harry Potter and the Half-Blood Prince	321795	973	English	250.0	2009.0	
10	Zack Snyder	673.0	330.2	[Action, Adventure, Sci-Fi]	Henry Cavill	Batman v Superman: Dawn of Justice	371639	3018	English	250.0	2016.0	

Charts

Three new columns 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, the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.

```
# new columns 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.
Meryl_Streep=movies[['actor_1_name','movie_title','num_critic_for_reviews','num_user_for_reviews']]
Leo_Caprio=movies[['actor_1_name','movie_title','num_critic_for_reviews','num_user_for_reviews']]
Brad_Pitt=movies[['actor_1_name','movie_title','num_critic_for_reviews','num_user_for_reviews']]
```

Meryl_Streep

```
# Meryl_Streep Movies
Meryl_Streep=Meryl_Streep.loc[Meryl_Streep['actor_1_name']=='Meryl Streep',:]
Meryl_Streep.head()
```

Out[185]:

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
410	Meryl Streep	It's Complicated	187.0	214
1106	Meryl Streep	The River Wild	42.0	69
1204	Meryl Streep	Julie & Julia	252.0	277
1408	Meryl Streep	The Devil Wears Prada	208.0	631
1483	Meryl Streep	Lions for Lambs	227.0	298

Leo_Caprio

```
# Leo_Caprrio Movies
Leo_Caprio=Leo_Caprio.loc[Leo_Caprio['actor_1_name']=='Leonardo DiCaprio',:]
Leo_Caprio.head()
```

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
26	Leonardo DiCaprio	Titanic	315.0	2528
50	Leonardo DiCaprio	The Great Gatsby	490.0	753
97	Leonardo DiCaprio	Inception	642.0	2803
179	Leonardo DiCaprio	The Revenant	556.0	1188
257	Leonardo DiCaprio	The Aviator	267.0	799

Brad_Pitt

```
# Brad_Pitt movies
Brad_Pitt=Brad_Pitt.loc[Brad_Pitt['actor_1_name']=='Brad Pitt',:]
Brad_Pitt.head()
```

	actor_1_name	movie_title	num_critic_for_reviews	num_user_for_reviews
101	Brad Pitt	The Curious Case of Benjamin Button	362.0	822
147	Brad Pitt	Troy	220.0	1694
254	Brad Pitt	Ocean's Twelve	198.0	627
255	Brad Pitt	Mr. & Mrs. Smith	233.0	798
382	Brad Pitt	Spy Game	142.0	361

Append of the rows

```
# New column for all Append of the rows
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	187.0	214
1106	Meryl Streep	The River Wild	42.0	69
1204	Meryl Streep	Julie & Julia	252.0	277
1408	Meryl Streep	The Devil Wears Prada	208.0	631
1483	Meryl Streep	Lions for Lambs	227.0	298
1575	Meryl Streep	Out of Africa	66.0	200
1618	Meryl Streep	Hope Springs	234.0	178
1674	Meryl Streep	One True Thing	64.0	112
1925	Meryl Streep	The Hours	174.0	660
2781	Meryl Streep	The Iron Lady	331.0	350
3135	Meryl Streep	A Prairie Home Companion	211.0	280
26	Leonardo DiCaprio	Titanic	315.0	2528
50	Leonardo DiCaprio	The Great Gatsby	490.0	753
97	Leonardo DiCaprio	Inception	642.0	2803
179	Leonardo DiCaprio	The Revenant	556.0	1188
257	Leonardo DiCaprio	The Aviator	267.0	799
296	Leonardo DiCaprio	Django Unchained	765.0	1193
307	Leonardo DiCaprio	Blood Diamond	166.0	657
308	Leonardo DiCaprio	The Wolf of Wall Street	606.0	1138
326	Leonardo DiCaprio	Gangs of New York	233.0	1166
361	Leonardo DiCaprio	The Departed	352.0	2054
452	Leonardo DiCaprio	Shutter Island	490.0	964
641	Leonardo DiCaprio	Body of Lies	238.0	263
911	Leonardo DiCaprio	Catch Me If You Can	194.0	667
990	Leonardo DiCaprio	The Beach	118.0	548
1114	Leonardo DiCaprio	Revolutionary Road	323.0	414
1422	Leonardo DiCaprio	The Man in the Iron Mask	83.0	244
1453	Leonardo DiCaprio	J. Edgar	392.0	279
1560	Leonardo DiCaprio	The Quick and the Dead	63.0	216
2067	Leonardo DiCaprio	Marvin's Room	45.0	71

Combined column

```
# Group the combined column
Actor_name=Combined.groupby('actor_1_name')
Actor_name
```

Mean of the num_critic_for_reviews and num_users_for_review

```
# The mean of the num_critic_for_reviews and num_users_for_review
Critic_reviews=Actor_name['num_critic_for_reviews'].mean().sort_values(ascending=False)
Critic_reviews
```

```
actor_1_name
Leonardo DiCaprio    330.190476
Brad Pitt            245.000000
Meryl Streep         181.454545
Name: num_critic_for_reviews, dtype: float64
```

Represents the decade to which every movie belongs to

```
# The change in number of voted users over decades using a bar chart.
# Creating a column called decade which represents the decade to which every movie belongs to.
Audience_reviews=Actor_name['num_user_for_reviews'].mean().sort_values(ascending=False)
Audience_reviews.head()
```

```
movies['decade']=movies['title_year'].apply(lambda x: (x//10) *10).astype(np.int64)
movies['decade']=movies['decade'].astype(str)+'s'
movies=movies.sort_values(['decade'])
movies
```

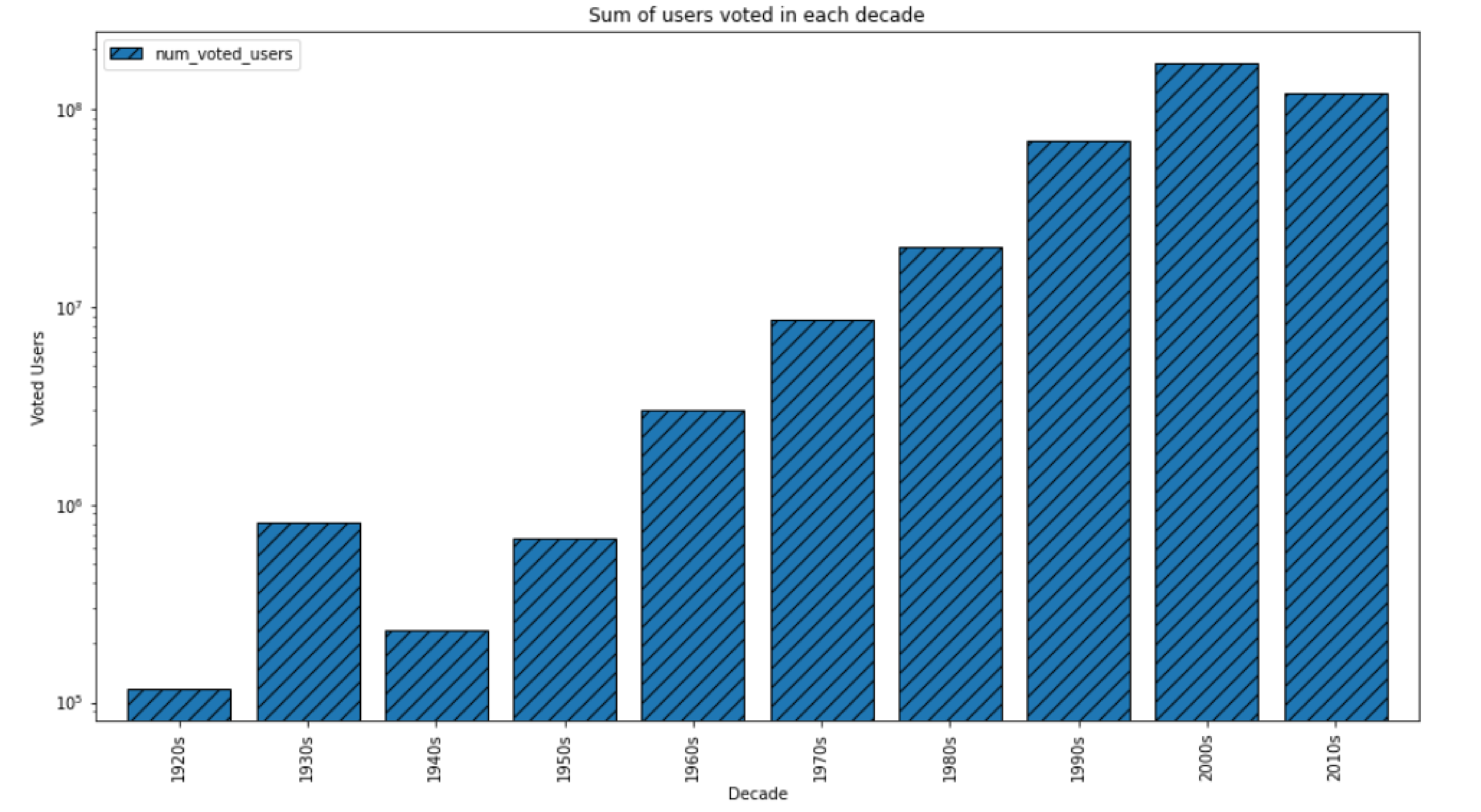
```
df_by_decade=movies.groupby('decade')
df_by_decade['num_voted_users'].sum()
df_by_decade=pd.DataFrame(df_by_decade['num_voted_users'].sum())
df_by_decade
```

```
df_by_decade.plot.bar(figsize=(15,8),width=0.8,hatch="//",edgecolor='k')
plt.xlabel("Decade")
plt.ylabel("Voted Users")
plt.title("Sum of users voted in each decade")
plt.yscale('log')
plt.show()
```

	director_name	num_critic_for_reviews	gross	genres	actor_1_name	movie_title	num_voted_users	num_user_for_reviews	language	budget	title_yea
4812	Harry Beaumont	36.0	2.8	[Musical, Romance]	Anita Page	The Broadway Melody	4546	71	English	0.4	1929.
4958	Harry F. Millarde	1.0	3.0	[Crime, Drama]	Stephen Carr	Over the Hill to the Poorhouse	5	1	NaN	0.1	1920.
2734	Fritz Lang	260.0	0.0	[Drama, Sci-Fi]	Brigitte Helm	Metropolis	111841	413	German	6.0	1927.
4157	Victor Fleming	213.0	22.2	[Adventure, Family, Fantasy, Musical]	Margaret Hamilton	The Wizard of Oz	291875	533	English	2.8	1939.
4706	Mark Sandrich	66.0	3.0	[Comedy, Musical, Romance]	Ginger Rogers	Top Hat	13269	98	English	0.6	1935.
...	...	...	...	...	...	...	...	...	...	...	.
3470	Steven Soderbergh	324.0	113.7	[Comedy, Drama]	Channing Tatum	Magic Mike	108843	281	English	7.0	2012.
781	Martin Campbell	258.0	43.3	[Crime, Drama, Mystery, Thriller]	Bojana Novakovic	Edge of Darkness	75201	256	English	80.0	2010.
2495	Malcolm D. Lee	56.0	70.5	[Comedy, Drama]	Harold Perrineau	The Best Man Holiday	11600	64	English	17.0	2013.
1668	Steven Soderbergh	450.0	32.2	[Crime, Drama, Thriller]	Channing Tatum	Side Effects	148327	274	English	30.0	2013.
3264	Michael Haneke	447.0	0.2	[Drama, Romance]	Isabelle Huppert	Amour	70382	190	French	8.9	2012.

3856 rows × 17 columns

decade	num_voted_users
1920s	116392
1930s	804839
1940s	230838
1950s	678336
1960s	2983442
1970s	8524102
1980s	19987476
1990s	69735679
2000s	170908676
2010s	120640994



Result

I have enjoyed while doing this project was fun because I've learned various numpy, pandas functions and formulas to gain insight from any dataset. I came to know a new set of regulation and to use another set of rules to extract insights from the extracted csv dataset.

STUDIO SHODWE

Thank You