ProjectNumber 5

IMDB Movie Analysis

Project Description

This project has provided us with an IMDB movie dataset that needs to be studied and analysed to answer the questions asked and to draw insights from the analysis. This project on completion will answer many important questions like which movies earned highest or which director was best or which movies lie in the top imdb score list and many more.

Approach

The approach to do task is enough straight forward. I started out with the data cleaning process since the handed dataset is huge and similar huge datasets tend to have problems like missing data, indistinguishable data and wrong format. After all the cleaning process, I took a look at the questions asked and set up a way to answer them after performing analysis on the new clean data. For all these tasks I used Jupyter Notebook i.e. python programming language.

Tech Stack used

- Google collab
- Excel
- Google drive

My collab file link:

https://colab.research.google.com/drive/1uIE18_M3Txet99pDvgFEaN4nVuYtke5J?usp=sharing

Insights:

A) **Your task:** Clean the data.

Step by step approach:

Step 1) imported libraries

Step 2) imported the dataset using the pandas library into my collab notebook and made a copy of the original dataset.

Step 3) ran an issnull() function to see the places with missing data.

Step 4) found the percentages of null values present in the dataset COLUMNWISE. Using the sum() and sort_values() functions.

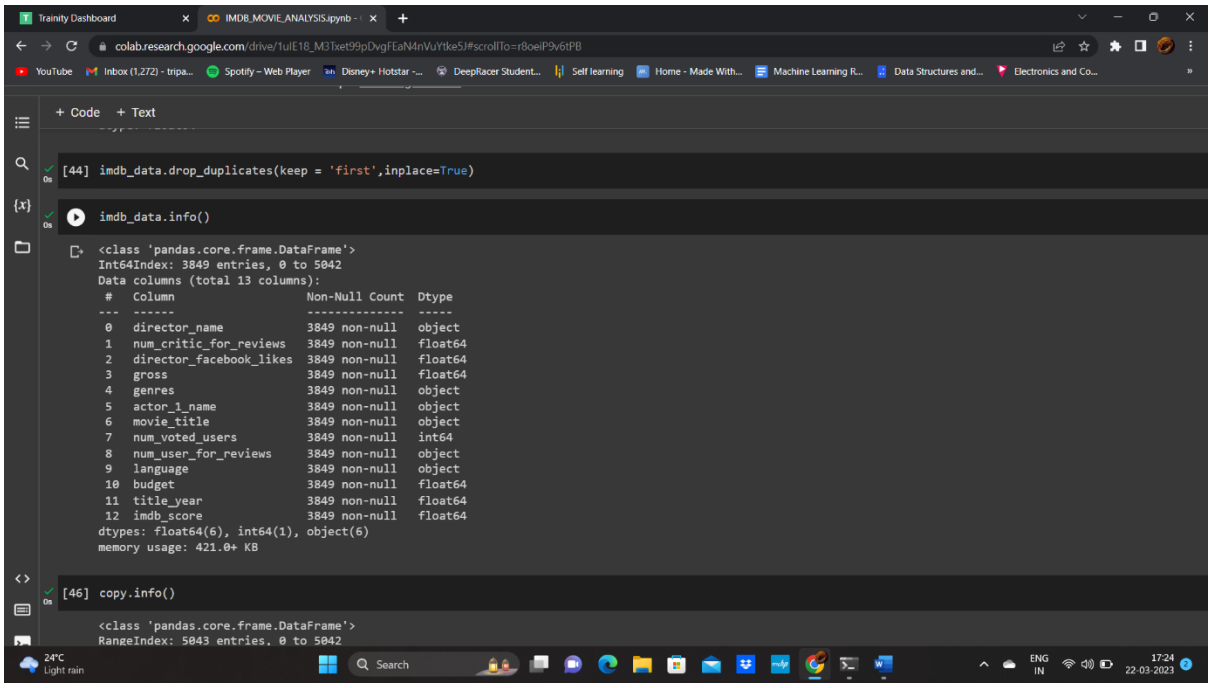
Step 5) dropped unnecessary columns that need not to be present for analysis and thus making our dataset light and easy. Function used: drop() function.

Step 6) there are some columns that have null values in them and so to remove null values from our dataset , I removed the ROWS which had missing values pertaining to the columns.
For example: **imdb_data = imdb_data[imdb_data['gross'].notnull()]**

Step 7) there were duplicate rows present in the data. So removed those using drop_duplicates() function.

Step 8) printed information about the new cleaned dataset and and compared its information with the copy of the original dataset.

The ss showing details of the cleaned dataset.



Final result of this task:

Dimension of the data:

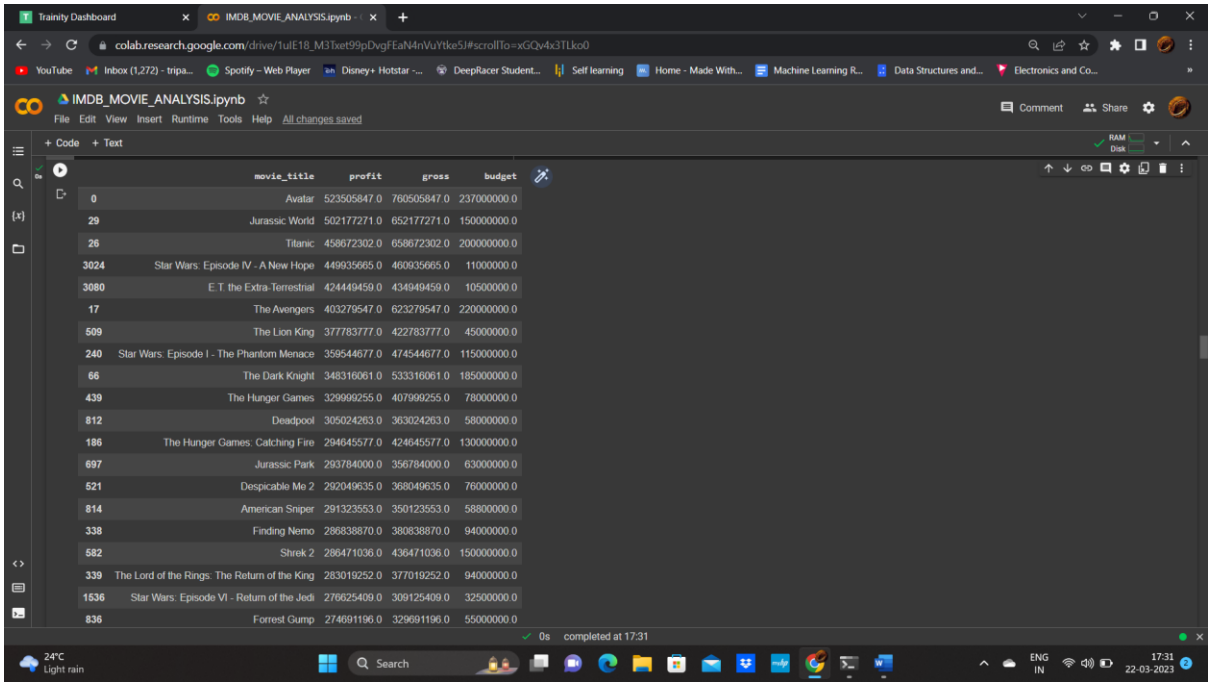
- Before cleaning: 5043 x 28
- After cleaning: 3849 x 13

B) **Your task:** Find the movies with the highest profit? Plot profit (y-axis) vs budget (x- axis) and observe the outliers using the appropriate chart type.

Step by step approach:
Step 1) created a 'profit' column in our dataset having difference of the 'gross' and the 'budget' column.
Step 2) sorted the dataset according to the new profit column and extracted top 20 of the movies. Function used : sort_values() and head().
Step 3) printed the data with not all but necessary columns to verify the task.
Step 4) created a barplot of 'movie_title' Vs 'profit' for better visualization. The graph shows how the profit is decreasing along the x axis.

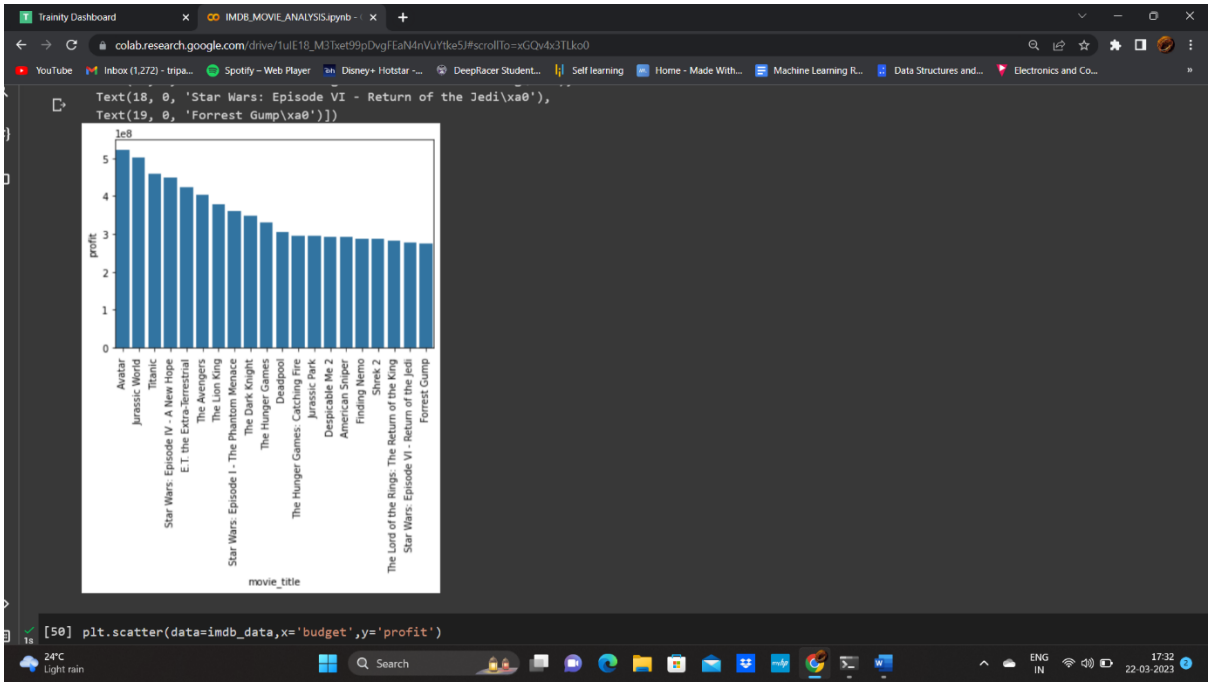
Step 5) for the second part of the task, I plotted a scatter plot of ‘budget’ Vs ‘profit’ to see the outlier distribution in our data.

The ss shows top 20 profitable movies information.

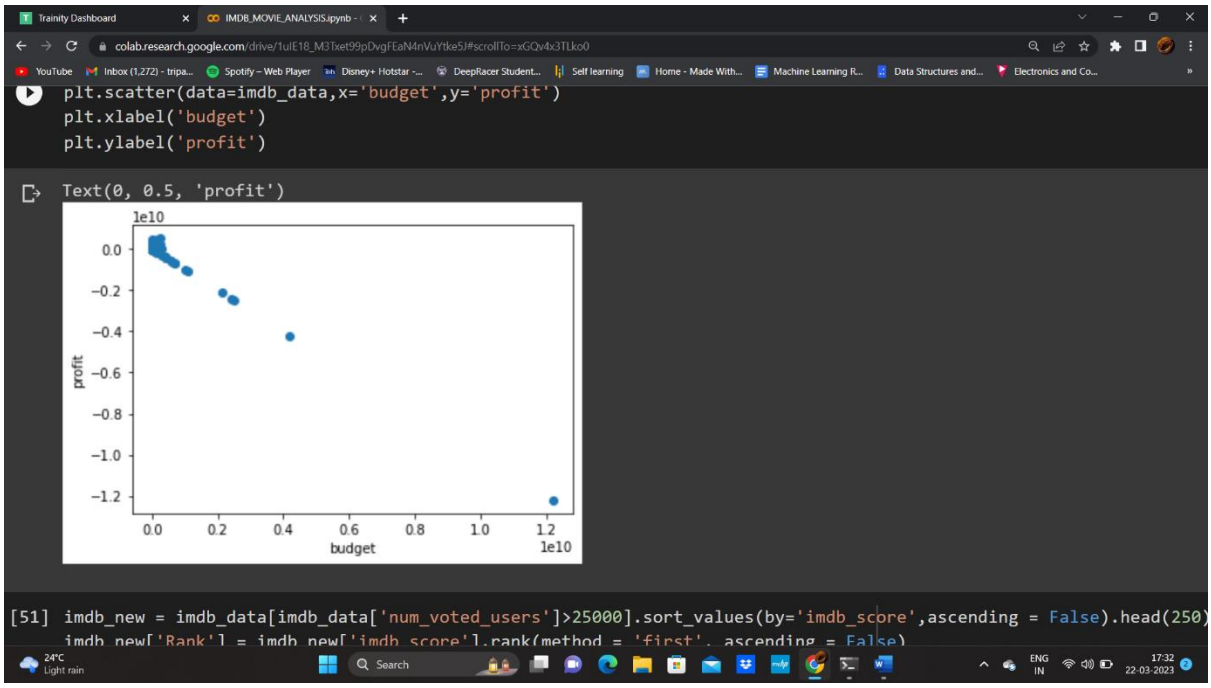


	movie_title	profit	gross	budget
0	Avatar	523505847.0	760505847.0	237000000.0
29	Jurassic World	502177271.0	652177271.0	150000000.0
26	Titanic	458672302.0	658672302.0	200000000.0
3024	Star Wars: Episode IV - A New Hope	449935665.0	460935665.0	11000000.0
3080	E.T. the Extra-Terrestrial	424449459.0	434949459.0	10500000.0
17	The Avengers	403279547.0	623279547.0	220000000.0
509	The Lion King	377783777.0	422783777.0	45000000.0
240	Star Wars: Episode I - The Phantom Menace	359544677.0	474544677.0	115000000.0
66	The Dark Knight	348316061.0	533316061.0	185000000.0
439	The Hunger Games	329999255.0	407999255.0	78000000.0
812	Deadpool	305024263.0	363024263.0	58000000.0
186	The Hunger Games: Catching Fire	294645577.0	424645577.0	130000000.0
697	Jurassic Park	293784000.0	356784000.0	63000000.0
521	Despicable Me 2	292049635.0	368049635.0	76000000.0
814	American Sniper	291323553.0	350123553.0	58800000.0
338	Finding Nemo	286838870.0	380838870.0	94000000.0
582	Shrek 2	286471036.0	436471036.0	150000000.0
339	The Lord of the Rings: The Return of the King	283019252.0	377019252.0	94000000.0
1536	Star Wars: Episode VI - Return of the Jedi	276625409.0	309125409.0	32500000.0
836	Forrest Gump	274691196.0	329691196.0	55000000.0

The ss shows the bar plot of movies Vs profit for visualization.



The ss shows scatter plot of budget Vs profit to see the outlier distribution in dataset.



c) **Your task: Find IMDB Top 250.** Also make sure that for all of these movies, the num_voted_users is greater than 25,000. Also create a new column named rank to give rank from 1 to 250 for these movies. Find out all the columns in this top 250 movies which are not in English language.

step by step approach:

Step 1) created a new dataset form the imdb_movies dataset and extracted only that data for which the 'num_voted_users' were greater than 25000 and then sorted the data on the basis of 'imdb_score' and extracted first 250 of them.

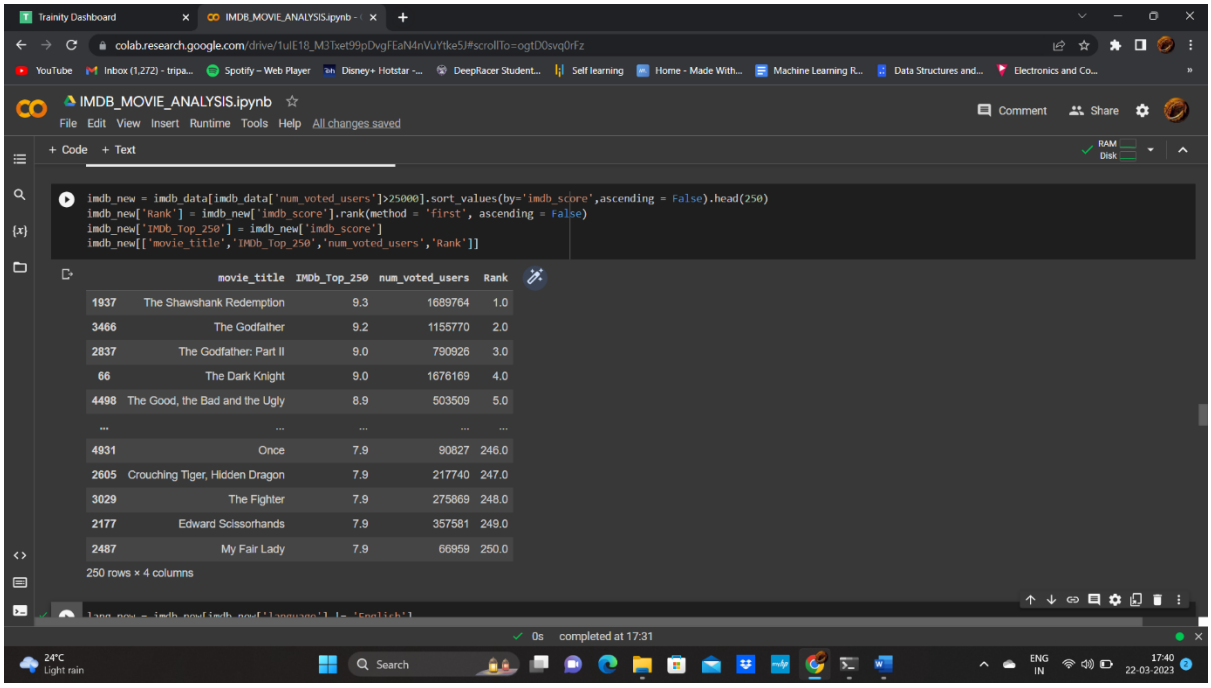
Step2) created a new column named 'Rank' and generated the ranks ranging from 1 to 250 for every row using the .rank() function.

Step 3) created a column named 'IMDB_Top_250' and stored the 'imdb_score' data in it.

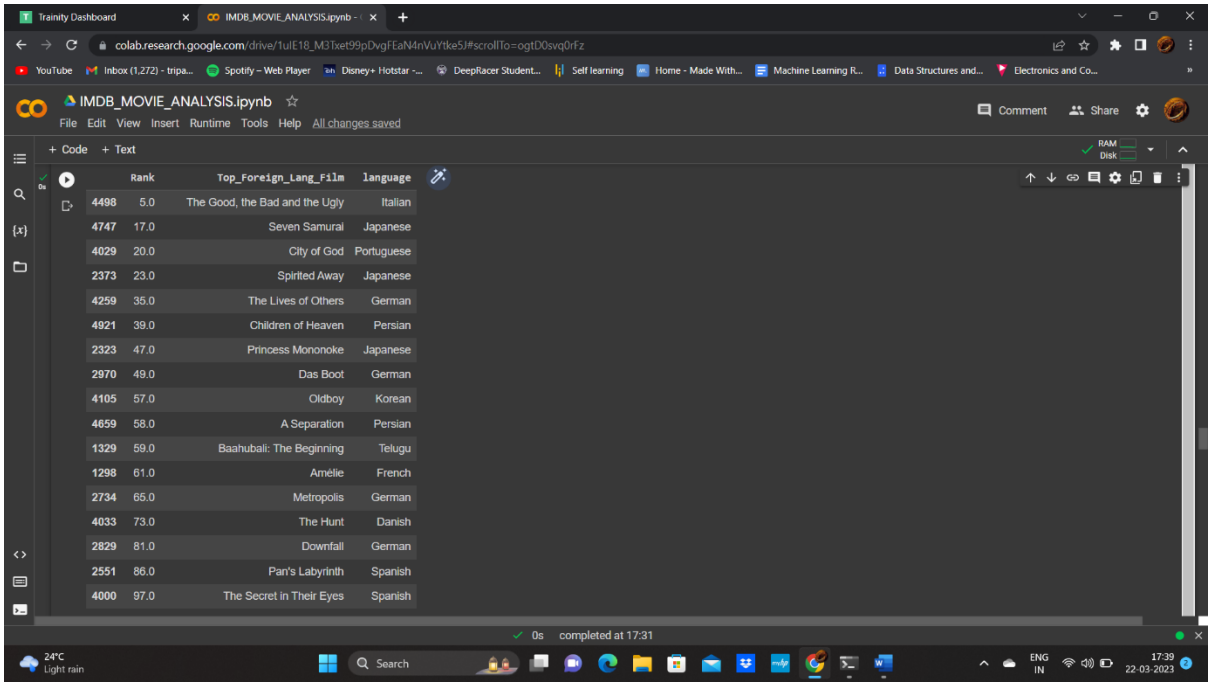
Step 4) displayed some of the columns from the dataset that verify that the task was done like 'movie_title' , 'IMDB_Top_250' , 'num_voted_users' , 'Rank' .

Step 5) now for the second part of the task, I created a new dataset for storing those data out of the data of the top 250 imdb scored movies that are NOT IN ENGLISH LANGUAGE. Added a new column named 'Top_Foreign_Lang_Film' and then displayed relevant information verifying the task.

The ss shows details about top 250 imdb movies with num_voted_users >25000.



The ss shows details of movies in the top 250 imdb movies where language was NOT ENGLISH.



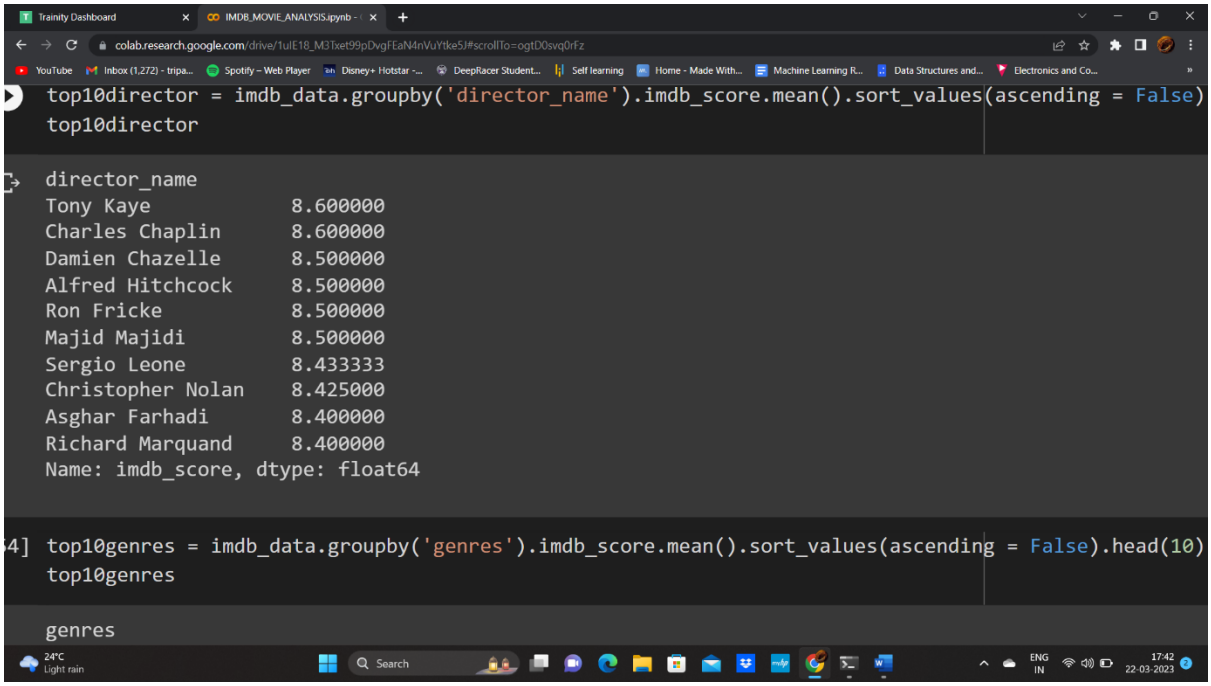
	Rank	Top_Foreign_Lang_Film	language
4498	5.0	The Good, the Bad and the Ugly	Italian
4747	17.0	Seven Samurai	Japanese
4029	20.0	City of God	Portuguese
2373	23.0	Spirited Away	Japanese
4259	35.0	The Lives of Others	German
4921	39.0	Children of Heaven	Persian
2323	47.0	Princess Mononoke	Japanese
2970	49.0	Das Boot	German
4105	57.0	Oldboy	Korean
4659	58.0	A Separation	Persian
1329	59.0	Baahubali: The Beginning	Telugu
1298	61.0	Amélie	French
2734	65.0	Metropolis	German
4033	73.0	The Hunt	Danish
2829	81.0	Downfall	German
2551	86.0	Pan's Labyrinth	Spanish
4000	97.0	The Secret in Their Eyes	Spanish

D) **Your task:** Find the best directors based on the mean of the 'imdb_score'.

Step by step approach:

Step 1) I grouped the data by the 'director_name' and then found the mean of the 'imdb_score' for this grouped data and sorted the resulting data in descending order and extracted top 10 of them.
Step2) stored them in 'top10directors' and displayed.

The ss shows top 10 directors based on mean of imdb score along with the mean.



```
top10director = imdb_data.groupby('director_name').imdb_score.mean().sort_values(ascending = False)
top10director
```

```
director_name
Tony Kaye            8.600000
Charles Chaplin      8.600000
Damien Chazelle      8.500000
Alfred Hitchcock     8.500000
Ron Fricke           8.500000
Majid Majidi         8.500000
Sergio Leone         8.433333
Christopher Nolan    8.425000
Asghar Farhadi       8.400000
Richard Marquand     8.400000
Name: imdb_score, dtype: float64
```

```
4] top10genres = imdb_data.groupby('genres').imdb_score.mean().sort_values(ascending = False).head(10)
top10genres
```

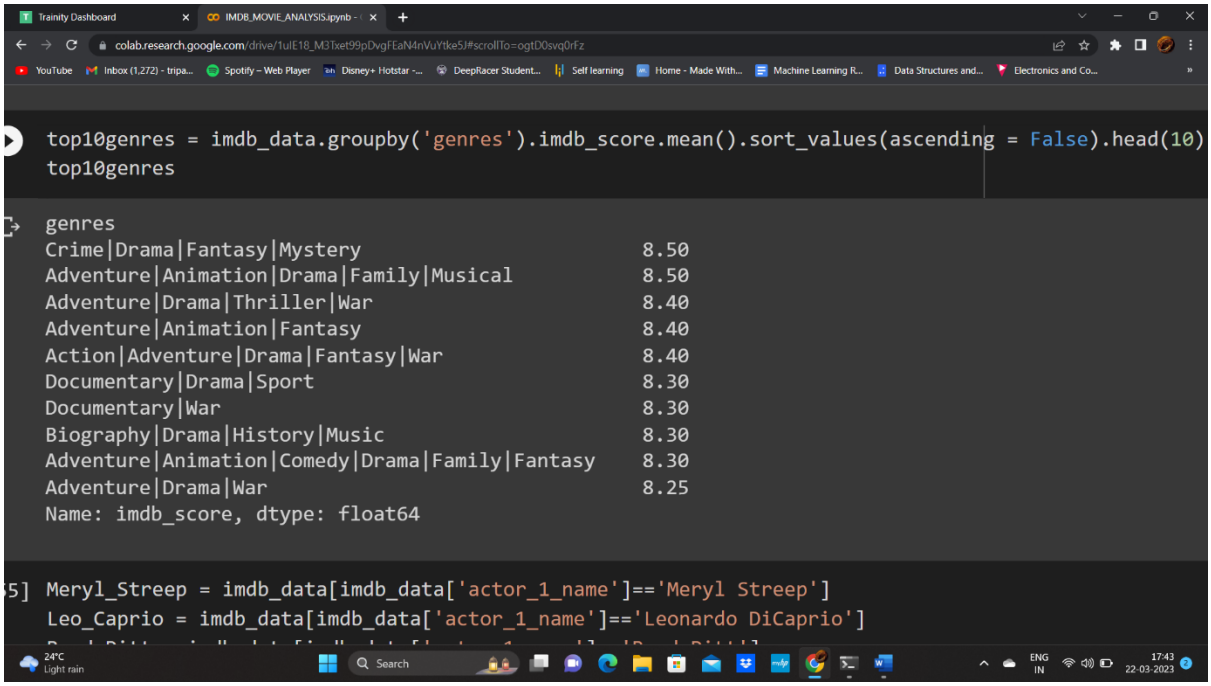
```
genres
```

E) **Your task:** Find the popular genres.

Step by step approach:

Step 1) I grouped the data by the 'genres' and then found the mean of the 'imdb_score' for this grouped data and sorted the resulting data in descending order and extracted top 10 of them.
Step2) stored them in 'top10genres' and displayed.

The ss shows the top 10 genres based on mean of imdb score.



```
top10genres = imdb_data.groupby('genres').imdb_score.mean().sort_values(ascending = False).head(10)
top10genres
```

```
genres
Crime|Drama|Fantasy|Mystery                          8.50
Adventure|Animation|Drama|Family|Musical             8.50
Adventure|Drama|Thriller|War                         8.40
Adventure|Animation|Fantasy                          8.40
Action|Adventure|Drama|Fantasy|War                   8.40
Documentary|Drama|Sport                              8.30
Documentary|War                                      8.30
Biography|Drama|History|Music                        8.30
Adventure|Animation|Comedy|Drama|Family|Fantasy      8.30
Adventure|Drama|War                                  8.25
Name: imdb_score, dtype: float64
```

```
Meryl_Streep = imdb_data[imdb_data['actor_1_name']=='Meryl Streep']
Leo_Caprio = imdb_data[imdb_data['actor_1_name']=='Leonardo DiCaprio']
```

F) **Your task:** Find out the critic favorite as well as audience favorite actor among these 3 actors. Create 3 columns named, Meryl_Streep, Leo_Caprio, and Brad_Pitt with movies only sin which these 3 were the top actors. Add the rows of all these columns and store them in a newly created column with name Combined.

use actor 1 name column to group the combined column.
Create a column with name decade in which the decade to which every movie belongs to. Sort the column 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.

Step by step approach:

Step 1) created 3 new columns with the names 'Meryl_Streep' , 'Leo_Caprio' , 'Brad_Pitt' from the dataset having these three actors specified in the 'actor_1_name' column respectively.

Step 2) using the first data i.e. 'Meryl_Streep' . I appended the rest tow into a new data named 'Combined'

Step 3) for finding the critic favourite actor out of these 3, I grouped the data by 'actor_1_name' and found the mean of the 'num_for_critic_review'. The one with the highest mean will be critic favourite. Answer here: ***Leonardo DiCaprio***

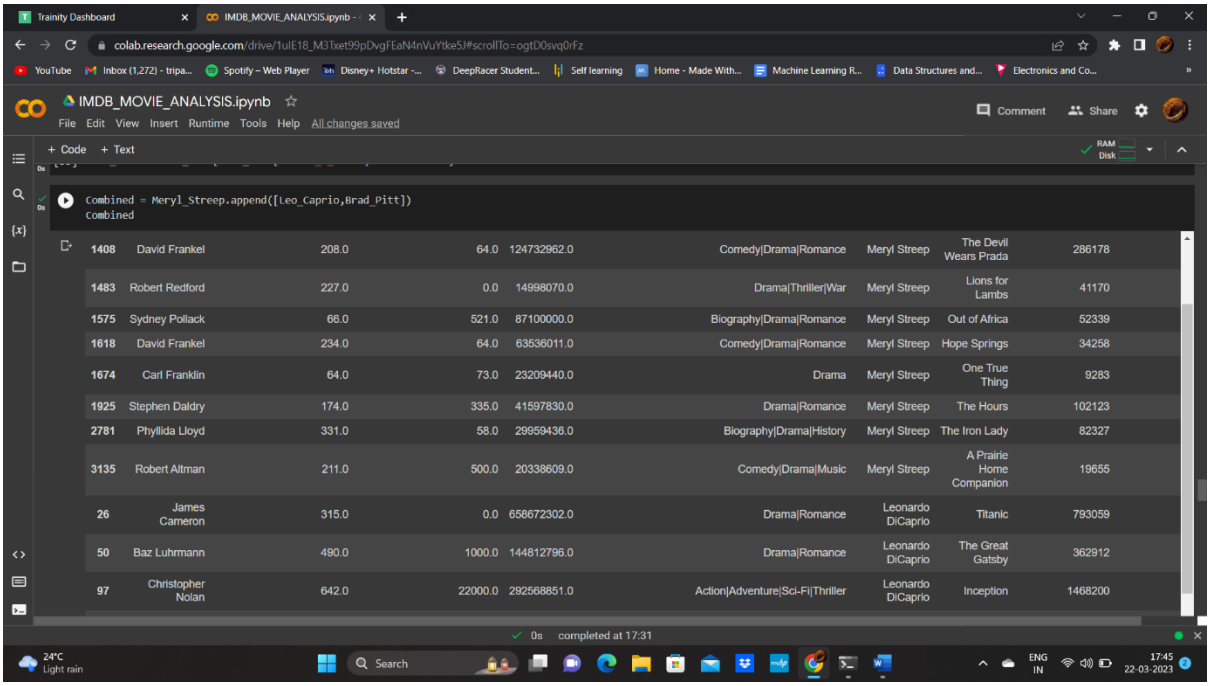
Step 4) for finding the audience favourite actor out of these 3, I grouped the data by 'actor_1_name' and found the mean of the 'num_user_for_reviews'. The one with the highest mean will be audience favourite. Answer here: ***Leonardo DiCaprio***

Step 6) for the second part of the task, I needed to create intervals of 10 years indicating each ‘deacde’
So I found the minimum year = 1927 and maximum year = 2016
Now using for loop I created a list ‘bin’ having years with the difference of 10 years.
Now using this ‘bins’ list I crated a new column ‘decade’ using the pd.cut() function to crate bins i.e. intervals of 10 years against each movie according to the ‘title_year’

Step7) sorted the data according to the ‘decade’

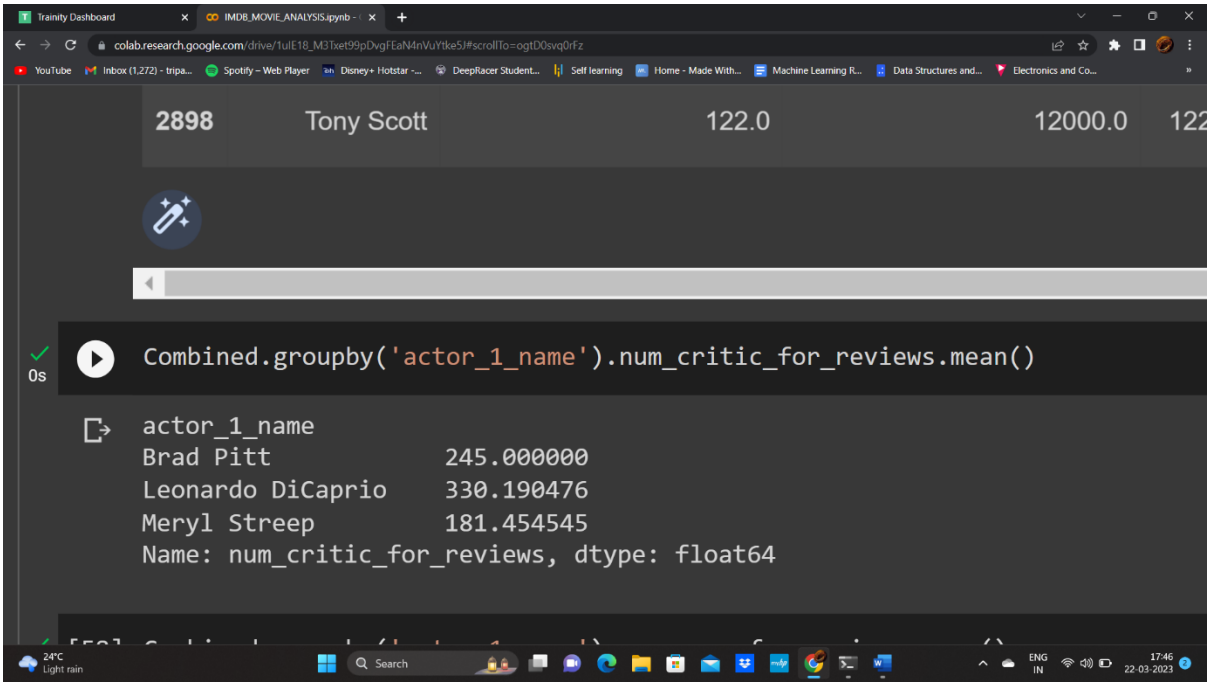
Step 8) created ‘df_by_decade’ having sum of ‘num_voted_users’ grouped by ‘decade’ and displayed the decade-wise sum of ‘num_voted_users

The ss shows the ‘Combined’ data created.



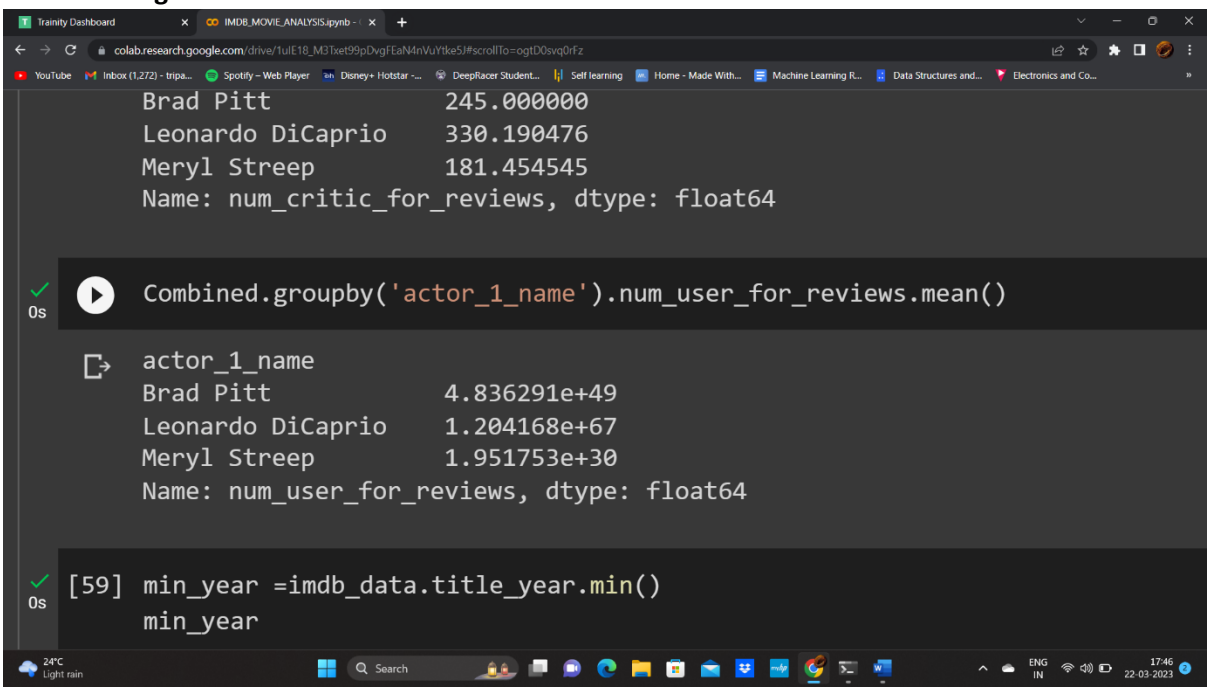
1408	David Frankel	208.0	64.0	124732962.0	Comedy|Drama|Romance	Meryl Streep	The Devil Wears Prada	286178
1483	Robert Redford	227.0	0.0	14998070.0	Drama|Thriller|War	Meryl Streep	Lions for Lambs	41170
1575	Sydney Pollack	66.0	521.0	87100000.0	Biography|Drama|Romance	Meryl Streep	Out of Africa	52339
1618	David Frankel	234.0	64.0	63536011.0	Comedy|Drama|Romance	Meryl Streep	Hope Springs	34258
1674	Carl Franklin	64.0	73.0	23209440.0	Drama	Meryl Streep	One True Thing	9283
1925	Stephen Daldry	174.0	335.0	41597830.0	Drama|Romance	Meryl Streep	The Hours	102123
2781	Phyllida Lloyd	331.0	58.0	29959436.0	Biography|Drama|History	Meryl Streep	The Iron Lady	82327
3135	Robert Altman	211.0	500.0	20338609.0	Comedy|Drama|Music	Meryl Streep	A Prairie Home Companion	19655
26	James Cameron	315.0	0.0	658672302.0	Drama|Romance	Leonardo DiCaprio	Titanic	793059
50	Baz Luhrmann	490.0	1000.0	144812796.0	Drama|Romance	Leonardo DiCaprio	The Great Gatsby	362912
97	Christopher Nolan	642.0	22000.0	292568851.0	Action|Adventure|Sci-Fi|Thriller	Leonardo DiCaprio	Inception	1468200

The following ss shows the data of critic mean data.

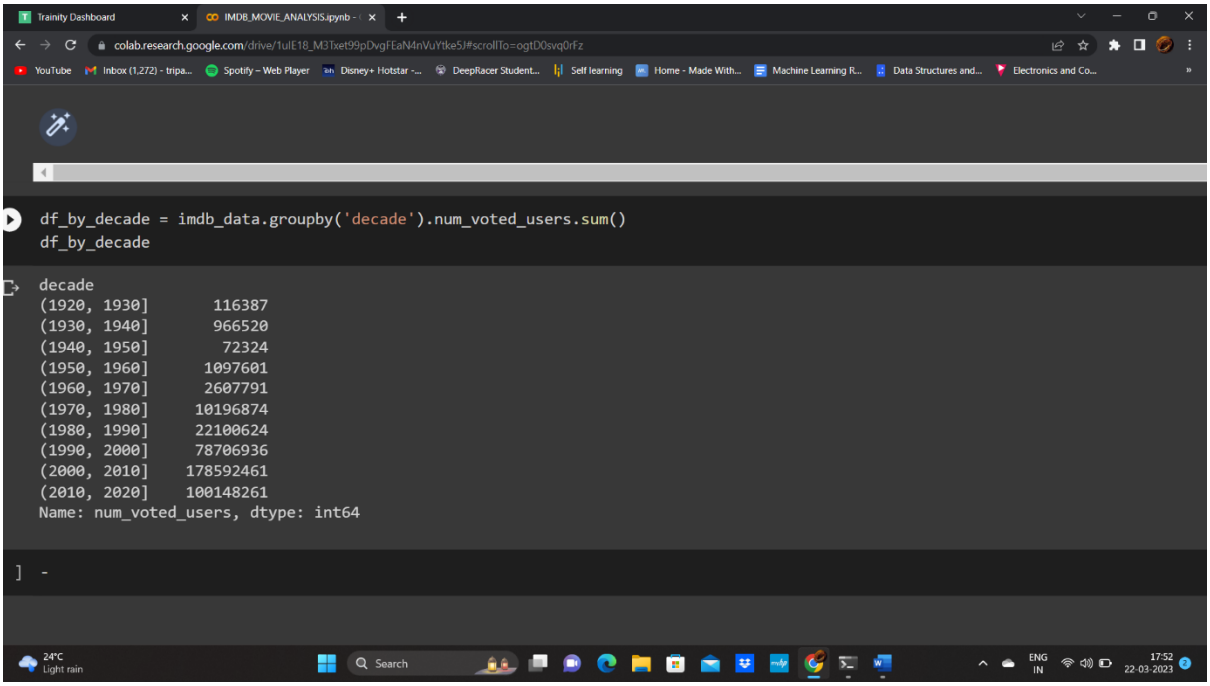


Answer: ***Leonardo DiCaprio***

The following ss shows the data for audience users.



The following ss shows the sum of voted users for each decade.



Result

Doing project involved numerous challenges which made me understand everything and work on it. Using python and its popular libraries for data drawing and analysis is what in which now I'm confident.