**IMDB Movie Analysis**

**Final Project-1**

**Author: Deeptendu Chiki**

**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?

Use these questions to guide your thinking:

What do you see happening?

What is your hypothesis for the cause of the problem? (this will be broadly based on intuition initially)

What is the impact of the problem on stakeholders?

What is the impact of the problem not being solved?

Answering these questions will help you define a problem you are trying to solve and will allow you to find the right data to solve it.

Once you have defined a problem, clean the data as necessary, and use your Data Analysis skills to explore the data set and derive insights.

Make sure to use 5 Whys Analysis in your analysis and use this to create a report which conveys a data story.

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?

Five 'Whys' approach

Once you have the problem better defined, you can use 5 Whys technique to determine its root cause by repeatedly asking the question “Why”.

It's also called the Root Cause Analysis, developed by Sakichi Toyoda, founder of Toyota Industries. Here’s an example of how this technique could be used to figure out the cause of the following problem: A business went over budget on a recent project.

Q: “Why did we go over budget on our project?”

A: It took much longer than we expected to complete.

Q: “Why did it take longer than expected to complete?”

A: We had to redesign several elements of the product.

Q: “Why did we have to redesign elements of the product?”

A: Features of the product were confusing to use.

Q: “Why were the features of the product confusing to use?”

A: We made incorrect assumptions about what users wanted.

Q: “Why did we make incorrect assumptions about what users wanted?”

A: Our user experience research team didn’t ask effective questions.

As you see above, what looked like a budgeting problem turned out to be a problem with the user experience team not working effectively.

While asking Why is easy, what we're interested in is the answer. Each time you answer why the next time gets more difficult as you must think deeper behind the reasons for this. As you ask why, you may find that you have multiple answers for the same question.

**Approach:**

Due to lack of time, we will be using Google Colab and with the help of python libraries such as numpy and pandas, solving this project will be much faster than MS. Excel.

Loaded the dataset in the colab platform and performed the necessary steps for Data Cleaning and EDA. Codes are efficiently and outcomes are given in the colab file which is mentioned in this document.

**Tech\_Stack Used:** Google Colab and Python.

**Insights:**

**Colab Link:**

<https://drive.google.com/file/d/1zoas_l2ldawkCeiTH5YI2kYoUDhz2Hyy/view?usp=sharing>

**Cleaning the data:** This is one of the most important step to perform before moving forward with the analysis. Use your knowledge learned till now to do this. (Dropping columns, removing null values, etc.)

**Your task:** Clean the data

**Solution:**

Cleaning the Data: Determine the total number of Null values in all columns and rows. Determine the percentage of Null values in each column as well. Round the percentages to the nearest two decimal places.

Drop unnecessary columns

For this project, we will mostly be analyzing the movies with respect to the ratings, gross collection, popularity of movies, etc. So many of the columns in this dataframe are not required. So, dropping the following columns.

color

director\_facebook\_likes

actor\_1\_facebook\_likes

actor\_2\_facebook\_likes

actor\_3\_facebook\_likes

actor\_2\_name

cast\_total\_facebook\_likes

actor\_3\_name

duration

facenumber\_in\_poster

content\_rating

country

movie\_imdb\_link

aspect\_ratio

plot\_keywords

Columns list after the drop:

Index (['director\_name', 'num\_critic\_for\_reviews', 'gross', 'genres',

'actor\_1\_name', 'movie\_title', 'num\_voted\_users',

'num\_user\_for\_reviews', 'language', 'budget', 'title\_year',

'imdb\_score', 'movie\_facebook\_likes'],

dtype='object')

**Observation:** On closer investigation, you may see that some columns have a high percentage (more than 5%) of Null values. Dropping all rows with Null values in those columns.

Gross and Budget has 17.53% and 9.76% of Null values. As a result, these are eliminated. Some rows may contain more than five NaN values. Such rows aren't useful for the analysis and should be eliminated. You may have noticed that some of the values in the language column are NaN. On closer examination, you will notice that it is safe to replace all of the missing values with.

You might notice that two of the columns viz. num\_critic\_for\_reviews and actor\_1\_name have small percentages of NaN values left. You can leave these columns alone for the time being. We still have roughly 77% of the rows after performing all of the tasks, according to the number and percentage of rows kept. As a result, Our Data Cleaning is complete.

**2. Movies with highest profit:** Create a new column called profit which contains the difference of the two columns: gross and budget. Sort the column using the profit column as reference. Plot profit (y-axis) vs budget (x- axis) and observe the outliers using the appropriate chart type.

**Your task:** Find the movies with the highest profit?

**Solution:**

Converting the budget and gross columns' units from $ to million $. Adding a new column named profit that holds the difference between the two columns: gross and budget.

Sort the dataframe with the profit column as a guide. Remove the top ten profitable movies in descending order and save them in a new dataframe - top10.

You may have noticed a duplicate value after discovering the top ten profitable movies. As a result, it appears that the dataframe contains duplicate values as well. Drop the duplicate values from the dataframe and execute the previous code again. Final output is given below:



**Observation:** Avatar, Avengers: Age of Ultron and The Dark Knight Rises are at the top.

**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. You can use your own imagination also!

**Your task:** Find IMDB Top 250

.

**Solution:**

Make a new dataframe called IMDb\_Top\_250 and fill it with the top 250 movies having the highest IMDb Rating (equivalent to the column: imdb\_score). Also, ensure that the num\_voted\_users for all of these movies is larger than 25,000. Add a Rank column with the numbers 1 to 250 reflecting the positions of the corresponding films.

Remove all movies from the IMDb\_Top\_250 dataframe that are not in English and place them in a new dataframe called Top\_Foreign\_Lang\_Film. Check the colab file link for the output mentioned in this document.

**4. Best Directors:** TGroup 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 top10director. In case of a tie in IMDb score between two directors, sort them alphabetically.

**Your task:** Find the best directors

**Solution:**

|  |  |
| --- | --- |
| **director\_name** | **imdb\_score** |
| Charles Chaplin | 8.6 |
| Tony Kaye | 8.6 |
| Alfred Hitchcock | 8.5 |
| Ron Fricke | 8.5 |
| Damien Chazelle | 8.5 |
| Majid Majidi | 8.5 |
| Sergio Leone | 8.433333 |
| Christopher Nolan | 8.425 |
| S.S. Rajamouli | 8.4 |
| Marius A. Markevicius | 8.4 |

**5. Popular Genres:** Perform this step using the knowledge gained while performing previous steps.

**Your task:** Find popular genres

**Solution:**

We have observed that the genres column in the dataframe, which has all of the movie genres separated by a pipe (|). The first two movie genres are the most important for every picture.

* Extract the first two genres from the genres column and store them in two new columns: genre\_1 and genre\_2. Some films may exclusively belong to one genre. In such circumstances, extract the single genre into both columns, i.e. for such movies the genre\_2 will be the same as genre\_1.
* Use genre 1 as the primary column and genre 2 as the secondary column to group the dataframe.
* Find the five most popular genre combinations by calculating the mean of the gross values in the gross column and storing them in a new dataframe called PopGenre.

|  |  |  |
| --- | --- | --- |
|  |  | **gross** |
| **genre\_1** | **genre\_2** |  |
| Family | Sci-Fi | 434.95 |
| Adventure | Sci-Fi | 228.62875 |
|  | Family | 118.918824 |
|  | Animation | 116.998462 |
| Action | Adventure | 109.59551 |

**Observation:** From the above data, we can conclude that Family + Sci-Fi is the most popular genre combination

**6. Charts:** Create 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, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.

Append the rows of all these columns and store them in a new column named Combined.

Group the combined column using the actor\_1\_name column.

Find the mean of the num\_critic\_for\_reviews and num\_users\_for\_review and identify the actors which have the highest mean.

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 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.

**Your task:** Find the critic-favorite and audience-favorite actors

**Solution:**

**actor\_1\_name num\_critic\_for\_reviews**

Brad Pitt 245.000000

Leonardo DiCaprio 330.190476

Meryl Streep 181.454545

**actor\_1\_name num\_user\_for\_reviews**

Brad Pitt 742.352941

Leonardo DiCaprio 914.476190

Meryl Streep 297.181818

**Observation:** Leonardo DiCaprio earns a perfect score on both lists.

**Results:**

What's particularly important for me when developing a solution for a project is to be absolutely clear on the goals from the beginning and then to create a strategy with milestones. I also try to start with the most difficult sections of the tasks so that if there are any major complications. I usually break down major jobs into smaller portions so that I know where to begin. To ensure that a major project runs smoothly, detailed planning is essential. And that is exactly what I learned and experienced after completion of this project.