**Predicting IMDb scores**

****

**Problem Definition**

**- \*\*Understanding the Issue\*\*: This involves thoroughly comprehending the problem at hand. It's not just about the symptoms, but the root cause as well.**

**- \*\*Stakeholder Analysis\*\*: Identifying and involving all relevant stakeholders who are affected by or have a stake in the problem.**

**- \*\*Setting Clear Objectives\*\*: Defining specific, measurable, achievable, relevant, and time-bound (SMART) objectives is essential. This gives a clear direction to the problem-solving process.**

**- \*\*Constraints and Resources\*\*: Recognizing any limitations, constraints, or resources available for addressing the problem.**

**- \*\*Problem Statement\*\*: This is a concise, clear description of the issue, highlighting what needs to be solved.**

**Design Thinking**

**- \*\*Empathy\*\*: Understanding the perspective and experiences of those affected by the problem. This involves observing, interviewing, and immersing oneself in the users' environment.**

**- \*\*Define\*\*: This step aligns with the problem definition phase. It involves synthesizing the collected information to create a user-centric problem statement.**

**- \*\*Ideate\*\*: Generating a wide range of possible solutions without judgment. This encourages creativity and out-of-the-box thinking.**

**- \*\*Prototype\*\*: Creating a tangible representation of the chosen solution, even if it's just a rough draft. This allows for testing and refining.**

**- \*\*Test\*\*: This involves putting the prototype in front of users and gathering feedback. It's a cycle of iteration until a satisfactory solution is reached.**

**These steps are often iterative, meaning you may need to revisit and refine your problem definition and solution ideas as you learn more through the design thinking process. Remember, the goal is to create solutions that truly meet the needs of the users or stakeholders involved.Background**

This dataset contains the information about the movies . For a movie to be commercial success , it depends on various factors like director, actors ,critic reviews and viewers reaction. Imdb score is one of the important factor to measure the movie's success.

**Description of dataset attributes**

Please find the details for the dataset attributes:-

1. Colour :- Movie is black or coloured
2. Director\_name:- Name of the movie director
3. num\_critic\_for\_reviews :- No of critics for the movie
4. duration:- movie duration in minutes
5. director\_facebook\_likes:-Number of likes for the Director on his Facebook Page
6. actor\_3\_facebook\_likes:- No of likes for the actor 3 on his/her facebook Page
7. actor2\_name:- name of the actor 2
8. actor\_1\_facebook\_likes:- No of likes for the actor 1 on his/her facebook Page
9. gross:- Gross earnings of the movie in Dollars
10. genres:- Film categorization like ‘Animation’, ‘Comedy’, ‘Romance’, ‘Horror’, ‘Sci-Fi’, ‘Action’, ‘Family’
11. actor\_1\_name:- Name of the actor 1
12. movie\_title:-Title of the movie
13. num\_voted\_users:-No of people who voted for the movie
14. cast\_total\_facebook\_likes:- Total facebook like for the movie
15. actor\_3\_name:- Name of the actor 3
16. facenumber\_in\_poster:- No of actors who featured in the movie poster
17. plot\_keywords:-Keywords describing the movie plots
18. movie\_imdb\_link:-Link of the movie link
19. num\_user\_for\_reviews:- Number of users who gave a review
20. language:- Language of the movie
21. country:- Country where movie is produced
22. content\_rating:- Content rating of the movie
23. budget:- Budget of the movie in Dollars
24. title\_year:- The year in which the movie is released
25. actor\_2\_facebook\_likes:- facebook likes for the actor 2
26. imdb\_score:- IMDB score of the movie
27. aspect\_ratio :- Aspect ratio the movie was made in
28. movie\_facebook\_likes:- Total no of facebook likes for the movie

**Case Study**

The dataset here gives the massive information about the movies and their IMDB scores respectively. We are going to analyze each and every factors which can influence the imdb ratings so that we can predict better results.The movie with the higher imdb score is more successful as compared to the movies with low imdb score.

**Data Preprocessing**

In [2]:

*#Reading the Data*

movie\_df=pd.read\_csv("/kaggle/input/imdb-5000-movie-dataset/movie\_metadata.csv")

In [3]:

*#Displaying the first 10 records*

movie\_df.head(10)

Out[3]:

|  | color | director\_name | num\_critic\_for\_reviews | duration | director\_facebook\_likes | actor\_3\_facebook\_likes | actor\_2\_name | actor\_1\_facebook\_likes | gross | genres | ... | num\_user\_for\_reviews | language | country | content\_rating | budget | title\_year | actor\_2\_facebook\_likes | imdb\_score | aspect\_ratio | movie\_facebook\_likes |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | Color | James Cameron | 723.0 | 178.0 | 0.0 | 855.0 | Joel David Moore | 1000.0 | 760505847.0 | Action|Adventure|Fantasy|Sci-Fi | ... | 3054.0 | English | USA | PG-13 | 237000000.0 | 2009.0 | 936.0 | 7.9 | 1.78 | 33000 |
| 1 | Color | Gore Verbinski | 302.0 | 169.0 | 563.0 | 1000.0 | Orlando Bloom | 40000.0 | 309404152.0 | Action|Adventure|Fantasy | ... | 1238.0 | English | USA | PG-13 | 300000000.0 | 2007.0 | 5000.0 | 7.1 | 2.35 | 0 |
| 2 | Color | Sam Mendes | 602.0 | 148.0 | 0.0 | 161.0 | Rory Kinnear | 11000.0 | 200074175.0 | Action|Adventure|Thriller | ... | 994.0 | English | UK | PG-13 | 245000000.0 | 2015.0 | 393.0 | 6.8 | 2.35 | 85000 |
| 3 | Color | Christopher Nolan | 813.0 | 164.0 | 22000.0 | 23000.0 | Christian Bale | 27000.0 | 448130642.0 | Action|Thriller | ... | 2701.0 | English | USA | PG-13 | 250000000.0 | 2012.0 | 23000.0 | 8.5 | 2.35 | 164000 |
| 4 | NaN | Doug Walker | NaN | NaN | 131.0 | NaN | Rob Walker | 131.0 | NaN | Documentary | ... | NaN | NaN | NaN | NaN | NaN | NaN | 12.0 | 7.1 | NaN | 0 |
| 5 | Color | Andrew Stanton | 462.0 | 132.0 | 475.0 | 530.0 | Samantha Morton | 640.0 | 73058679.0 | Action|Adventure|Sci-Fi | ... | 738.0 | English | USA | PG-13 | 263700000.0 | 2012.0 | 632.0 | 6.6 | 2.35 | 24000 |
| 6 | Color | Sam Raimi | 392.0 | 156.0 | 0.0 | 4000.0 | James Franco | 24000.0 | 336530303.0 | Action|Adventure|Romance | ... | 1902.0 | English | USA | PG-13 | 258000000.0 | 2007.0 | 11000.0 | 6.2 | 2.35 | 0 |
| 7 | Color | Nathan Greno | 324.0 | 100.0 | 15.0 | 284.0 | Donna Murphy | 799.0 | 200807262.0 | Adventure|Animation|Comedy|Family|Fantasy|Musi... | ... | 387.0 | English | USA | PG | 260000000.0 | 2010.0 | 553.0 | 7.8 | 1.85 | 29000 |
| 8 | Color | Joss Whedon | 635.0 | 141.0 | 0.0 | 19000.0 | Robert Downey Jr. | 26000.0 | 458991599.0 | Action|Adventure|Sci-Fi | ... | 1117.0 | English | USA | PG-13 | 250000000.0 | 2015.0 | 21000.0 | 7.5 | 2.35 | 118000 |
| 9 | Color | David Yates | 375.0 | 153.0 | 282.0 | 10000.0 | Daniel Radcliffe | 25000.0 | 301956980.0 | Adventure|Family|Fantasy|Mystery | ... | 973.0 | English | UK | PG | 250000000.0 | 2009.0 | 11000.0 | 7.5 | 2.35 | 10000 |

10 rows × 28 columns

In [4]:

*#Shape of the dataset (no of rows and no of columns)*

movie\_df.shape

Out[4]:

(5043, 28)

In [5]:

*#Displaying the data type of the dataset attributes*

movie\_df.dtypes

Out[5]:

color object

director\_name object

num\_critic\_for\_reviews float64

duration float64

director\_facebook\_likes float64

actor\_3\_facebook\_likes float64

actor\_2\_name object

actor\_1\_facebook\_likes float64

gross float64

genres object

actor\_1\_name object

movie\_title object

num\_voted\_users int64

cast\_total\_facebook\_likes int64

actor\_3\_name object

facenumber\_in\_poster float64

plot\_keywords object

movie\_imdb\_link object

num\_user\_for\_reviews float64

language object

country object

content\_rating object

budget float64

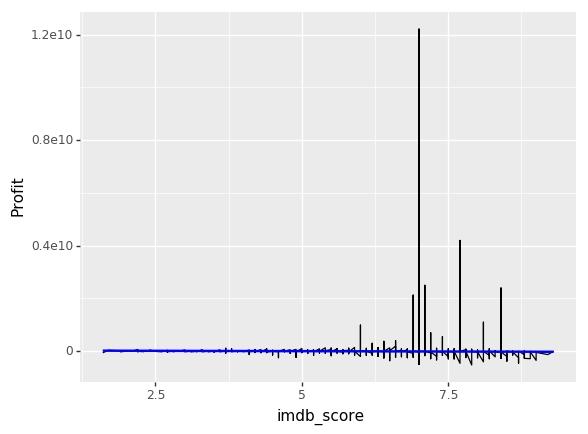
title\_year float64

actor\_2\_facebook\_likes float64

imdb\_score float64

aspect\_ratio float64

movie\_facebook\_likes int64

**Most of the movies above rating 8.75 are from USA**

In [33]:

*#Finding the corelation between imdb\_rating with respect to no of facebook likes*

(ggplot(movie\_df)

+ aes(x='imdb\_score', y='movie\_facebook\_likes')

+ geom\_line()

+ labs(title='IMDB\_Score vs. Facebook like for Movies', x='IMDB scores', y='Facebook Likes for movies')

)

**Movie with high IMDB rating have most no of facebook likes**

In [34]:

*#Top 20 movies based on the profit they made*

plt.figure(figsize=(10,8))

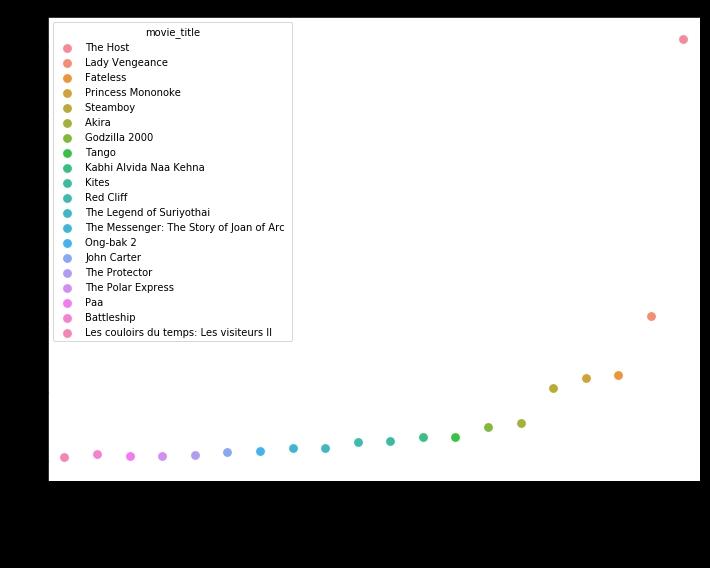
movie\_df= movie\_df.sort\_values(by ='Profit' , ascending=False)

movie\_df\_new=movie\_df.head(20)

ax=sns.pointplot(movie\_df\_new['Profit'], movie\_df\_new['budget'], hue=movie\_df\_new['movie\_title'])

ax.set\_xticklabels(ax.get\_xticklabels(), rotation=40, ha="right")

plt.tight\_layout()

****plt.show()



**Data acquisition & cleaning**

As a quick note, IMDb has an API available to download bulk data, but a primary requirement for this project was to obtain data through web scraping; so, I went along and got the information from IMDb using requests and Beautiful Soup. Requests is the module required to take the webpage and turn it into an object in python. Beautiful Soup takes that object, which is the HTML information behind the webpage, and makes searching and accessing specific information within the HTML text easy. You really need both in order to fully complete the process of web scraping.

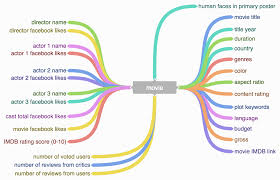
On the IMDb page, I used the advanced search feature to access titles between 2000 and 2020. The results spanned thousands of pages and each page held the titles and links to 100 movies. Upon further inspection, I noticed the URL contained the phrase: ‘start=1’. Increasing this start number by 100 would flip through each page. With a helper function, I used requests and Beautiful Soup to pull the links for each page and returned a list of those links.

To utilize that list of movie hyperlinks, I created another function to extract as much data as I could from each page. This function took in a link and returned a dictionary containing the following information: title, IMDb rating, the number of IMDb raters, MPAA rating, genres, directors, writers, top three stars, initial country of the release, original language of the release, release date, budget, opening weekend USA, gross USA, cumulative worldwide gross, production companies, and runtime.

As part of the EDA, some data had to be cleaned. This consisted of turning any numerical value from a string into an integer. Runtime had to be converted into minutes, all of the monetary values needed commas and dollar signs removed, and the release date had to be converted into datetime. Additionally, categories that contained lists needed to be converted from strings into actual python lists (genres, directors, stars, production companies). The retrieval function did most of this cleaning, but after putting the data into a DataFrame, some other cleaning was necessary.

With over 2,000 movies in a DataFrame, I needed to do some more processing to get a functional DataFrame for modeling. This meant dropping movies without information on budget, movies with a budget below $1,000, and movies with a sum of raters under 1,500. In regards to that last requirement, movies with a low number of raters proved to report the more extreme movie ratings (movies leaning towards a perfect 10 or a big goose egg). All in all, I ended up with a DataFrame consisting of over 1,100 movies. Now it’s time to start modeling.

Pairplots: Before moving on to the next section, I’d like to mention pairplots. Pairplots is a great visualization tool for exploring relationships within the data and informing where to start for an MVP. It seems like a lot of information, but when you format your DataFrame with the first or last column being the target, it is a lot easier to interpret all of this information. For this pairplot, the plots in the first column show relationships between the independent variables and the target. Although I did not use most of the numerical data, it is obvious that there are linear and exponential relationships, which can easily inform where to start modeling

****