

CS432/532: Final Project Report

Project Title: IMDB Data Analysis

Team Member(s): Rohan Ravindra Saraf and Satrio Baskoro Yudhoatmojo

I. PROBLEM

IMDb is the world most popular sites for movies, TV and celebrity content which has been around since 1990 [1]. There are several datasets already curated by people that we can use for this project. We proposed to do an explorative analysis on the curated IMDb dataset. Based on our naïve search of the IMDb curated dataset, we found that two IMDb datasets [2, 3]. We decided to use the dataset from [3] for our project because it consists more attributes and the number data is larger than [2].

We propose to analyze the following problems. First, we analyze the top 10 movies genre on each year. Second, we explore the correlation between movie rating and movie revenue. Third, derive the conclusion about the profit-loss for the movie based on revenue. Also we worked to study the increasing and decreasing trends in the genres of the movies.

II. SOFTWARE DESIGN AND IMPLEMENTATION

Following are the details about the Software design, NoSQL – Database, and the tools that we have used in the developing the Project – 3.

A. Software Design and NoSQL-Database and Tools Used

We propose to store the curated dataset in MongoDB DBMS. We would preprocess the curated dataset in a way so that it is a good fit to store it in MongoDB.

In the application layer, we will use Python programming language for performing the analysis stated in Section I. Python programming language will be used to retrieve data from MongoDB and transform the data into the needed format for the analysis.

After retrieving the data using the python, we have used the HTML, CSS and JavaScript to represent the data in user friendly manner with the proper GUI. Also the top layer of the HTML, gives user the provision to give the input to the application by either selecting year, selecting genre, sliding the lower and upper bound of the IMDB Score etc.

To wrap the whole project, we will wrap the project in the form of web-based application. We may use micro-framework Flask as the scaffolding for developing this web-based application.

Also on top of all this we have used Microsoft Excel to clean the data from the Null and invalid values. We have done following things to clean and purify the data:

- For the Null values in the column with Number as Data type, we have filled it with Average of the column.
- For the Columns of Revenue, we have generated the random number between the specific ranges to fill out null values.
- Use of Excel formulas to trim and remove the

unnecessary blanks spaces in between the words and at the end.

- For null values in the columns with the string datatype we have replace it with the dummy string.

B. Data Import Process in the MongoDB Database

As mentioned earlier, we have used the dataset [3], so we have imported it using a following command into the MongoDB Database.

Command:

```
mongoimport --type csv -d (DB Name) -c (Collection Name) --headerline --drop (Filename/path to File)
```

Here in our case we have given “**movies**” as the database name and “**movies**” as the collection name.

C. Libraries for Animation and Chart/Graphs

We have used the libraries of the animation and displaying the chart/graphs.

1. **Animation:** Wow.min.js and css → we need give the respective class to the HTML div and also give the time ins ‘Microseconds’ for the animation and include the script in the page.
2. **Graphs:** Chart.min.js and css → we pass the finalized data to the library and library help us to generated the graphs.

D. Supported Queries

All the operations are the Selection Operations to get the details from the Project. We have some of the basic queries along with the sophisticated queries which mainly use to derive the conclusion using either a table or a pictorial graph.

The following are supported queries and functionalities that we have implemented in the Project:

1. List of movies yearly (basic query)

This query help us to list the all movies for the selected year in the tabular format. We take the “**year**” as an input from the user via a drop down menu. After he clicks submit, we fire a select query in the MongoDB Database to fetch all the movies released in that particular year.

LIST OF MOVIES YEARLY

List all movies released on user selected year in the tabular format.

Choose movie release year: 1916 * | Submit

List of Movies in Year 1935

No.	Movie Title	Director Name	Content Rating	Duration	Language	Country	Genres	IMDB Score
1	Top Hat	Mark Sandrich	Approved	81	English	USA	Comedy Musical Romance	7.8

Fig. List of movies for the selected year

2. List of movies based on genre (basic query)

This query help us to list the all movies for the selected year and selected genre in the tabular format. We take the “year” and “genre” as an input from the user via a drop down menu. After he clicks submit, we fire a select query in the MongoDB database to fetch all the movies released in that particular year and search movies in that year based on genre.

LIST OF MOVIES BASED ON GENRE
List all movies released on user selected year with user selected genre in a tabular format.

Choose movie release year: 1916 • Choose movie genres: Action • Submit

List of Movies in Year 1936 with the Genre of Action

No.	Movie Title	Director Name	Content Rating	Duration	Language	Country	Genres	IMDB Score
1	The Charge of the Light Brigade	Michael Curtiz	Approved	100	English	USA	Action Adventure Romance War	7.1

Fig. Movies for the selected year and genre

3. List of actors and number of movies starred by the actor (basic query)

This query help us to list the all actors and number of movies they starred in the tabular format. Here we fire a select query in the MongoDB database to fetch all the actors from the dataset, we have 3 columns for the name of actors as there are multiple actors in a single movie. Firstly we get all the actors and then we count the number of movies they starred in to get the actual count.

LIST OF ACTORS AND THE NUMBER OF MOVIES STARRED BY THE ACTOR
List all the actor names and number of movies they have starred in as per current dataset.

List of Actors and The Number of Movies They Starred in

No.	Actor Name	Number of Movies
1	5D Cent	5
2	A. Michael Baldwin	1
3	A.J. Buckley	5
4	A.J. DeLuca	1
5	A.J. Langer	1
6	A.J. Michalko	3
7	Aaliyah	2
8	Aaron Ashmore	2
9	Aaron Hill	1
10	Aaron Hughes	1
11	Aaron Kwok	1
12	Aaron Stanford	4
13	Aaron Staton	1
14	Aaron Yoo	6
15	Aashishkai Rathija	1
16	Audi Marvoti	4
17	Adobe Correll	8
18	Aditya Elbert	1
19	Aditya Mukherjee	1
20	Aditya Singh	3
21	Aditya Singh	1
22	Aditya Singh	4
23	Abraham Benrubi	3
24	Ace Manero	1
25	Adam Abou Madi	1
26	Adam Arkin	4
27	Adam Baldwin	6
28	Adam Boyer	1

Fig. Name of Actors and count of movies

4. Visualizing the top 10 movie genre on each year (sophisticated query)

This query help us to visualize the top 10 movies for the selected year in the graphical format. We take the “year” as an input from the user via a drop down menu. After he clicks submit, we fire a select query in the MongoDB database to fetch all the movies released in that particular year and select the only top 10 of them and give a graph in a user friendly format. We select the top 10 movies based on the IMDB score that we have for the movie in the dataset.

VISUALIZING THE TOP 10 MOVIES ON SPECIFIC YEAR

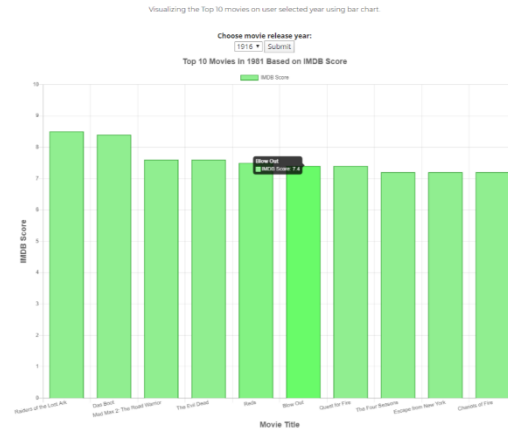


Fig. Top 10 movies based on IMDB Score for that year

5. Correlation analysis between movie rating and movie revenue (sophisticated query).

This query help us to conclude or derive a relation for movie about how much has earned as compared to the IMDB rating given by the people. It happens that the movie has not collected much revenue but based on the story and script makes the movie popular between the compelling them to give high IMDB Score. Vice Versa is also possible that movie collected a lot revenue to popularity of its actors but the actual story, script and other things are so-so, normal so it has low IMDB Score. Here we take the “year” as the input from the year so generate the scenario for that particular year.

CORRELATION ANALYSIS BETWEEN MOVIE RATING AND MOVIE REVENUE

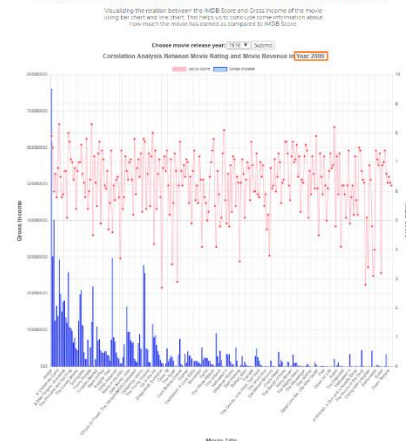


Fig. Correlation between the income and IMDB Score

6. Number of profit-loss movie revenue in each year (sophisticated query)

This query compares the budget of movies and the gross income made by movie. We took a very coarse average of budgets and gross incomes. We averaged all budgets and gross income of all movies in each year. The result is displayed using bar chart. By using this visualization, we can which years the average budget is greater or lesser than what the movie received as the gross income.

NUMBER OF PROFIT-LOSS MOVIE REVENUE IN EACH YEAR

Visualizing the relation between how much money was spent in making it and how much did it earned using bar chart?

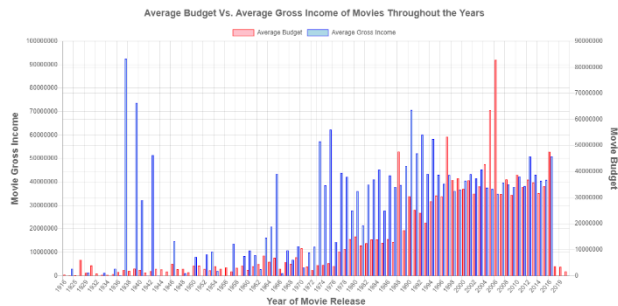


Fig. Visualizing the Profit-Loss of movies in different year.

7. Movies to watch based on ratings (basic query)

This query help us to list the movies they between the given range of the IMDB Score in the tabular format. Here we fire a select query in the MongoDB database to fetch all those movies. This query helps the users to see movies, which other people have rated high. So he can filter the movies and choose the movie of his choice to watch. Also we take the “year” as an input so that we can filter the movies as per the year and then from those set displaying only those IMDB score is in between selected ones.

MOVIES TO WATCH BASED ON RATINGS

Listing the Movies in a tabular format based on the range of IMDB Score selected by the user.

Choose movie release year: 1916

Lower bound rating: 2.6

Upper bound rating: 8.4

Submit

List of Movies in Year 1939
with Rating between 2.6 and 8.4

No.	Movie Title	Director Name	Content Rating	Duration	Language	Country	Genres	IMDB Score
1	Gone with the Wind	Victor Fleming	G	226	English	USA	Drama History Romance War	8.2
2	The Wizard of Oz	Victor Fleming	Passed	102	English	USA	Adventure Family Fantasy Musical	8.1
3	Mr. Smith Goes to Washington	Frank Capra	Not Rated	120	English	USA	Comedy Drama	8.2

Fig. Movies for the selected rating and year

8. Increasing/Decreasing trend of the movie genres for different years (sophisticated query)

We have generated the line graph for this query where we are able to see all the genres with different colors and their increasing/decreasing trends over the year. Like for e.g. If we consider the genre as crime, in the early making the movie on this genre was less, then it directors started making movies based on this genre, so graph started going up. Recently again the graph started coming down. So this way we can get the clear idea of trend of different genres in the passing years.

INCREASING/DECREASING TREND OF MOVIE GENRES FOR DIFFERENT YEARS

Showing the trends of the particular movie genre in the particular years. This helps to conclude increasing or decreasing trend of a particular genre in the film industry as the years pass by.

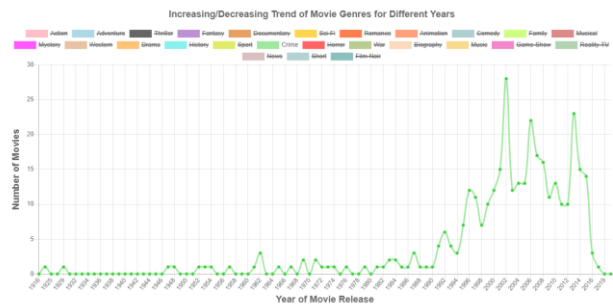


Fig. Trend for genre “Crime”

INCREASING/DECREASING TREND OF MOVIE GENRES FOR DIFFERENT YEARS

Showing the trends of the particular movie genre in the particular years. This helps to conclude increasing or decreasing trend of a particular genre in the film industry as the years pass by.

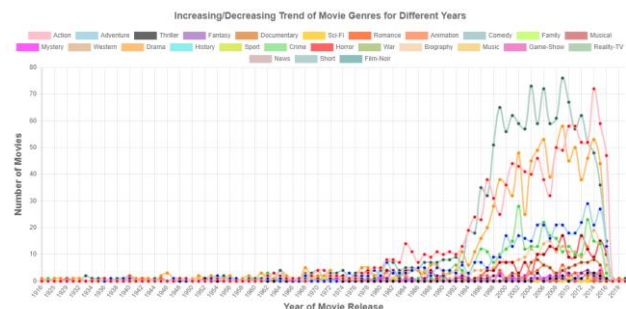


Fig. Trend for all genres

REFERENCES

- [1] IMDB. “What Is IMDb?” *IMDb*, IMDb.com, help.imdb.com/article/imdb/general-information/what-is-imdb/G836CY29Z4SGNMK5?ref_=helpnav_1#.
- [2] Promptcloud. “IMDB Data from 2006 to 2016 - Dataset by Promptcloud.” *Data.world*, 26 June 2017, <https://data.world/promptcloud/imdb-data-from-2006-to-2016>.
- [3] Yueming. “IMDB 5000 Movie Dataset.” *Kaggle*, 16 Dec. 2017, www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset.
- [4] Flask official Documentation <https://flask.palletsprojects.com/en/1.1.x/>
- [5] HTML Documentation and tutorial from <https://www.w3schools.com/html/>
- [6] CSS and Bootstrap from <https://getbootstrap.com/>
- [7] JQuery from <https://jqueryui.com/>
- [8] Referring fonts from <https://fonts.google.com/>
- [9] Animation from <https://wowjs.uk/docs>
- [10]
- [11] Searching slider and favicon from <https://www.google.com/imgph?hl=en>