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 data is larger than [2].

We propose to analyze the following problems. First, we analyze the top 10 movies 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 Project – 3.

A. Software Design and NoSQL-Database and Tools Used

We propose to store the curated dataset in MongoDB database. 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 used 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 HTML, CSS and JavaScript to represent the data in user friendly manner with 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 for the IMDb Score range etc.

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

On top of all this we used Microsoft Excel to clean the data which had some Null and Invalid values. Following things are done 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 consisting revenue related data, we have generated the random number between the specific ranges to fill out null values.
- Used 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 string as a datatype we have replaced it with the dummy string.

B. Data Import Process in the MongoDB Database

As mentioned earlier, we have used the dataset [3] for the project. Now to first thing we did is, imported dataset into MongoDB using a following command.

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 following libraries for animation and displaying chart/graphs.

1. **Animation:** Wow.min.js and css → we need give the respective classes to the HTML tag and also give the time in 'Microseconds' for the animation and include the script on the page.
2. **Graphs:** Chart.min.js and css → we pass the finalized data to library and it helps us to generate graphs.

D. Supported Queries

All the operations in this application are Selection Operations which gets 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 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.

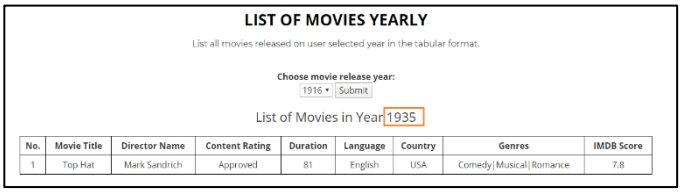


Fig.1 List of movies for the selected year

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 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.

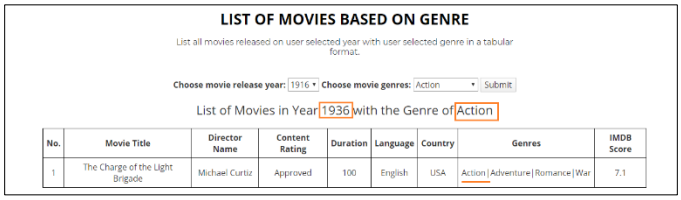
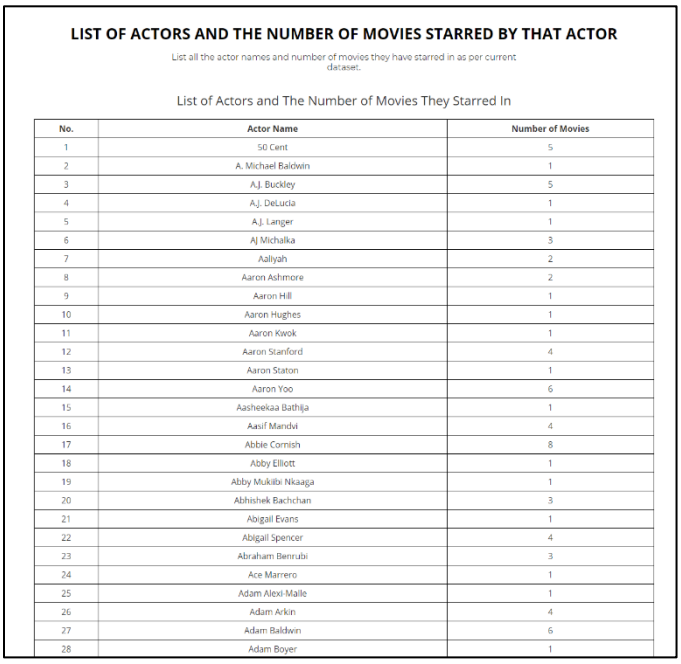


Fig.2 Movies for the selected year and genre

3. List of actors and number of movies starred by that 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.



No.	Actor Name	Number of Movies
1	50 Cent	5
2	A. Michael Baldwin	1
3	AJ. Buckley	5
4	A.J. DeLucia	1
5	A.J. Langer	1
6	AJ Michalka	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	Aasheekaa Bathija	1
16	Aasif Mandvi	4
17	Abbie Cornish	8
18	Abby Elliott	1
19	Abby Mukiibi Nkaaga	1
20	Abhishek Bachchan	3
21	Abigail Evans	1
22	Abigail Spencer	4
23	Abraham Benrubi	3
24	Ace Marrero	1
25	Adam Alexi-Malle	1
26	Adam Arkin	4
27	Adam Baldwin	6
28	Adam Boyer	1

Fig.3 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 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.

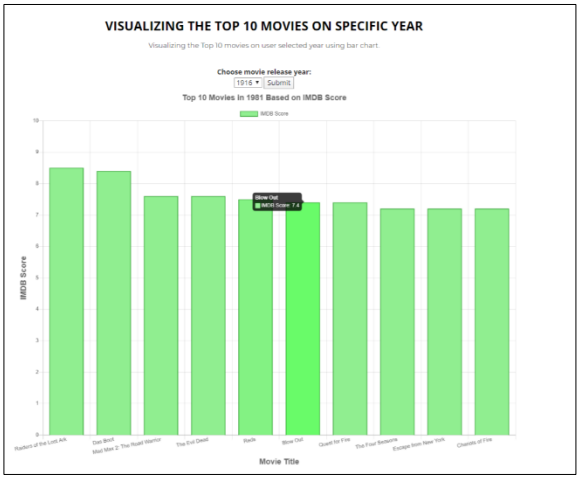


Fig.4 Top 10 movies based on IMDb Score for that year

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 it has earned as compared to the IMDb rating given by the people. It happens that the movie has not collected much revenue but the story and script makes the movie popular between the people compelling them to give a high IMDb Score. Vice Versa is also possible that movie collected a lot revenue due 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.

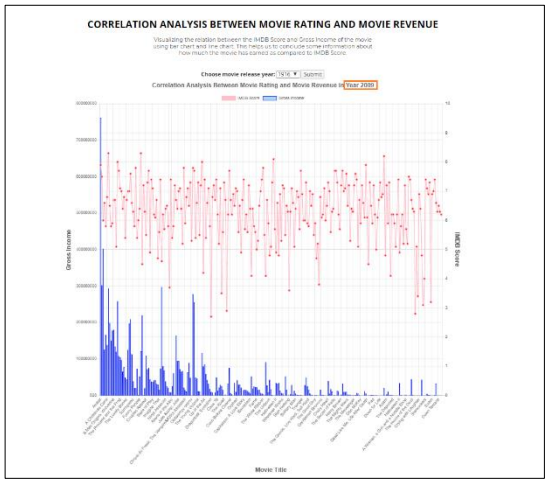


Fig.5 Correlation between the income and IMDb Score

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 decide which years the average budget is greater or lesser than what the movie received as the gross income.

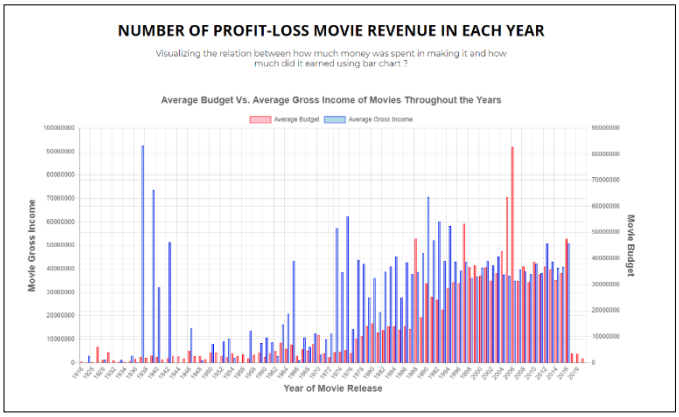
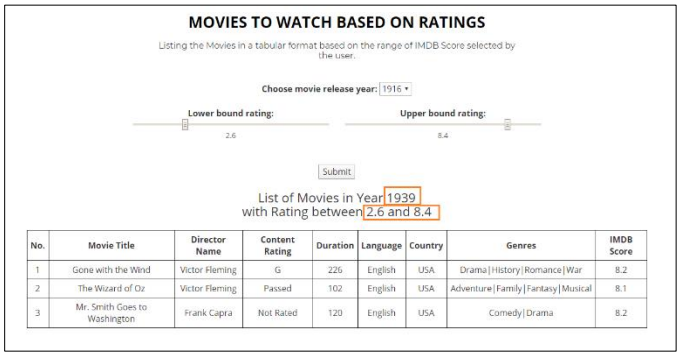


Fig.7 Visualizing the Profit-Loss of movies in different year

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

This query help us to list the movies between the selected ranges 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 user 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 whose IMDb score is in between selected ones.



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.8 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 years, making the movie on this genre was less, then 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.

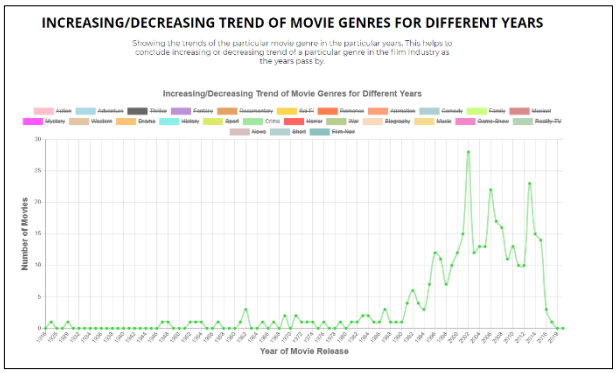


*Fig.9 Trend for genre "**Crime**"*

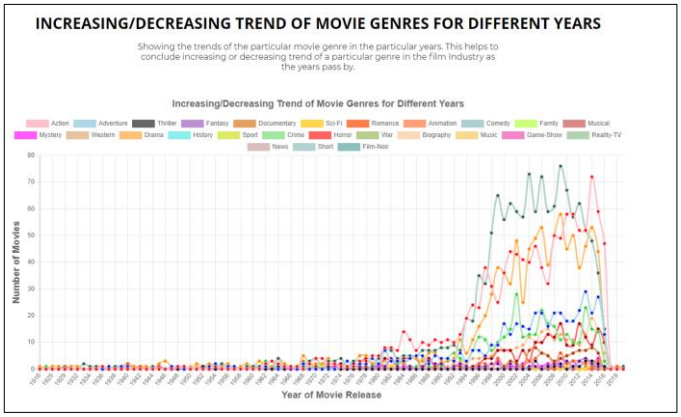


Fig.10 Trend for all genres

III. PROJECT OUTCOMES

- GitHub URL to the source code of Project:

  https://github.com/satrio-yudhoatmojo/CS532-Database-Systems-Spring-2020.git

- Link to the YouTube Video:

  https://youtu.be/xNyx5_ZKYxY

- Link to the Presentation Slides:

  https://drive.google.com/file/d/1cMcnUi4TPCDmNj8cD4f9JtV04k0M9mz7/view?usp=sharing

REFERENCES

[1] IMDB. "What Is IMDb?" *IMDb*, IMDb.com, help.imdb.com/article/imdb/general-information/what-is-imdb/G836CY29Z4SGNMK5?ref_=helpart_nav_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: https://www.w3schools.com/html/

[6] CSS and Bootstrap: https://getbootstrap.com/

[7] JQuery: https://jqueryui.com/

[8] Fonts: https://fonts.google.com/

[9] Animation: https://wowjs.uk/docs

[10] Charts: https://www.chartjs.org/

[11] Images of slider and favicon: https://www.google.com/imghp?hl=en