IMDB Data Analysis

Rohan Ravindra Saraf (B00804752)

Satrio Baskoro Yudhoatmojo (B00818460)

Problem Statement:

- IMDb is the world most popular sites for movies, TV and celebrity content which has been around since 1990, but being a huge website it difficult to know various details about the movies, celebrities and different correlations in between them.
- This application aims to give better analysis and experience to the users by giving the curated analysis about the different movies based on their corresponding IMDB scores, revenue generated and also gives a broader view to the user to see what movies they can watch and what is its statistics.
- Performing all this analysis on such a huge data might be difficult using the structured DB but this is somewhat easy by using the NoSQL-DB.
- Reading all the analysis in just textual format is also somewhat tedious and time consuming. Hence we have also designed the line and bar graphs to visualize the data in a better way.

Software Design/Technologies and Tools Used:

- **Database:** We have used the NoSQL DB "***MongoDB***" to store all the data. It is basically an open source, document based database and also it provides the support to store unstructured data and that can be retrieved in different forms.
- For our project we have stored the data in a database named "***movies***" and inside the database there is a collection named "***movies***".
- **Frontend:** After the data being retrieved in python, we have displayed the data in a user friendly manner using HTML, CSS and JavaScript. 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 range etc.
- **Backend:** We have used Python programming language for performing the analysis stated in the proposal. Python programming language will be used to retrieve data from MongoDB and transform the data into the needed format for the analysis.
- **Middleware:** To wrap the whole project and create a connection we have used micro-framework "**Flask**" as the scaffolding for developing the web-based application.

Data Cleaning Process:

We have done following things to clean and purify the data so that all the results are proper and valid:

- For the Null values in the column with number as datatype, we have filled it with average of that column.
- For the columns of consisting of data related to revenue, we have generated the random number between the specific ranges to fill out null values.
- Excel formulas are used to trim and remove the unnecessary blanks spaces in between the words and at the end of the word.
- For null values in the columns with the string datatype we have replaced it with dummy string.

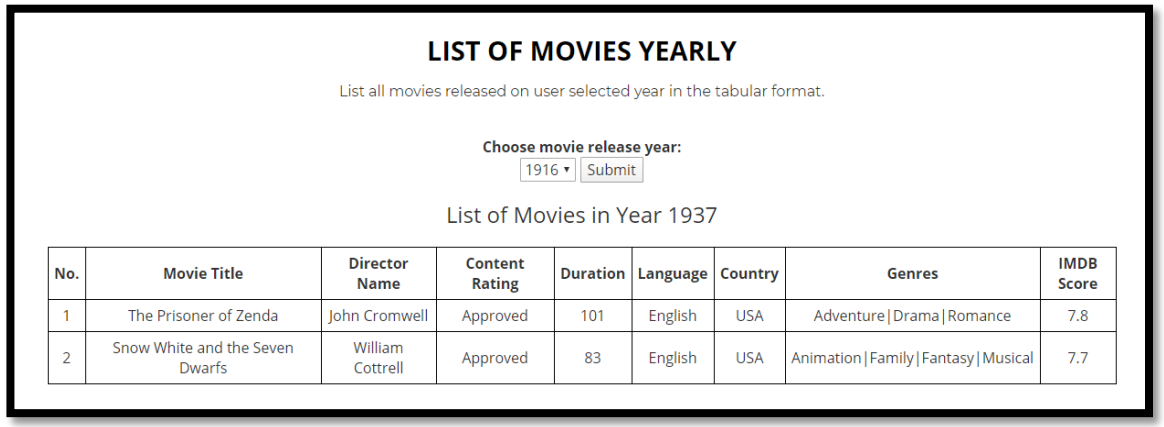
Operations/Queries Supported:

- All the queries are “select” queries, we fire the queries to the MongoDB Dataset, retrieve the data and we then represent it in the format as needed.
- We simply need to pass the parameter as an input like:
  - Year → Year of the movie released
  - Genre → Movies from the specific year of specific genre.
  - Range slider → Slider to give the lower and upper bound of the IMDB Score.
- The search or select operation is facilitated by the mongo query ‘**find**’ in a MongoDB.
- Following slides describe the queries supported by this application

Query 1:
List of movies yearly

- 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 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.
- **Query:**

```
db.movies.find({'title_year': int(year)},
{'_id': 0, 'movie_title': 1, 'director_name': 1,
'content_rating': 1, 'duration': 1, 'language': 1, 'country': 1,
'genres': 1, 'imdb_score': 1})
```



LIST OF MOVIES YEARLY

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

Choose movie release year:

1916 ▾ Submit

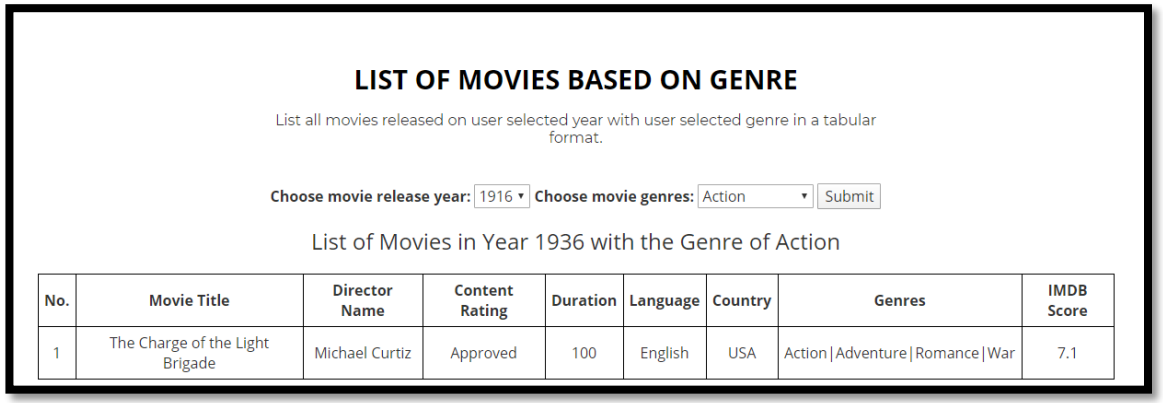
List of Movies in Year 1937

No.	Movie Title	Director Name	Content Rating	Duration	Language	Country	Genres	IMDB Score
1	The Prisoner of Zenda	John Cromwell	Approved	101	English	USA	Adventure|Drama|Romance	7.8
2	Snow White and the Seven Dwarfs	William Cottrell	Approved	83	English	USA	Animation|Family|Fantasy|Musical	7.7

Query 2:
List of movies based on genre

- 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 filter the movies in that year based on genre.
- **Query:**

```
db.movies.find({"genres": {"$regex": reg_exp}, "title_year": int(year)}).sort("movie_title")
```



Query 3:
List of actors and number of movies starred by that actor

- This query help us to list the all actors and get the count of number of movies they have starred, in the tabular format.
- 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.
- Firstly, we get all the actors and then we count the number of movies they starred in to get the actual count.
- **Query:**

```
db.movies.find({}, {'_id': 0, 'actor_1_name': 1,
'actor_2_name': 1, 'actor_3_name': 1})


db.movies.find({'$or':[{'actor_1_name':person},
{'actor_2_name': person}, {'actor_3_name': person}]},
{'_id': 0, 'movie_title': 1}).count()
```

LIST OF ACTORS AND THE NUMBER OF MOVIES STARRED BY THAT ACTOR

List all the actor names and number of movies they have starred in as per current dataset.

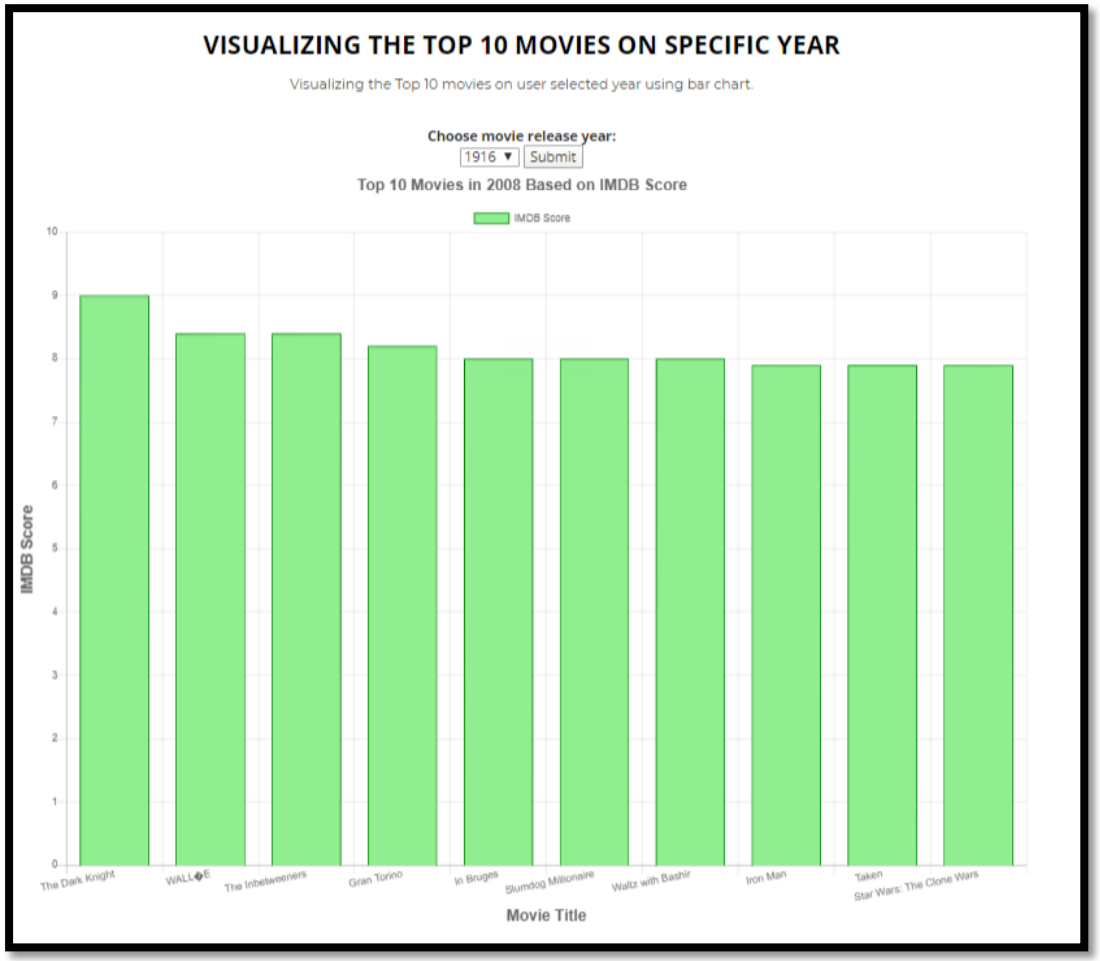
List of Actors and The Number of Movies They Starred In

No.	Actor Name	Number of Movies
1	50 Cent	5
2	A. Michael Baldwin	1
3	A.J. 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

Query 4:
Visualizing the top 10 movie genre on each year

- With the help of this query we 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.
- 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.
- **Query:**

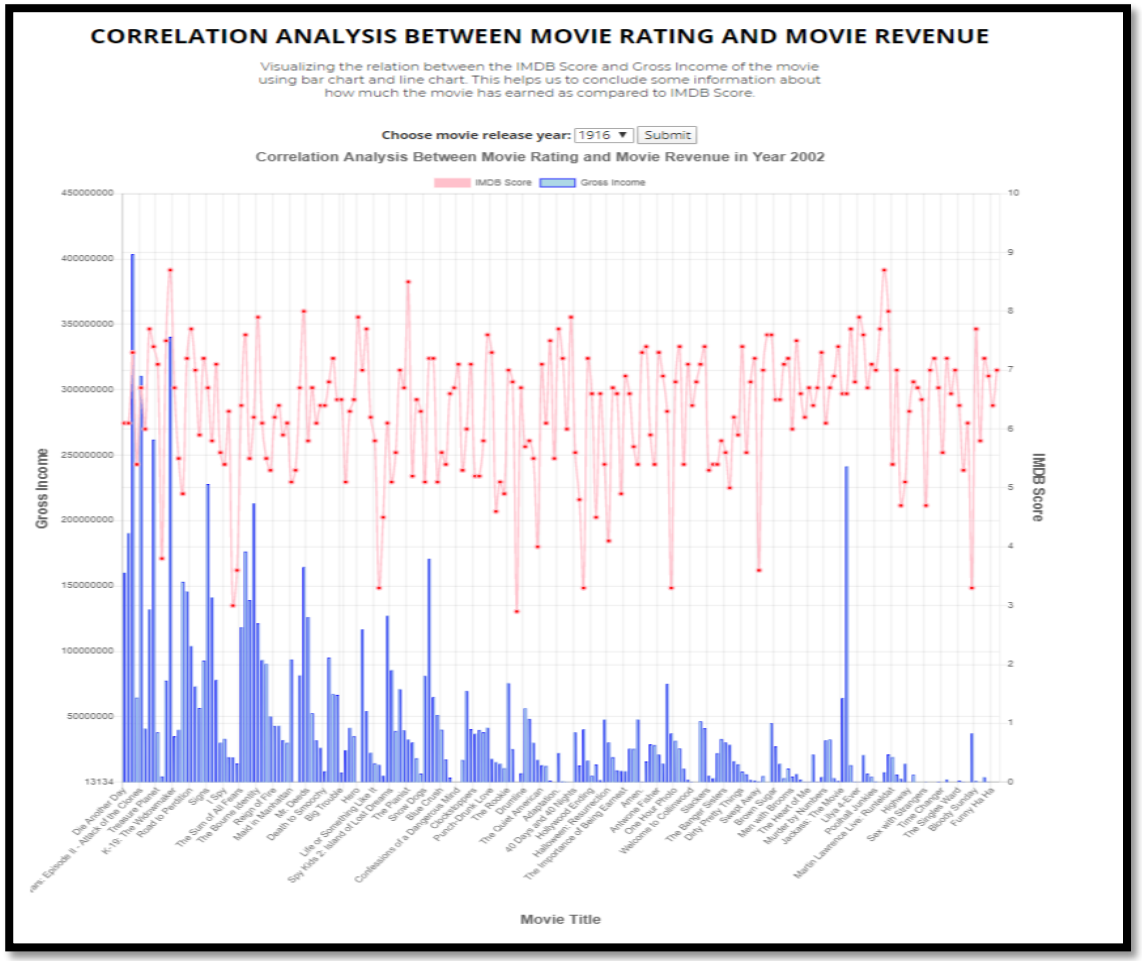
```
db.movies.find({"title_year": int(year)}).sort("imdb_score", -1).limit(10)
```



Query 5:
Correlation analysis between movie rating and movie revenue

- 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 the movie got popular between the people compelling them to give high IMDb Score.
- It is also possible that movie collected a lot of 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 user and generate the scenario for that particular year.
- **Query**:

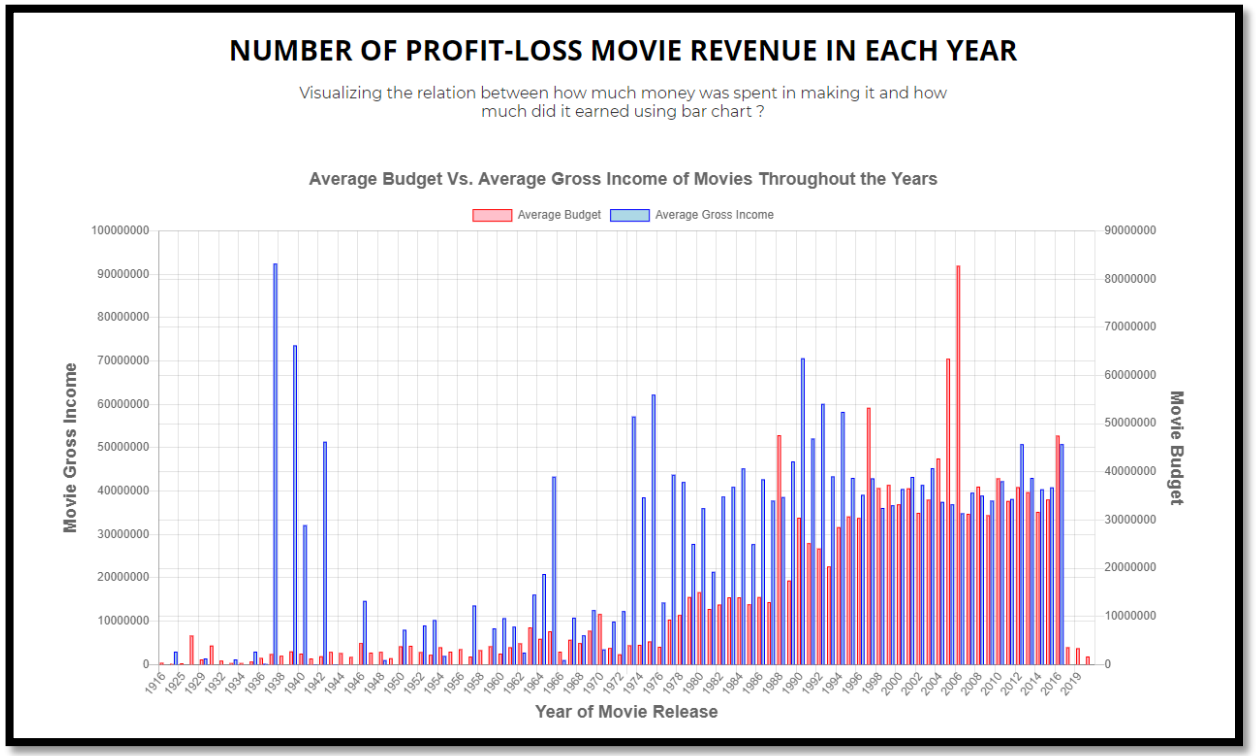
db.movies.find({'title_year': int(year)})



Query 6:
Number of profit-loss movie revenue in each year

- This query compares the budget of movies and the gross income made by movies.
- 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 find that in which years the average budget is greater or lesser than what the movies received as the gross income.
- **Query:**

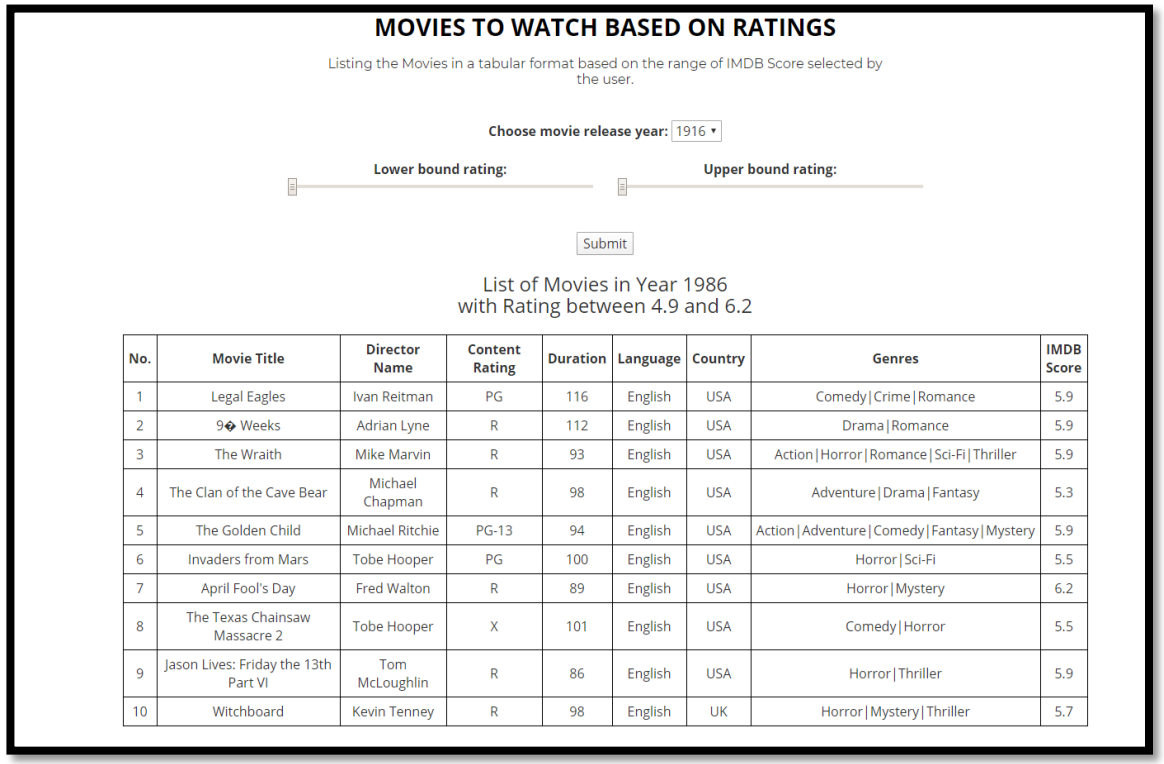
```
db.movies.aggregate([{"$group":{"_id":"$title_year",
"avg_budget": {"$avg": "$budget"}, "avg_gross": {"$avg":
"$gross"}}}, {"$sort": {"_id": 1}}])
```



Query 7:
Movies to watch based on ratings

- This query help us to list the movies whose IMDb score lies in between the selected range in the tabular format.
- 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.
- We take the "**year**" as an input so that we can filter the movies as per the year and then from those set of movies we display only those, IMDb score is in between selected range.
- **Query:**

```
db.movies.find({"imdb_score": {"$gte": float(low_range),
"$lte": float(high_range)}, "title_year": int(year)})
```



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: **Upper bound rating:**

Submit

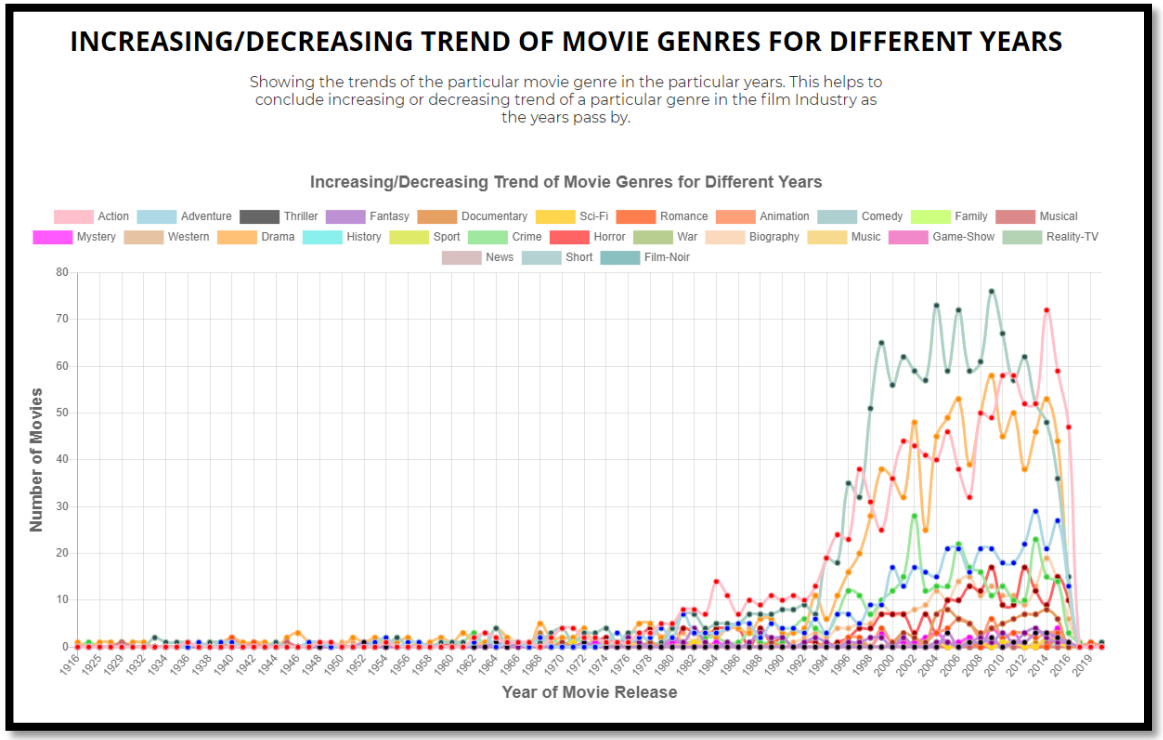
List of Movies in Year 1986
with Rating between 4.9 and 6.2

No.	Movie Title	Director Name	Content Rating	Duration	Language	Country	Genres	IMDB Score
1	Legal Eagles	Ivan Reitman	PG	116	English	USA	Comedy|Crime|Romance	5.9
2	9 Weeks	Adrian Lyne	R	112	English	USA	Drama|Romance	5.9
3	The Wraith	Mike Marvin	R	93	English	USA	Action|Horror|Romance|Sci-Fi|Thriller	5.9
4	The Clan of the Cave Bear	Michael Chapman	R	98	English	USA	Adventure|Drama|Fantasy	5.3
5	The Golden Child	Michael Ritchie	PG-13	94	English	USA	Action|Adventure|Comedy|Fantasy|Mystery	5.9
6	Invaders from Mars	Tobe Hooper	PG	100	English	USA	Horror|Sci-Fi	5.5
7	April Fool's Day	Fred Walton	R	89	English	USA	Horror|Mystery	6.2
8	The Texas Chainsaw Massacre 2	Tobe Hooper	X	101	English	USA	Comedy|Horror	5.5
9	Jason Lives: Friday the 13th Part VI	Tom McLoughlin	R	86	English	USA	Horror|Thriller	5.9
10	Witchboard	Kevin Tenney	R	98	English	UK	Horror|Mystery|Thriller	5.7

Query 8:
Increasing/Decreasing trend of the movie genres for different years

- 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 "**crime**", in the early years, making the movie on this genre was less popular, then the directors started making more 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 different years.
- **Query:**

```
db.movies.aggregate([{"$match":{"genres":{"$regex":
reg_ex}}},{"$group": {"_id": "$title_year", "count": {"$sum":
1}}},{"$sort": {"_id": 1}}])
```



References:

- IMDB. “What Is IMDb?” IMDb, IMDb.com, help.imdb.com/article/imdb/general-information/what-is-imdb/G836CY29Z4SGNMK5?ref_=helpart_nav_1#
- Yueming. “IMDB 5000 Movie Dataset.” Kaggle, 16 Dec. 2017, www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset
- Flask official Documentation https://flask.palletsprojects.com/en/1.1.x/
- HTML Documentation and tutorial from https://www.w3schools.com/html/
- CSS and Bootstrap from https://getbootstrap.com/
- JQuery from https://jqueryui.com/
- Referring fonts from https://fonts.google.com/
- Animation: https://wowjs.uk/docs
- Charts Library: https://www.chartjs.org/
- Images of Slider and favicon are referred https://www.google.com/imghp?hl=en


Thank
You