CS 7450 | Milestone 7 - Project Final Report | Team Fresh Tomatoes

Team member: Xinyu Li, Xiongxin Li, Yiran Ma, Mario Wijaya, Bum Jun Park

Introduction

MovieHunt is an interactive visualization to help users find the next movies to watch. The visualization provides both the overall trend of movie ratings from 1987 to 2016, and the connection, defined by common director and actors, between a selected movie and the other ones. Users can filter the data cases by genres, ratings, year range, and change color encoding based on popularity, gross and budget. They can also explore the movies which share the same director and actors, and see the detail information of each one on the side.

Target User

Our system is designed for two groups of users: movie enthusiasts and casual movie viewers. Here is a brief description of the two user groups based on our interviews:

Table 1 User Persona

Enthusiasts	Casual viewers
• Maintain a habit of watching a certain number(~10) of movies per month; • Are Interested in certain directors and actors; • Have their own standards of good movies.	• Watch a limited number of movies per month & only watch more movies when having abundant free time; • Decide on movies based on good ratings/recommendations; • Tend to follow what's trending.

The two groups of people share the same need: looking for movies to watch. But have different pain points. Enthusiasts want to find movies that are related to a certain movie they like (in light of directors, actors and production team), but they find it hard to get this information from Google/Wikipedia; Casual viewers watch fewer movies, therefore having a stronger need to find just the right one that they will like. They tend to follow popular trends with an emphasis on ratings but they don't even more where to start while searching.

Here is a list of questions that users are able to look into with our system.

1. “What are the movies that are similar to the movie I like?”
   The visualization will generate a network graph based on the movie that users select or search for. The graph will locate the original movie in the center, and show the movies that share same directors and actors with that movies.

2. “How are these movies related to the movie I like?”
   With the same network graph, our system will code and label the connections. Users can also see more details about the movie.

3. “What are the high-rated movies in the genre(s) that I like?”
   The system will present an overview of the all the movies using a scatter plot mapping along the year and rating. Then users can filter certain genres that they are interested in. Also, they can choose to see a certain range of the ratings.

4. “How are the ratings of movies of different genres and movies that win awards distributed?”
   Based on the same overview of movies, users can choose to only see oscar-winning movies and movies of certain genres, and explore how their ratings are distributed.

Explanation

User Task

The primary goal for both of our user groups using our system is to find the movies to watch next. The visualization supports the different tasks based on user’s different level of knowledge of movies and different conditions when “hunting the movies”:

1. Browsing: When users are not sure what to look for, the visualization presents a scatterplot of movies to let them explore freely. The dots on the scatterplot are color coded by the level of popularity, gross or budget based on users’ choice to when they browse and explore.
2. Retrieve Value: When users identify a movie of interest, they can get the basic information of a movie the tooltip and on the detail card, compare the new movie to the original one and make decision based on the information.
3. Filter: When users know what category/range of movies to look for, they can filter the options (genres, ratings or award) as well as the range of the years to only show the data that they are interested in.
4. Find extreme and anomalies: When users are interested in finding top-rated movies(especially for casual viewers), the scatter plot helps them identify the

highest-rated movie for each year by spreading the movies out along the rating axis.

5. Searching: When users know what kind of movies they want to start with, they can search for the movie of interest by keywords, directors, or actors. Alternatively, when they don't have a clear idea of where to start the search, the visualization has a suggestion of actor, director, or movie will be displayed when user click "Surprise Me!" button.
6. Categorize distribution: If users are interested in a specific actor or director, they can search for it in the search bar and see how the movies that are related to this person are distributed over the x-axis(time) and y-axis(rating) on the scatter plot. Users can view overall average imdb score over the time period as well as looking at general distribution on based on imdb score yearly.

User Interaction

Our visualization is designed to support all the basic interactions users can perform.

1. Select: Users are able to see the title and rating of a movie by hovering over a dot, and they can see more detailed information, such as trailer link, actors and directors of a movie by clicking on the dot.

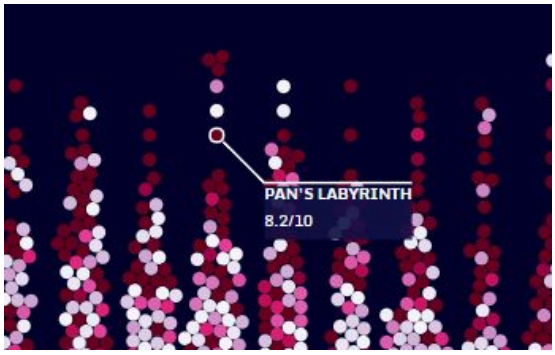

Figure 1.1 Hover to see brief annotation

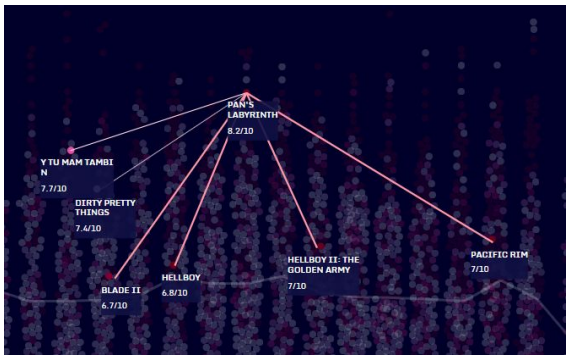

Figure 1.2 Click to see connections

Figure 1.3 Click on one data case and hover on another to see their details on right hand side

2. Explore: By clicking on one data case, users can explore the movies connected to the selected one, where the connection is defined by common actors or directors. This connection is calculated based on the user research we did with our target audience, who showed interests in movies directed by the same director or performed by the same actor.
3. Reconfigure: Users are able to change the color code of the data cases by selecting the attribute they are interested in: gross, budget or popularity (defined by number of users who voted on the movie on IMDB). The number is sequentially scaled with the color scheme and darker color means larger number.

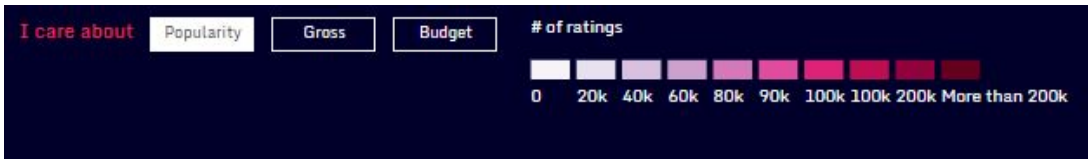

Figure 1.4 Change color encoding for different attributes

4. Encode: We use different color schemes to encode different attributes on data cases. Popularity is encoded in purple, gross is encoded in blue, and budget is encoded in green. We use d3-scale-chromatic (https://github.com/d3/d3-scale-chromatic) to sequentially encode the number to color, where larger number is represented in darker color. We also use double

encoding to show the strength of connection between two movies by using the stroke width and color luminance -- thicker line with darker color means the two movies have more people in common.

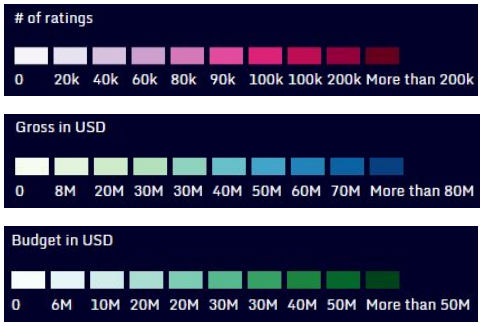

Figure 1.5 Three different color schemes

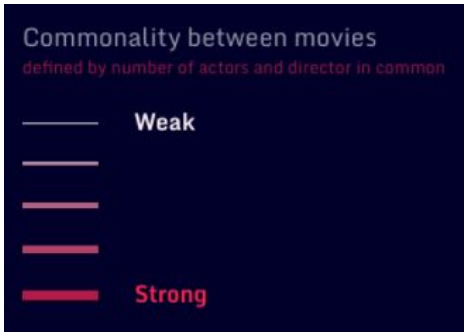

Figure 1.6 Line thickness+color encoding

5. Abstract/elaborate: We present an overview of all the data cases first, and users can see the detail information of each data case by hovering over (a brief annotation next to the data case) and clicking on (detailed information on right hand side) it. Users can also hover over the average rating line or the connection line to see the detailed meaning of the line.

Figure 1.7 Detailed Information (Selected)

Figure 1.8 Detailed Information (Mouseover)

6. Filter/limit: Users can choose to see a specific movie by searching on top or a group of movies with features they are interested by using the filters on the left hand side or the year range slider on the bottom.

Figure 1.9 Search functionality

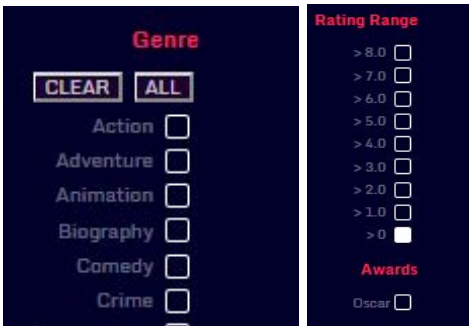

Figure 1.10 Genre, Rating, and Awards Filters

Design

Visual mapping

Each movie is mapped to a dot on the graph. Its position is determined by rating and year; and the color of the dot is determined by one of the three attributes (popularity which is shown by default, gross and budget) selected by users.

The average rating of the movies shown on the graph is drawn as a line to show the trend of the rating through the years.

The connection between two movies is also represented as a line. Its thickness and line color both encode the strength of connection. Thicker line with darker color connecting two dots means the two movies have more common actors and director.

Layout

We present the main graph in the middle and put the filters of ratings and genres on the left side, and the year range slider right below the x-axis so the proximity can convey the meaning of range slider better. The detailed information of selected/hoverd movie is put on the right side. Color attribute selection and its corresponding legend are put on the top of the graph. Search feature is put on the header next to the title of the whole visualization.

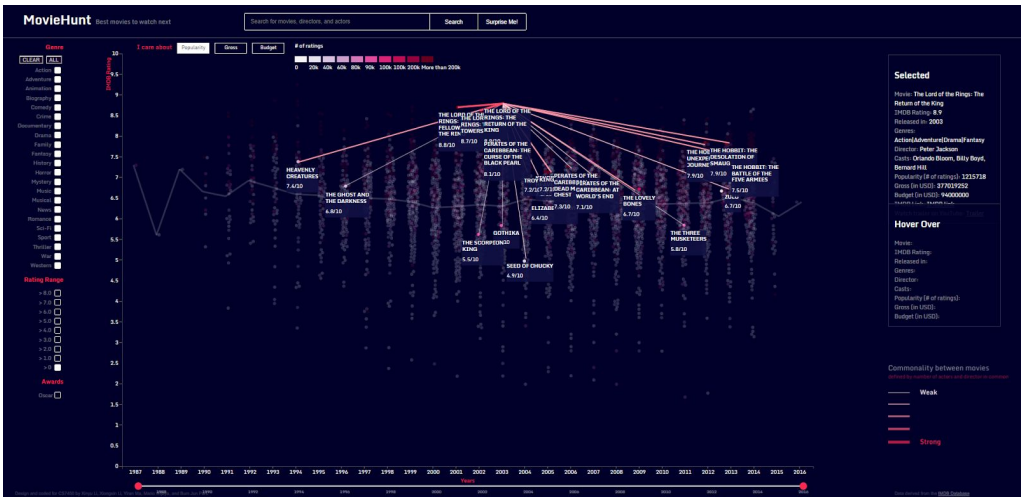

Figure 2.1 Overall layout of the visualization

Other elements

Design Alternatives:

We had several design iterations and we made some design changes after receiving feedbacks from Teaching Assistant and Professor Endert after Milestone 6. There were several design decisions we implemented and later chose not to due to better implementations:

1. Panning in canvas vs. Limit width of each year

Since the x axis is the year range from 1987 to 2016 and we want to display the overview of the whole dataset (which contains more than 3500 data cases), a single window screen cannot contain all the information. Users have to pan the browser to the very right to see the most recent years. We implemented panning of the x-axis and all the dots while fixing the y-axis on the left based on the suggestion provided by TA. However, during user testing, we found our users were surprised by the panning feature. In the end, we decided to increase the resolution of the screen and make the whole graph contained in one screen, so the users do not need to pan the browser to see the whole graph.

2. Growth of the force directed graph

   Initially we presented a single layer force directed graph to display the movies connected to the movie user selected. We also tried to implement a multi-layer one based on the feedbacks we received from Milestone 6. However, the graph looked really messy and non-informative, especially when the movie had a lot of connections (Figure 2.1). Even the movie having fewer connections can still have lots of cycles and force directed layout could not even resolve the crossing edge issue (Figure 2.2). Therefore we decided to remove this graph completely and let the user explore the connections solely on the main graph. They could trace the second layer of connections of each node connected to the movie they selected by clicking on the individual node.

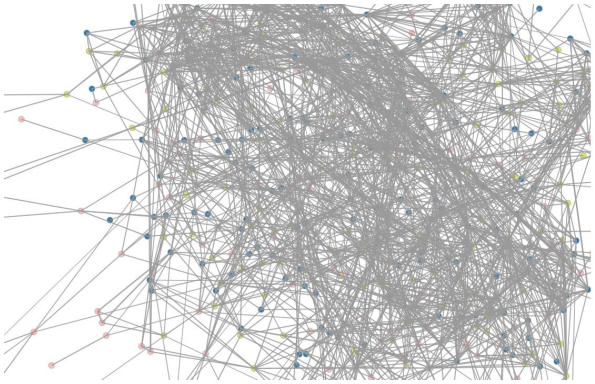

Figure 3.1 Force-directed graph with many connections connection

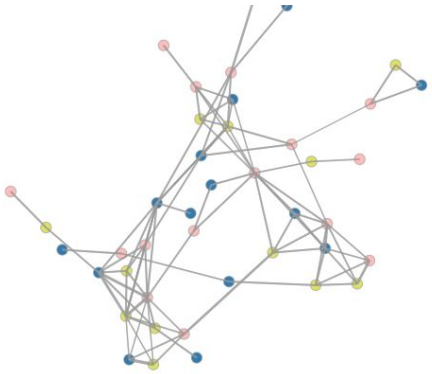

Figure 3.2 Force-directed graph with fewer

Data Source

Our data source is primarily based on a CSV file generated by Chuan Sun (https://github.com/sundeepblue/movie_rating_prediction) using data from IMDB and www.the-numbers.com. We merged it with another two CSV datasets, one containing the Oscar winning information (https://www.kaggle.com/theacademy/academy-awards) and one containing the tagging information (https://www.kaggle.com/grouplens/movielens-20m-dataset). After merging, we eliminated movies that were produced earlier than 1980, which we considered as outliers in the current movie dataset. We also removed some of the attributes, such as the number of faces appearing in the movie poster and aspect ratio, which our target audience showed no interest in.

Table 2 The Summary of our four movie data sources

Source	Characteristics	URL
IMDB + the-numbers .com	- Contains 28 variables and 5043 data points. - Each movie has its unique IMDB url which contains IMDB id number. - Contains names of each movie's top 3 actors/actresses and a director.	http://www.imdb.com/interfaces https://github.com/sundeepblue/movie_rating_prediction
OpusData	- Contains 13 variables and 1971 data points. - Each movie has its production budget, financial summary of domestic box office and international box office.	https://www.opusdata.com/dataservices.php
Kaggle	- Contains 6 different csv files (genome_scores, genome_tags, link, movie, rating, and tag) - These datasets shares common key variables, such as movieid and tagid. - The link dataset contains IMDB id which can be used to merge with the IMDB dataset. - The rating and tag datasets were generated by users (subjective data).	https://www.kaggle.com/grouplens/movielens-20m-dataset
	- Contains 5 variables and 9963 data points. - Shows the Academy Awards history (winners' names and awards) from 1927 to 2015.	https://www.kaggle.com/theacademy/academy-awards

Below is the list of data attributes, data type, and description of the dataset that we used for the visualization:

Table 3 Overview of the dataset

Attribute Name	**Data Type**	**Description**
actor_1_name	string	Name of actor 1
actor_2_name	string	Name of actor 2
actor_3_name	string	Name of actor 3
budget	int	The budget of the movie
director_name	string	Name of director
genres	string	A combination of 21 unique genres[1] separated by '|'
gross	int	The gross box earnings
imdb_score	double	The IMDB rating value
movie_imdb_link	string	IMDB URL of the movie
movie_title	string	The name of the movie
num_voted_users	int	The number of ratings on IMDB rating score
oscar	boolean	The attribute has the value 1 for if the movie has at least one oscar award and 0 if the movie has none
title_year	int	The year when the movie is released

[1] Action, Adventure, Horror, Drama, Animation, Comedy, Mystery, Crime, Biography, Fantasy, Documentary, Sci-Fi, Romance, Family, Music, Thriller, Western, History, Musical, War, News