Steve Letourneau

Peter Cheung

Jeff Hofer

Alex Neilson

Feb. 15, 2017

CMPT 350

Design Document

Our group project is to create a data visualization tool that allows users to find movies that they want to watch. The tool will be a web app which will help solve the issue of users spending unnecessary time and effort in determining which movie to watch.

Domain Situation:

It can be difficult deciding which movie to watch based on the multitude of choices. Some user stories for an app that would address this issue are:

* As a movie watcher, I want to discover movies that I find interesting.
* As a user, I want to find movies with similar subject matter that I am interested in.
* As a movie watcher, I want relevant suggestions of movies so I can find new movies to watch.
* As a movie watcher, I want to easily and quickly find movies to watch so that I can save time looking for movies.

The app will solve these problems and create many benefits for users. The app will help users:

* Find a movie or movies that the they will enjoy.
* Save users time by reducing the amount of time searching for movies.
* Broaden users’ knowledge in movies by introducing them to films that they may not have known or heard about.
* Potentially expand or reinforce a user’s interest in a genre, actor or director.

The vis tool will utilize a dataset of movie data which includes data such as: title, directory, year, duration, actors, Facebook likes, gross revenue, genre, plot keywords, budget, and IMDB score. The dataset is in a tabular format and contains approximately 5000 movies. The data source url is: <https://data.world/data-society/imdb-5000-movie-datase>. In addition, we will use the IMDB open API called OMDB: <https://www.omdbapi.com> . This has additional features such as Movie posters, Oscar awards and plot synopsis.

Task and Data Abstraction:

The task that the user is doing is searching the data. Once a target is found, they can query the data to see additional attributes. For example, the movies will be filtered based on similar movies to the movie the user selects. The app will also allow the user to query the movies selected by clicking over a movie and see additional data from the OMDB database such as a plot synopsis, Oscar wins and nominations and extra cast and crew information.

The data abstraction plan is to abstract data from the domain of all data into a more usable form. We will group the data by relating similar attributes between table values to create sets.

Visual Encoding:

The vis formats for the movies app is a Space Filled Graph and Bar Graphs. The app will initially fill the space with 15 random movies which will allow the user to begin exploring. Or the user can enter or select a movie using a Combo box to find a movie. The Space Filled Graph would then update based on the web application sorting algorithm which would find related movies. The size of the movies would indicate the magnitude of relatedness.

Movie data will also be presented to the user in form of bar graphs. The user will be able to see numeric data for movies such as user reviews, IMDB score, Facebook likes and IMDB votes cast.

Integration Idiom:

A user will be able to interact with the data be either selecting a movie or entering one or more movies to see related data. The user will have the option of entering multiple movies to see the movies related to more than one movie. In addition, the user can customize the sorting algorithm using filters for different attributes. For example, the user can adjust the weighting for properties such as plot keywords, actors, director, title, year and country of origin. This ability to adjust the setting will provide interactivity and customization for users to explore the data set. In addition, a user can tag movies that they are no longer interested in and those movies will be removed from the active view but not the database.

When a user selects a movie, the user will be able to rerun the search for related movies to the movie the user queried. Furthermore, when viewing the bar graph information, a user will be able to select a bar to query a specific movie which will show additional information specific to that movie.

Algorithm:

The algorithm uses the attributes of a film such as actors, director, genre, keywords to find other films with similar attributes. When two films have similar attributes, the score of that attribute is added to a total called the “relevancy score”. The films with the highest relevancy scores are deemed to be the most like the film selected by the user. Broader attributes, such as language, have lower weights and therefore have less effect on the relevancy score. Conversely, precise attributes like keywords yield higher scores and add more greatly to the relevancy score.