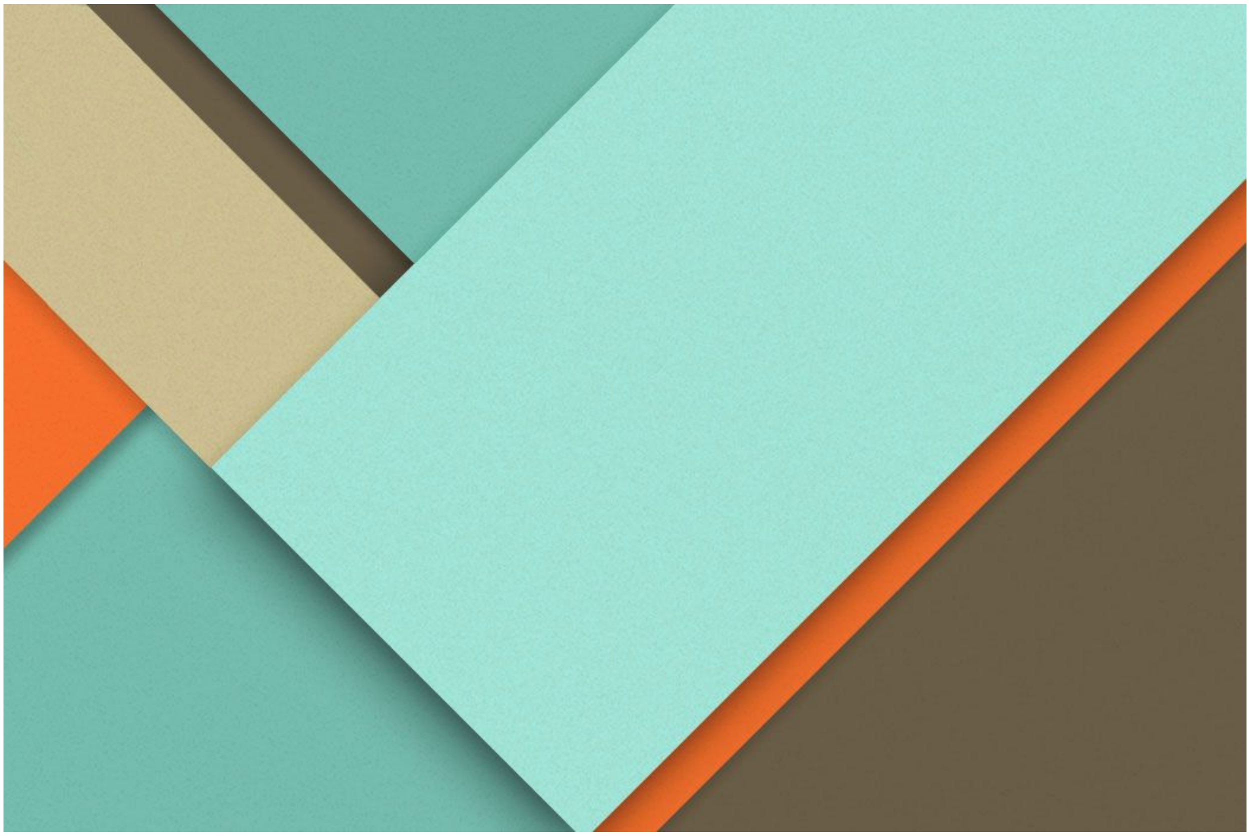


# MovieMiner

Process Book

—

Group

Sahar Mehrpour, mehrpour@cs.utah.edu

Sunipa Dev, sunipadev@gmail.com

Siddartha, siddartha1191@gmail.com

Zahra Fahimfar, zahra.fahimfar@gmail.com

## Overview and Motivation

**Provide an overview of the project goals and the motivation for it. Consider that this will be read by people who did not see your project proposal.**

Watching movies is one of the most fun activities people do in their free time. However, selecting a movie to watch without prior decision is not an easy task, especially with the volume of movies churned out every year. Every individual has specific preferences in terms of genres, themes, actors or even directors. Searching the potential favorite movies in the net takes plenty of time. A good and informative visualization of movies helps users in the manner of an indirect recommender system.

One of the visualization introduced in class was 'caleydo', http://www.caleydo.org/tools/upset/. The original motivation of the project is introducing a method for visualizing the intersecting sets. One of the datasets visualized by this method is the IMDB movie dataset.

## Related Work

**Anything that inspired you, such as a paper, a web site, visualizations we discussed in class, etc.**

In Data Visualization class, many visualization techniques were introduced: bar charts, scatter plots, pie charts, box plots, graphs, tables, etc. Not every visualization works for any dataset. To gather intuition for designing a decent visualization, we surfed the net and skim some of the existing visualization for a similar data.

- **Upset** (http://www.caleydo.org/tools/upset/): In this visualization, for each subset of movies, a brief summary of each subset is provided through different charts (bar charts and box plots). This project led us to design a table to visualize the information of a selected subset of [=filtered] movies.
- **Tableau** (https://public.tableau.com/en-us/s/gallery/imdb-movies-visualized): In this visualization, a user can filter movies based on years, ratings, countries, and genres. Then the filtered results were displayed in one main scatter plot. In this plot, each movie is displayed as a point in a year-rating chart. The filtering options are similar to what we had in mind. Nonetheless, none of the filter visualizations are similar to our designs, except for the year filter which is slightly different from our design.
- **Liveplasma** (www.liveplasma.com): This visualization is mainly based on queries of a specific movie. The result is displayed in a graph, in which it is not clear what information is displayed by edges.

  This visualization along with **Siggraph** (http://www.cs.utah.edu/~kwu/vis/sigvis.html) were the intuition for designing graphs  to visualize the connection between movies.
- **Cinemetrics** (http://cinemetrics.fredericbrodbeck.de/): This is another visualization for movie datasets. The objective of this project is mainly provide a "fingerprint" for each movie, which is clearly far from our project objectives.

- **YellowFin Website** (http://www.yellowfinbi.com/YFCommunityNews-Oscars-inspired-analysis-Dissecting-Hollywood-s-movie-industry-with-data-visuali-156742): This website provides a several charts to summarize a movie dataset. The visualizations are static (not interactive). These designs are pretty much similar to the ones we designed for the filters in our project.
- **Kaggle Website** (https://www.kaggle.com/snowsky/d/deepmatrix/imdb-5000-movie-dataset/movie-data-visualization): In this website, there are several visualizations for the dataset we are using in our project. In this particular visualization, four simple static visualizations are provided.
- **Kaggle Website** (https://www.kaggle.com/ruxizhang/d/deepmatrix/imdb-5000-movie-dataset/visualization/run/386697/code):
  This website includes different charts for the same dataset we are using. These charts helped us to get a better intuition about the data.
- **Graphlix** (http://vis.ninja/vis/graphlix/): This project is also a graph visualization of a movie dataset.
- **Amazonaws Website** (https://rstudio-pubs-static.s3.amazonaws.com/152157_162423dcce514673b0bc2e83f47084e9.html): In this website, visualization is used to analyze the movies distribution. The main purpose of these visualization was to provide statistical information about the data.

## Questions

**What questions are you trying to answer? How did these questions evolve over the course of the project? What new questions did you consider in the course of your analysis?**

We are trying to build a visualisation for filtering movies and getting as an output, movies related by certain aspects, sort of like what a recommender system would do. Just that, in our case, we are not using any machine learning and the filters and parameters are set by the user and not auto generated by a machine learning algorithm. This makes it easier for the user to understand the resultant recommendations. Some methods of filtering that we have or are aiming to provide are:

- What are the list of movies within particular genres, actors, directors, years/time periods etc?
- Who are the directors and actors working in a specific movie? What genre does it fall in? What is its gross, budget, rating?
- What are the movies related by “keywords”?
- In which other movies did this “actor” and this “director/actor” work together? (Maybe that is how a user can determine the next movie to watch)
- Region/language distribution of movies in a given year.

# Data

**Source, scraping method, cleanup, etc.**

We obtained the main dataset for movies from Kaggle website. Here is the link to the dataset: https://www.kaggle.com/deepmatrix/imdb-5000-movie-dataset. A (free) account is needed to download the dataset.

We also have an additional dataset for Academy Awards downloadable from: https://cs.uwaterloo.ca/~s255khan/oscars.html#download.

We added an official poster of each movie in one of the views (Information view). In the primary dataset, the IMDB link of each movie is provided. However, retrieving the poster is not trivial as the html element has not an ID. In addition, we doubted whether it is legal to 'data mine' the IMDB website (http://www.imdb.com/conditions). So instead, we extract the IMDB ID of the movie from the link provided, and then we used website https://www.omdbapi.com/ to get a json object for each movie, in which there exists a link for a poster. We used that link as a source of the poster.

# Exploratory Data Analysis

### What visualizations did you use to initially look at your data? What insights did you gain? How did these insights inform your design?

As we mentioned in the Related Work Section, there are a few visualization with the same dataset. We used these visualization to have a general insight about the data. There are several working systems which suggest movies to users (like Netflix) or help classify them (such as Calyedo). We wanted an amalgam where the user can filter and classify the movies himself and get movie suggestions in the end. Also, as we have filters in our visualization, and each visualization depicts some information, we wanted to know the cardinality of the data in each subset. We mainly used `d3.nest()` to view the nested data and have a better insight about the extreme cases; For example, in our dataset USA has 3807 movies while Egypt has only one movie.

# Design Evolution

**What are the different visualizations you considered? Justify the design decisions you made using the perceptual and design principles you learned in the course. Did you deviate from your proposal?**

It was clear that we cannot visualize all the data in one view without aggregation/summary. So we decided to select the features that we thought best represented the data and decided to provide filters to these features so that the user too can appreciate the dimensions of the data. Filters also visualize a decent summary of the dataset from a specific aspect. Each of the filters, result visualization, and information visualization have had their own evolution:

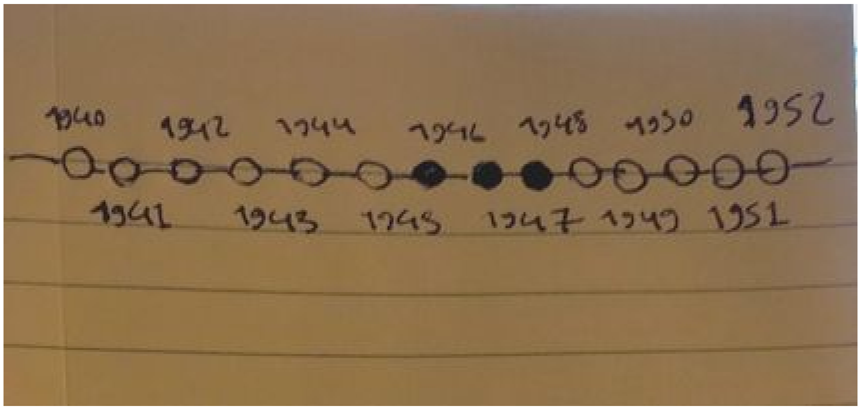
## Filters

Initially we designed as many filters as we could derive from the datasets, independant from each other, forming a query.

Critique: We realized too many filters makes the screen unnecessarily congested and distracts users from other views. So, we decided to keep number of filters minimal.

- **Year Filter**:

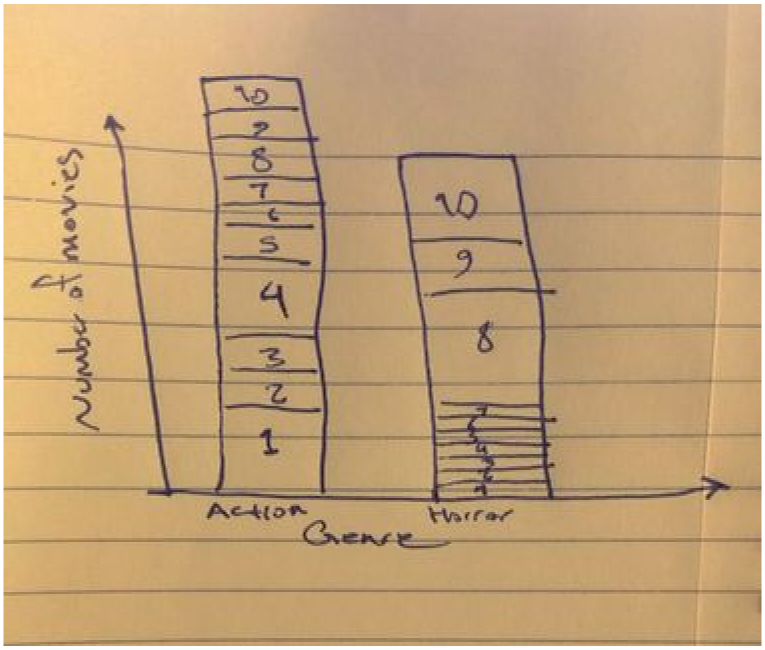
The goal of this filter to show accumulated information results of movies. In other words, instead of just showing the information of one single movie, the goal is being able to also show a set of movies in years. To achieve this, we analysed different kinds of strategies. However, each of these strategies has its own limitation. For instance, showing each year by a circle was an option, but the problem was that it didn't look good, since we have large number of years.



Therefore, axis data visualization was chosen to fulfil this task. Similarity, having a separate ticks for each movie was not very nice, and so, decade representation of years per tick was chosen. As a result, the most convenient way of choosing a set of year for the end user seem to be brush selection. Thus, we added a brush to the axis which enables the user to easily select a set of years.

- **Genre-Rating Filter**:

This filter attempts to show the relationship between movie genres and their ratings. The first challenge was the best method to show this relationship. Both of these features are discrete with large number of possible values. Therefore, methods like bar chart leads to a messy output. Circles seem to be the best option as we can change the size as well as the color of the circles based on the data.



The second challenge  was the fact that some genres has very few movies belonging to them. So, we decided to group such genres together as a single group called "others". As a result, we have 13 genre groups and 10 rating groups.

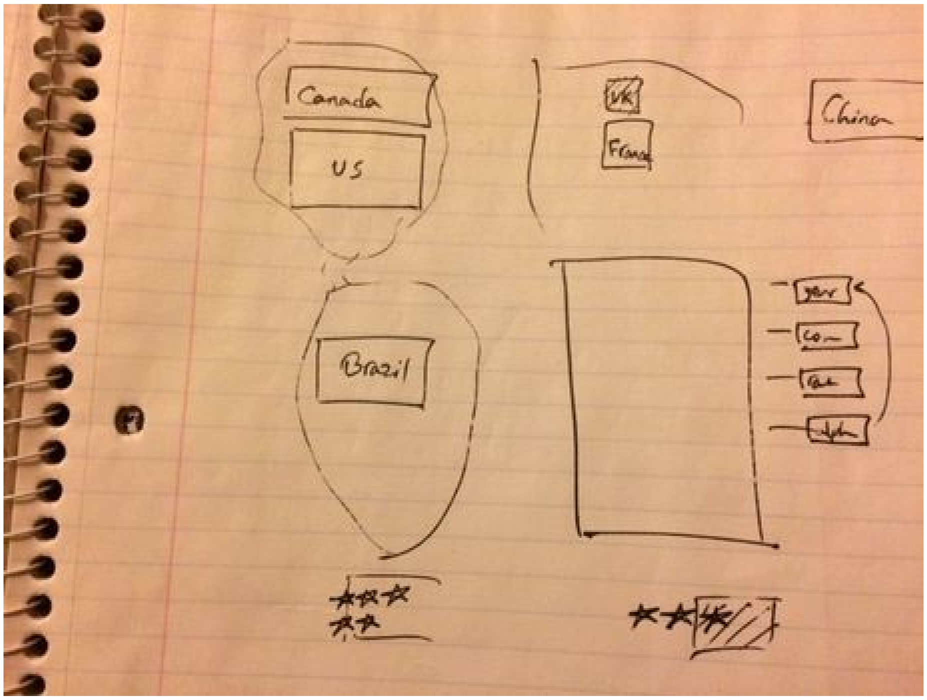
The third challenge was circles with small number of movies. In other words, specific genre-rating has very few number of movies, although the whole genre group has large number of movies  (i.e., not eligible for others group). To address this challenge, we don't draw a circle for genre-rating groups with less than 10 movies, since the corresponding circle is extremely small, which makes them useless for the users.

- **Map Filter:**

The map filter, as is apparent from the name, helps filter out the movie list based on country. The motivation for this filter is that users often wish to watch movies from certain nations or have a taste for foreign movies. The best way to spatially represent this is a map. For this, we tried several projections.

One concern was that since we cannot encode size in a map, some countries might get less importance than warranted by the number of movies they produce(encoded by colour in the map). A probable idea was to put in a globe like view and then letting the user turn the globe as that would make each country seem bigger in a given frame. But instead of giving it a pleasing appearance, the countries looked contorted due to the curvature of the globe and were harder to place.

We also considered having only the countries with desired movies appear on the map as an overlay on the outline of the continents:



But that we scrapped that plan as it did nothing to the problem with the size of the countries and made it look not as good as before. The mercator projection looks the best so far but we are looking at other possible options to represent the spatial distribution of the movies.

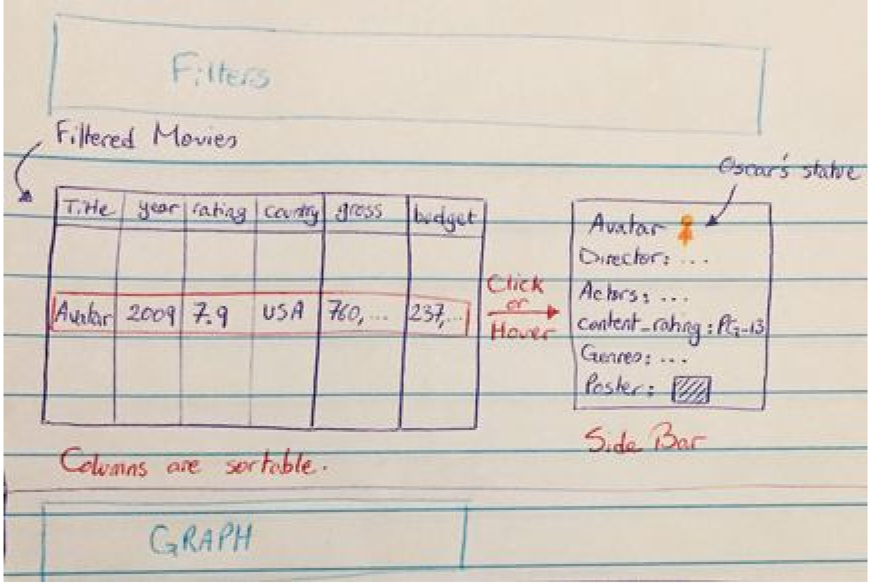
## Table, Graph, Information

- **Table**:

**First Design:** The table displays some information about a set of movies. There are a lot of attributes for each movie which can not be shown in the table at the same time. So, we must select the attributes which we thought they are informative. We selected Title, Country, IMDB score, Year, Gross, and Budget. The columns of the table is sortable (ascending/descending)

To display the rest of the information, we designed another view, Information. This view is updated by clicking or hovering over a movie in the table.

There are some drawbacks in this design. First, initially the table is empty and empty table is not appreciated. Second, assume the filtered movies are 1000, which is possible as the dataset has more than 5000 movies, we will have an extremely long table.



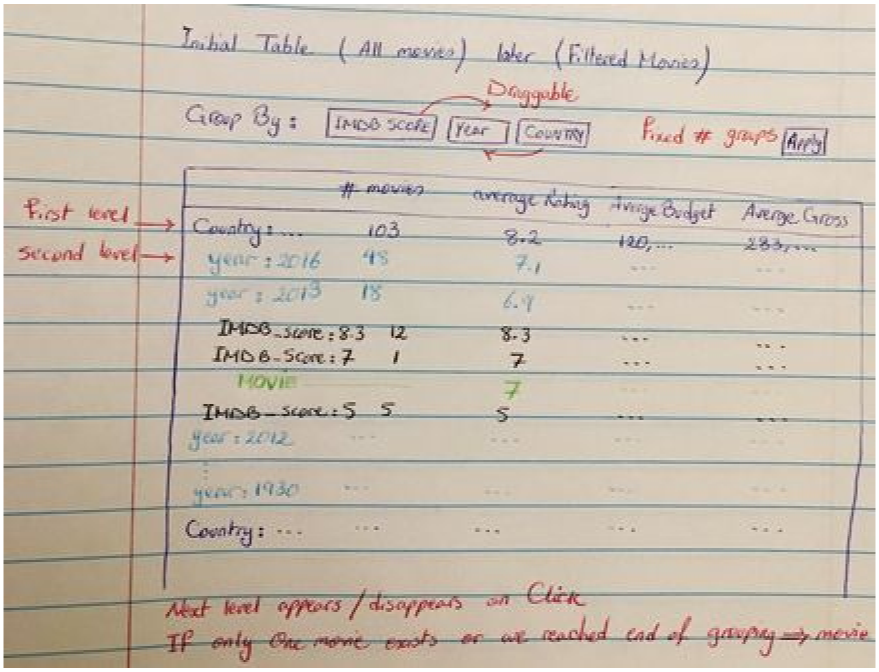
| Title | year | rating | country | gross | budget |
|---|---|---|---|---|---|
| Avatar | 2009 | 7.9 | USA | 760,... | 237,... |

Avatar
Director: ...
Actors: ...
content_rating: PG-13
Genres: ...
Poster: ▨

**Second Design:** In our second design, we consider movie attributes similar to the first design to display in the table. We also consider grouping the movies into categories ordered by a user. The categories are fixed and only can be reordered. There is an “apply” button which performs the grouping and populate the table.

Assume the first category is “Country” and the second is “Year”. The initial rows of the table displays a summary of the movies in each “Country”. Then by clicking on a row, saying “USA”, the overview of the movies in each “Year” built in the US is displayed. The same scenario happens for the rest of the categories. If we reached to the end of grouping or there is only one movie for a specific “summary” row, by clicking the movie will appear in the next row.

There are some drawbacks in this design as well. Assume the filtered movies are based on a specific country and a specific year. So, ordering based on “Country” and “Year” seems to be redundant. Also, we must consider a case when the user does not like to view “compulsory” grouped movies.



Inital Table (All movies) later (Filtered Movies)

Group By: IMDB SCORE Year COUNTRY Draggable Fixed # groups Apply

First level →

Second level →

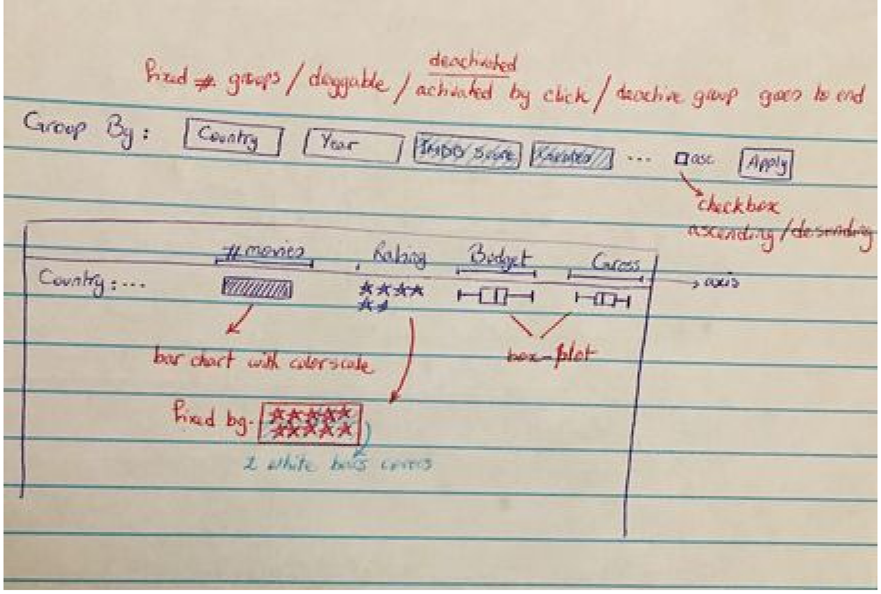
| | # movies | average Rating | Avrage Budget | Averge Gross |
|---|---|---|---|---|
| Country: ... | 103 | 8.2 | 120,... | 283,... |
| year: 2016 | 45 | 7.1 | ... | ... |
| year: 2013 | 18 | 6.9 | ... | ... |
| IMDB_Score: 8.3 | 12 | 8.3 | ... | ... |
| IMDB_Score: 7 | 1 | 7 | ... | ... |
| MOVIE | | 7 | ... | ... |
| IMDB_Score: 5 | 5 | 5 | ... | ... |
| year: 2012 | ... | ... | ... | ... |
| ⋮ | | | | |
| year: 1930 | ... | ... | ... | ... |
| Country: ... | ... | ... | ... | ... |

Next level appears / disappears on Click

If only One movie exists or we reached end of grouping → movie

**Third Design:** The third design is very similar to the second design. We consider the option of activating and deactivating group categories. By clicking on a specific group button, that group option is deactivated and moved to the end. The dragging option also exists in this design. We are also considering a checkbox to choose between ascending and descending orders.

The other changes we consider is displaying the summarized information with box plots and bar charts (with a color scale). This gives more information to the user. In addition, this visualization makes the table more appealing. Also, we are considering to show stars instead of a number for ratings, to make the table even more appealing.

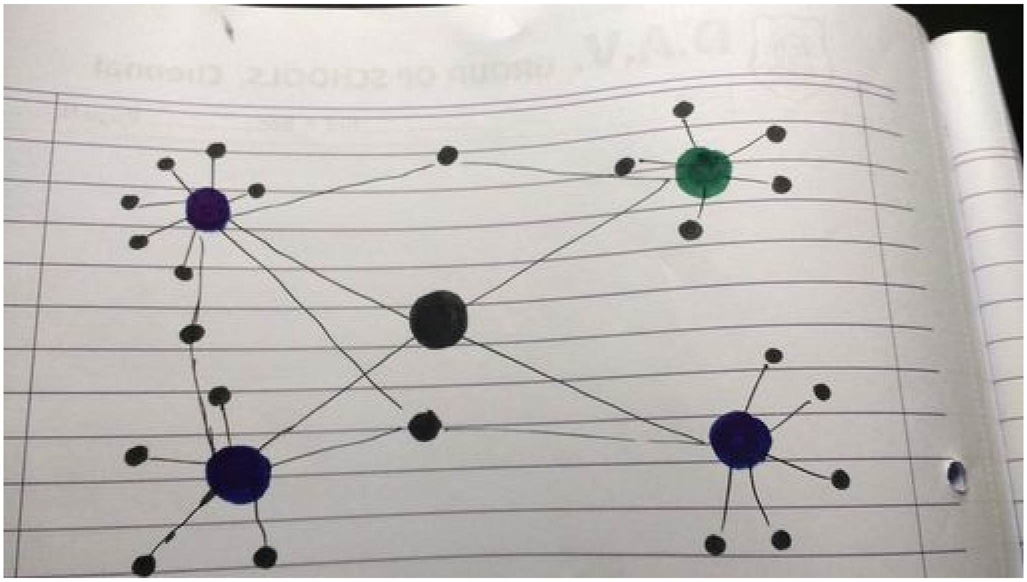


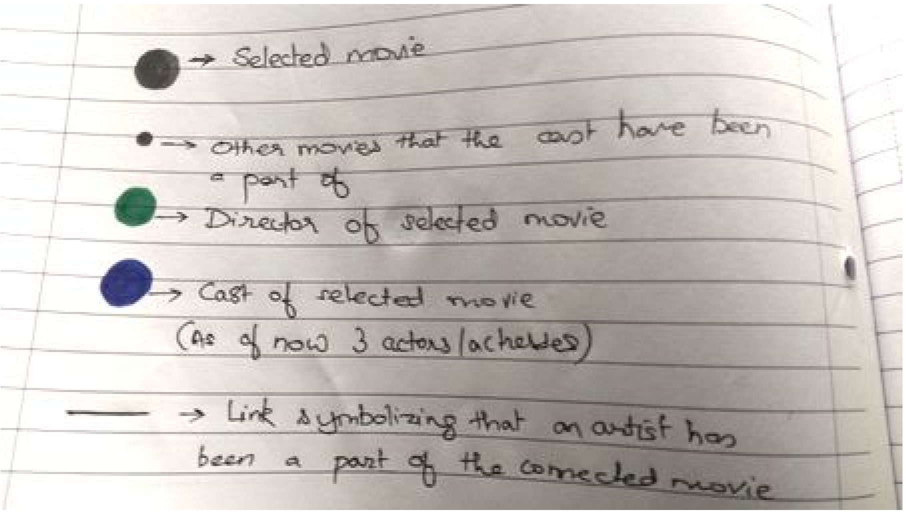
- **Information**:

From the beginning, the Information view had a concrete design. We want to display full information of the selected movie. The default design is presented in the sketch of the first design of the Table.

- **Graph**:

Often, the choice of a movie is based off movies watched from the past. The cast and directors play a key role in one's choice of watching a particular movie. The graph visualization enables the user to view this information. A simple illustration of how this would actually look can be seen below.





● → Selected movie

• → Other movies that the cast have been a part of

● → Director of selected movie

● → Cast of selected movie (As of now 3 actors/actresses)

—— → Link symbolizing that an artist has been a part of the connected movie

Implementation wise, a force directed graph fits our requirement. A few implementation specifics:

1. The distance between the nodes or the length of a particular link, depends on how many cast members have worked in the corresponding movie. Meaning the selected movie links are the longest, and movies with only a single link are the closest.
2. As of now, the selected movie title appears near the node. The name of the other movies and the cast members appear on mouse hover on their corresponding nodes.
3. On clicking another movie, the same graph is generated for it.

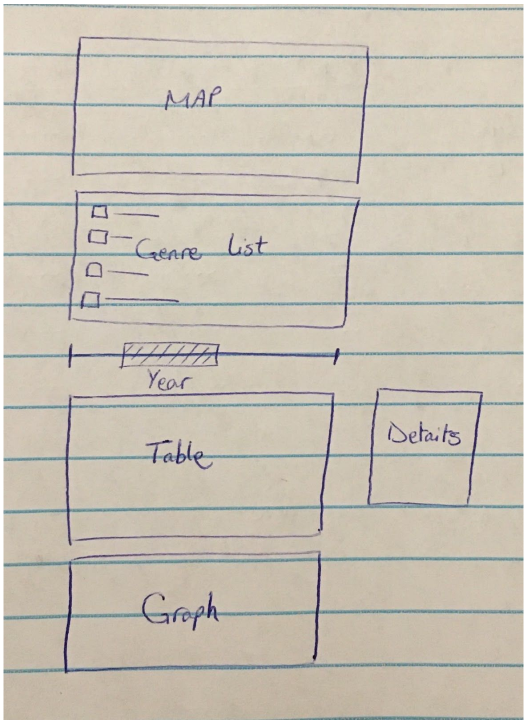
Work in progress:

1. Better aesthetics, size of the nodes, appearance of the nodes as circles/images.
2. Other features like changing opacity on hover to easily view the links and also the name of the movies.

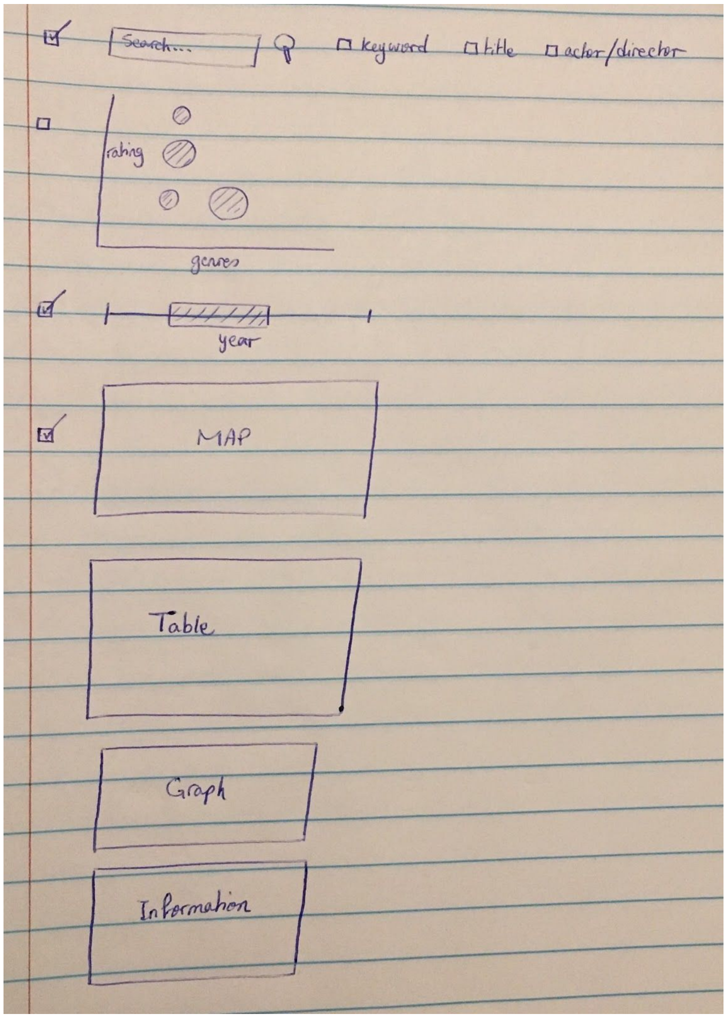
## Layout

We also had several designs for the full layout. We played around with the placement of the filters and devised some layouts. We were vacillating between having all our visualisations in one single page or getting an overlay for some sections.

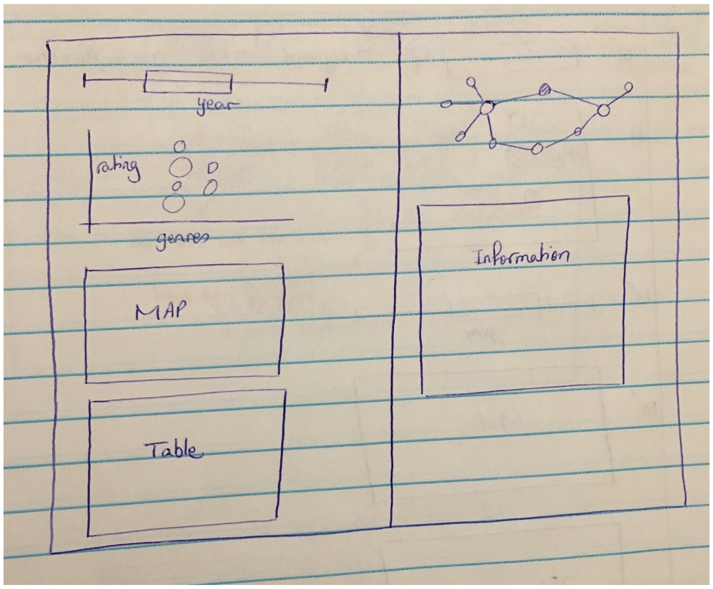
**First Layout:** In the first layout design our main focus was the location of different views. In this design we decided to place all views in a single column following each other, except the Information view, which we decided to put it on a side. Since the Graph is formed after selecting a specific movie from the Table, it had to be after the Table. One problem in this design was that not all the views need a wide screen. So this design was not space-efficient.



**Second Layout:** In the second layout, we decided to put the Information view at the end of the page. In this design we had the same problem of space-efficiency as in the first design. Here, we tried to answer: How does a user activate a particular filter? To answer this, we added check boxes near filters. Nonetheless, the check boxed did not seem appealing to the members of our group. So we passed up this design. In this design we also considered a search bar. However, we could not decide on the interaction of this bar and the rest of the views. So in our next design, we temporarily crossed out the search option.

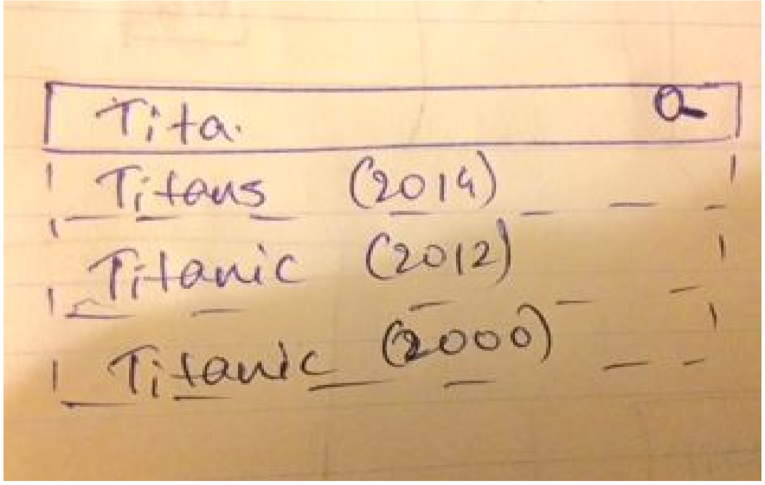


**Third Layout:** Our third design, we considered two columns. In the first column (on left) the filters and the table is drawn. On the right, the graph and the Information view is drawn. This layout has the potential of adding a search tool with an independant functionality with respect to the filters. In this layout we may add a search bar on the right column, so that by searching a particular movie, the graph and the Information view is populated.



## Search Option

An alternate way of getting recommendations from a movie is to feed in a movie and get the graph feature suggest movies directly. We want to do this by adding a search box with a select feature.



We are considering implementing this in more than one section.

# Implementation

### Describe the intent and functionality of the interactive visualizations you implemented. Provide clear and well-referenced images showing the key design and interaction elements.

We used the technique of object orientation. In the implementation, we have several objects defined in different javascript files. We have a total number of seven objects: Graph, Information, MapFilter, RatingGenreFilter, Table, YearFilter and Interactivity. MapFilter, RatingGenreFilter, and YearFilter are the objects for filtering views. Table is the object for the table view displaying the filtered movies. Graph and Information are the objects corresponding to the graph and information views which visualize some information about a specific movie. Interactivity object manages the interaction between different views.

The interaction in this visualization is as follows. Initially all the filters and the table are populated based on all the movies in the dataset. The user has several options: brush a range of consecutive years from the year chart, select one or several points (circles) from the genre-rating chart, select one or multiple countries from the map chart. Performing either of options updates the rest of the filters and the table based on the filter.

Lets consider the following scenario. First the user select years 1980-1990. The genre-rating chart and the map filter and the table only show information for movies produced between 1980 and 1990. In the next step, the user selects the US and Germany in the map filter. The year chart and the genre-rating, and the table will be updated showing information of movies produced between 1980 and 1990 in the US or Germany.

Selecting a particular movie will draw or update the graph and the information view. Selecting a different movie in the graph will update the existing graph and the information view.

At this time, the information view is independent and has no interaction with other views. However, we may consider interaction between the information view and the graph. For example, by selecting a director or an actor, the graph will be updated.