Milestone 3: Process Book

Omid Karimi Luca Bracone Emna Maghrebi

June 2022

1 Introduction

The goal was to create a data visualization with the dataset of our choice. We chose to work with the IMDb databases out of general interest for cinema.
In our visualization, we wanted to first have a rapid access to different movie rankings and basic data of the movies. Then visualize the ratings and number of votes of movies through the years according to their genres with plots. Here we wanted to observe whether some genres are more popular than others and how it evolved through time.
Lastly, we wanted to have a quick look at connection between actors through a graph, an edge between two actors if they played in a same movie.

2 Movie rankings visualization

First we built a ranking list of the movies. To this end, we have to first load the dataset and have a function which computes the top 10 movies given a metric (between average rating and number of votes), a genre and a year. Then we can create a list and add dropdown selectors which can be used to update the list. Each time a new value is selected in a dropdown, the old list is removed and the new list is computed and displayed. The list is made more beautiful with some CSS.
There is also a feature when one can click on a movie in the list, the movie will be highlighted and a tooltip will pop on the side giving more informations on the selected movie. Also, we fetch from another online movie database, which provides a nice API, the poster of the movie and display it in this tooltip.
It is very similar of what we had in mind from Milestone 2 (see Figure 1).

As a detail, we added a loading animation (found here) when updating the list and fetching the poster. The loading is actually "fake", the computation are very fast, but this create the illusion of loading.

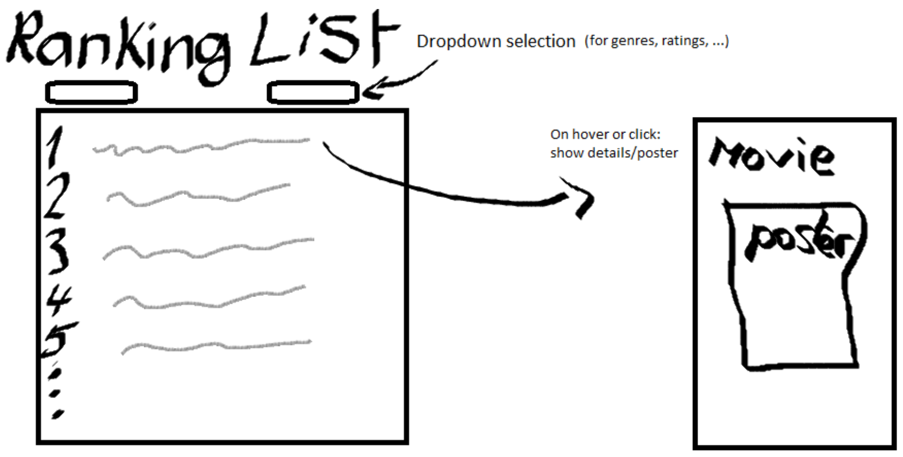
Ranking List
Dropdown selection (for genres, ratings, ...)
On hover or click: show details/poster
1
2
3
4
5
Movie
poster

Figure 1: M2 sketch for Part 1

3 The plots

3.1 Stacked area chart

With the suggestion of the professor, we added a normalized stack area chart. For this, we had to prepare the dataset: first group the movies by year and count the number of movies per genre. This way, we can use the d3 function "stack", which then can be used to create the different area and create the stacked chart.
We had a lot of trouble understanding how these functions worked, and which way to prepare the dataset to obtain a chart and in the end we chose to only keep 10 popular genres and starting from 1960 for a better visualization. We tried many example found online, the one that helped us the most can be found here.

3.2 Bar plots

Here the idea is the build a bar plot that can be updated from dropdown selection. It is very similar to part 1, we need to compute the dataset to plot on change of the dropdown value and display the new plot. There is also an animation for the bar plot. At first, we did not have a clear idea of which plot to use, as seen from the Milestone 2 sketch (Figure 2), and decided to go with a bar plot which is easier for comparisons.
We also hesitated to have multiple plots, with separate dropdown selections, making it easier to compare between them, since the idea is to observe a change in behaviour of the voters through the years. We opted not to keep this, because it did not look very good on the visualization.

As a small detail, we tried to create the "illusion" of a cinema projector. Each time the plot is recomputed, from the image of the projector, we added a white triangular

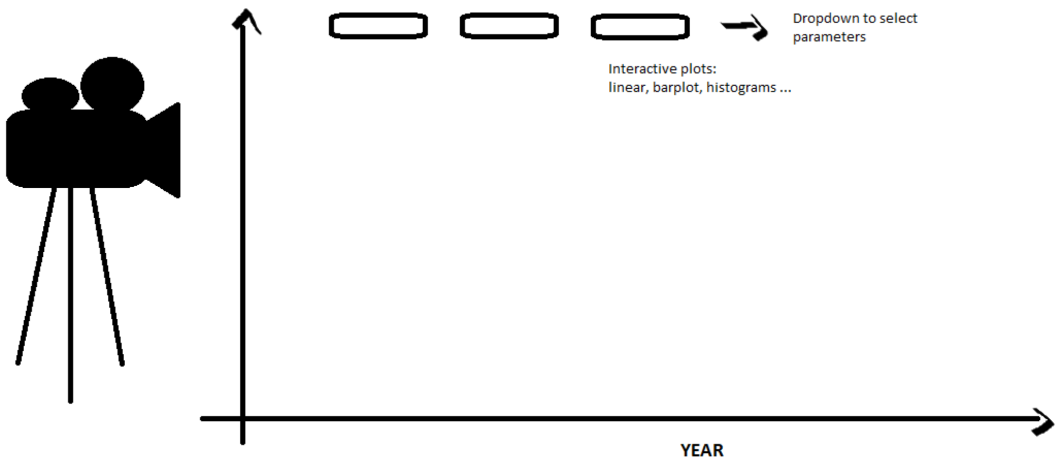
Dropdown to select parameters
Interactive plots:
linear, barplot, histograms ...
YEAR

Figure 2: M2 sketch for Part 2

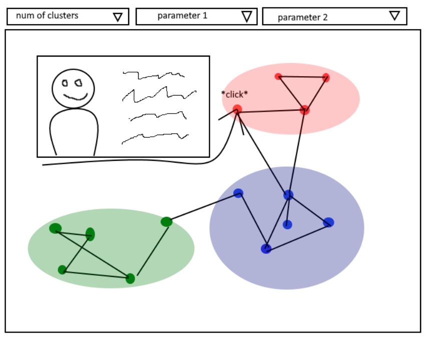

Figure 3: A sketch of the original idea for the actor graph

div with less opacity that flickers when the plot is being constructed. It looks kind of clumsy, we had some other ideas to make it look better, like having a better object than a triangular div or an actual image, but as it was not very important we kept it that way.

4 The actor graph

In the third part of the website we wanted to do a clustering of the actors based on if they starred together in a lot of movies or not. Figure 3 shows a sketch of what that should have looked like. The inspiration came from similar work that clustered movies and books based on their genre. So, the idea is the following: we have a graph whose nodes are the actors and two nodes are connected if the corresponding actors played in a movie together. Originally we planned to make a plot with the 30k most voted actors. While it would have been possible to make such a static plot in python, since the aim of the course was to develop the ability to interact with the data, we chose to use cytoscape.js [1] with a reduced number of actors (120). For the display, originally we wanted to perform spectral clustering [3], which is an algorithm we personally like a lot, in order to obtain a “map” of the actors like in similar projects (see for example this blog post [2]). In the end, we decided against it because of implementation constraints. Indeed, given that there was no ready-made package for it in JavaScript, doing all the linear algebra ourselves would have taken too much time. Instead we opted for the simpler “cose” layout which places the nodes according to a physics simulation where the edges are springs. Furthermore, we wanted to make it so that when the user clicks on a node, additional information about the actor would be shown. We could not find an API to request a picture of the actor, so we print some information that is available directly on our dataset, which is date of birth and of death.

References

[1] Max Franz et al. “Cytoscape. js: a graph theory library for visualisation and analysis”. In: *Bioinformatics* 32.2 (2016), pp. 309–311.

[2] Joshua Hamilton. *Visualizing High Dimensional Clusters.* URL: `https://www.kaggle.com/code/minc33/visualizing-high-dimensional-clusters/notebook`.

[3] Ulrike von Luxburg. “A Tutorial on Spectral Clustering”. In: (2007). DOI: `10.48550/ARXIV.0711.0189`. URL: `https://arxiv.org/abs/0711.0189`.