IGR204: Milestone 5

Final report

Group K

Beroule, Pascal

Bertrand, Lauriane

Mankai, Yassine

Waly, Mohamed Rached

Wei, Xiang

git repository: https://github.com/rachedwaly/Movies-DB-InfoVis

I/ The data set

The data set that we chose was described as “Three decades of movie data (1986-2016), scraped from IMDb using Python.” It is downloadable in the format of one csv file containing 15 columns. It is a standard movie database showcasing mainly information relevant to consumers. The different attributes are interesting from the point of view of classification and sorting in a sense where patterns can be established by examining one dimension but also by crossing different ones.

Our dataset has 6820 entries and the following attributes:

Dimension	Content	Variable type
Budget	The budget of a movie. Some movies don't have this, so it appears as 0	Quantitative (absolute)
Company	The production company	Qualitative (Nominal)
Country	Country of origin	Qualitative (Nominal)
Director	The director of the film	Qualitative (Nominal)
Genre	Main genre of the film	Qualitative (Nominal)
Gross	Revenue of the movie	Quantitative (absolute)
Name	Name of the movie	Qualitative (Nominal)
Rating	Rating of the movie (R, PG-13, etc.)	Qualitative (Nominal)
Released	Release date (YYYY-MM-DD)	Quantitative (absolute)
Runtime	Duration of the movie	Quantitative (absolute)
Score	IMDb user rating	Quantitative (Relative)
Star	Main actor/actress	Qualitative (Nominal)
Votes	Number of user votes	Quantitative (absolute)
Writer	Writer of the movie	Qualitative (Nominal)
Year	Year of release (YYYY)	Quantitative (absolute)

II/ Target users:

A good data visualization serves for inspiring people to find out what questions to ask and what answers they can get. Our design can be used to search for a specific question. But more, we want users to come without a question or a target, but after playing around with all maps, they will get some interesting ideas and learn more about our dataset.

Overall, Cinema is a mainstream consumable product. But, by exploring the different dimensions a movie database has, we can see that it is far from being mainstream knowledge. With that comes the idea that anyone who ever watched a movie may want to explore new ones and the choice would be mainly based on similarities. You liked some actors in one movie, you probably have more tendencies to look for their other appearances. Similar mechanisms happen with directors and companies but also financial and critical success. Also, crossing different dimensions may refine the search. With our visualisation, we're targeting mainly these moviegoers whether they are casual cinephiles or diehard fans of the industry who are in search of expanding their knowledge or in a quest to confirm a piece of information.

Back to our specific movie dataset, our users can understand more about the general trends in movies. Instead of looking for a specific movie or director, our magnets serve for attracting many items that meet several criterias. Our users don't need to have a clear question in mind before accessing a magnet map. They could just come to the page and follow the instructions.

The requirements in terms of visualisation tools experience varies in our design. The word cloud view only uses sliders and text input as interaction and the display is a set of words with an on demand tooltip to add details. The size encoding should be intuitive as it reflects accuracy. This view does not expect much from the user when it comes to interacting with the system or to decrypting the different used encoding. The magnet view is a little bit more complicated. The interaction is at its core still simple as we're also using sliders and text input to create filters and even the drag and drop feature of magnets is not difficult to manipulate. The encoding on the other hand needs more experience that can be gained by experimenting directly with the system but that should take time.

III/ Representative tasks:

The user basically aims to establish similarities between different entries of our data set. With our design, we expand the idea of entries' similarity to accept various levels. For example, when working with the magnet view, nodes that belong to the same cluster are 100% identical with regard to the complete set of filters but also the user can distinguish a partial similarity between two clusters having one or more filters in common. For the cloud view these levels of similarity are perceived through the font size of the words.

In terms of representative tasks, we considered mainly discovering and enjoying. Since our users include all the spectrum of moviegoers, they do not obey to a pressing professional or personal need, their encounter with the vis is based on pure curiosity and joy. The target is usually unknown to the user and results usually come in numbers. A detail on demand feature is implemented but it is best used when the search is sufficiently refined with filters. By browsing and exploring the dataset, the user can identify patterns and compare clusters.

Multiple questions helped us to create our design:

- Who are the users?

  The users of our design are mainly movie goers and movie enthusiasts.

- What do the users need?

  Exploration of the data set and learning new facts about the movies.

- What is the simplest way to present the key data?

  We decided to go with 2 ways to present our key data. One consists of a cloud of words showcasing the names of the movies and the second one is the magnet view where we give the user more freedom to classify our entries.

- Do we actually need an interactive form for our visualization?

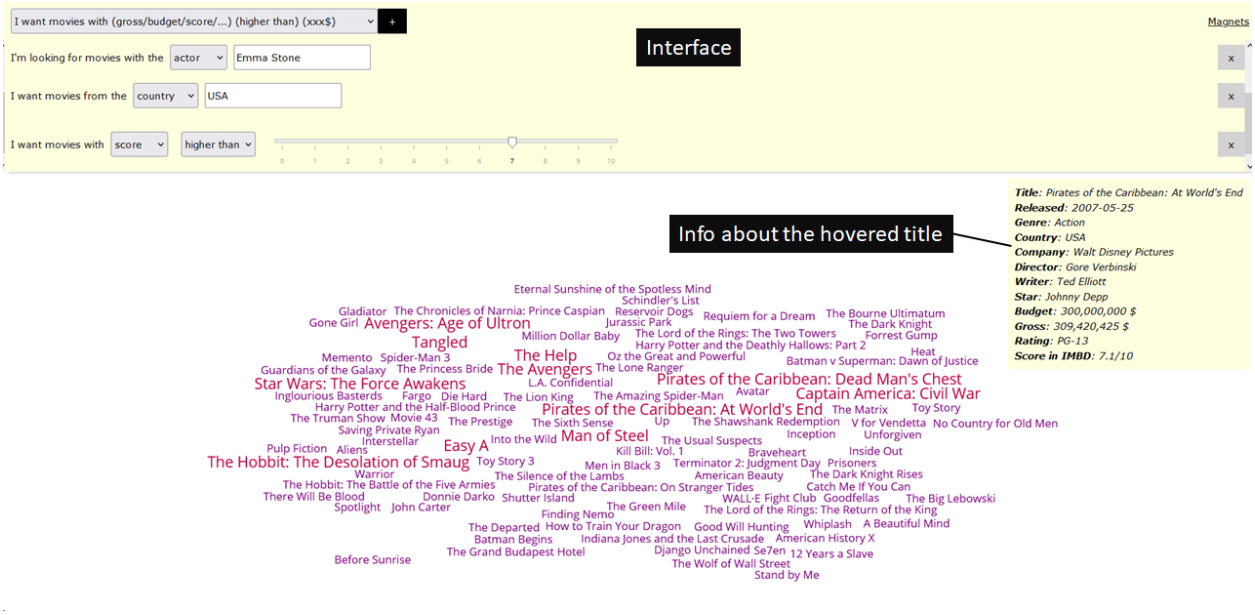
  Having a static visualization without any kind of interaction can be boring and will not get the user's attention plus with no interaction assimilating the facts would

be much harder to find. So we decided to involve the user in our design. The task of the user consists in filtering and clustering the different available entries.

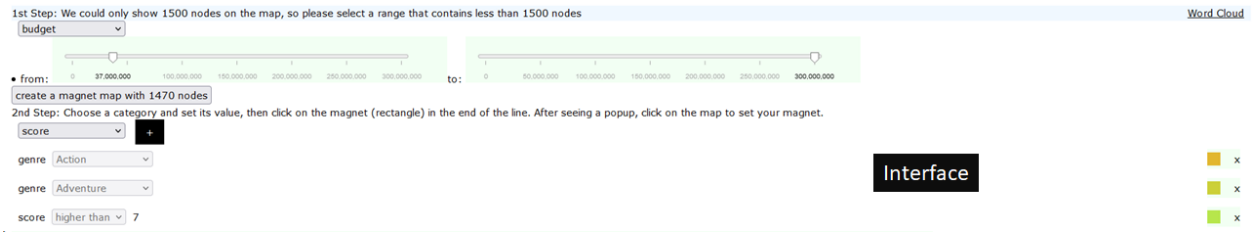
In the following part of the report we will get to the details of our design and the different choices that we made.

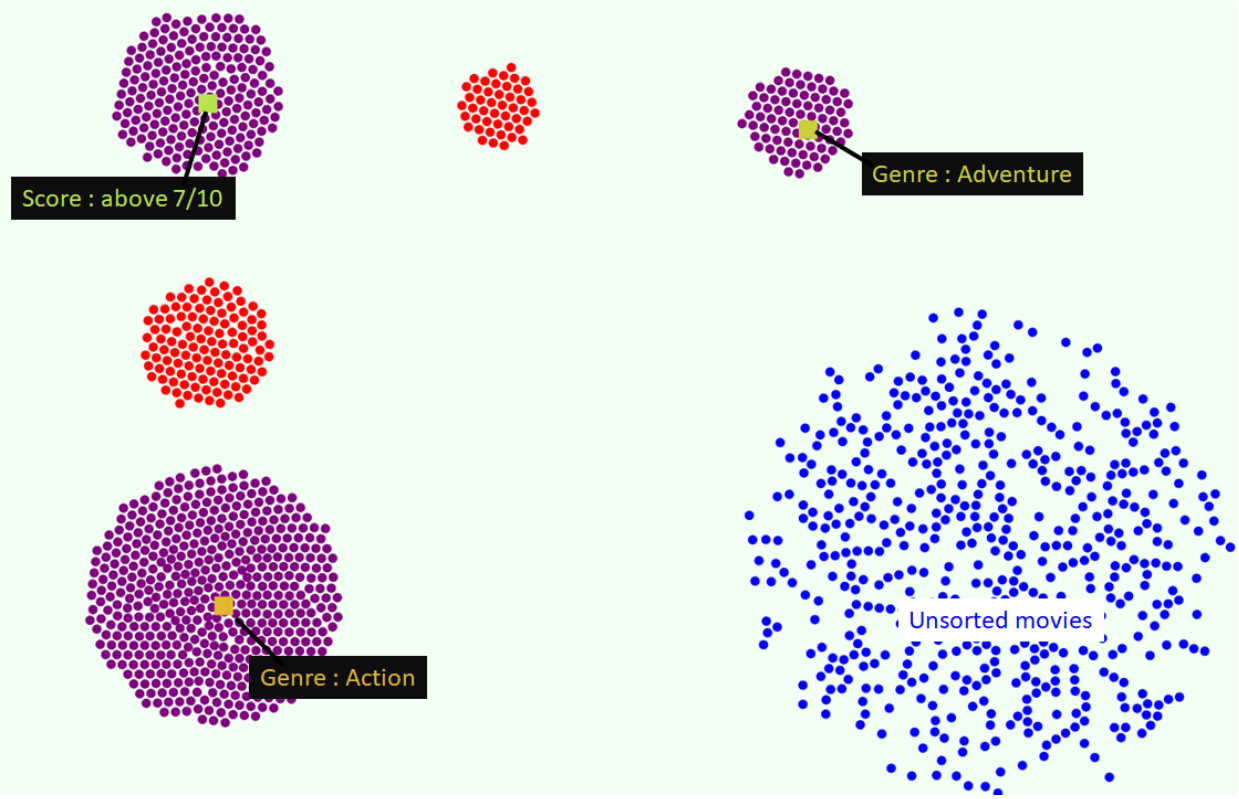
IV/ Final design:

For our final design we chose to merge two of our initial designs. In fact, one of the designs can be hard to manipulate for non-scientific users. So we came up with two different designs and the user can switch between both modes. That's why our final design has the word cloud view and the magnet view. Both of the views work with a filter, to select interesting data. You also can merge multiple filter options to select with more precision. We should note that this choice was also based on a thematic similarity between the two views. In fact, the main idea of our visualisation is to make the insight to be gained feels less like an exposition and more like an interesting byproduct of a fun experience. And, both views serve perfectly this goal as the cloud word is in its essence a form of art that the user helped shape its final state and the magnet view is an interactive system that could be viewed as an open game where the score is maximized according to the amount of knowledge the user got on the data.

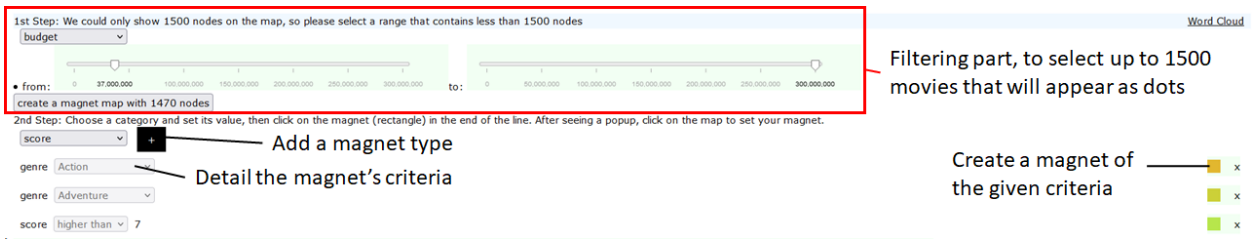


The word cloud view displays the title of the movies that corresponds most to the filter specification. The bigger the title size is and the closer to red the color is, the more the movie matches the criterias. The originality of our word cloud is the filter area. To make it more attractive we chose not to create a simple filter. To add criterias, the user needs to select a premade sentence and fill in the empty space. Thanks to this, the filter space is transformed into a small story. In our implementation, we only created 3 preset sentences so it doesn't really look like a story but the idea is to have a lot of choices so that the user can create his own story. Thus, the main interaction between the user and the view is the filter. We want the user to have a great time creating the story because there is no exploration with the output. However, as the main goal of the view is to highlight movies depending on the criterias and bring more information about them, we added an interaction with the word cloud, by hovering the title to display more information about the movie. Other interactions such as adding, deleting and modifying sentences are also possible.





About the magnet view, the idea is to create clusters of points (one point represents a movie), depending on the magnets they're attracted to. As we wanted to make it as understandable as possible even for non-scientific users - because the target user is a lambda person interested in movies - we added colors to simplify the understanding of the output. The points go from blue to red depending on how much magnets they are attracted to. To use the entire red-blue scale and keep the clusters separable in color, we decided to normalize this color mapping using the maximum magnet per node number. That way, the movies that are attracted by the most magnets appear red, the ones that are not attracted by any magnet appear blue and the ones that have an intermediary number of attracting magnets appear accordingly. While the fun exploration was in the filter for the word cloud view, here the fun is in the output area. The user can freely add, delete or move magnets on the board, to look for similarities or differences in movies. The basic interactions are present as well, such as filtering interactions and the display of information by hovering points. One thing to notice is that we encountered a limitation in the number of points that could be displayed at one time. To deal with this limitation, we added a mandatory filter option to firstly reduce the number of movies displayed on the screen.



Details of the magnets view interface

The target user is a random person interested in movies. He wants to use our design to explore the data in a funny way to find interesting information about the topic he's interested in. So, the two main points we focused on were 1) the fun the user has 2) the exploration. In our opinion we fulfilled those two goals, as we transformed the exploration into a game, either by creating a story in the word cloud view or playing with magnets in the magnet view. Moreover, we give a lot of freedom to the user (as many filters as the user wants, manipulation of magnets,..), which goes well with the idea of exploration. Then, we needed to fulfill the goal of the user: learn something about the movies. Indeed, playing is great but useless if the user doesn't learn anything. That's the reason why the user can retrieve information about the movie by hovering the point or the title as well as learning information by looking at their size, color, position,...

Our design is very good at piquing the users interest and making them feel free to manipulate the data as they wish. The magnet visualization is highly flexible and allows the user to quickly create multi dimensional custom visualizations. Most of the tested users really enjoyed the simple fact of moving magnets around and trying to find movies they know while learning about new ones they might like. It's also very good to find unexpected patterns in the data while playing around with it.

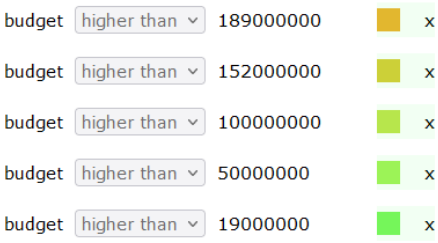
We would also point out the fact that the pipeline that we conceived to create this type of visualisation can be adapted to any kind of data. The magnet map can be easily adjusted to any dataset. For a single dataset, we just need to filter out less than 1500 pieces of data and we can show each piece of data as a circle on the map. When we try to filter a category based on the dataset attributes, we just need to return the id of the data who meets the filter criteria. (Ideally, we can use python or some tools to handle big data problems to define how we can narrow our data into 1500 pieces.)

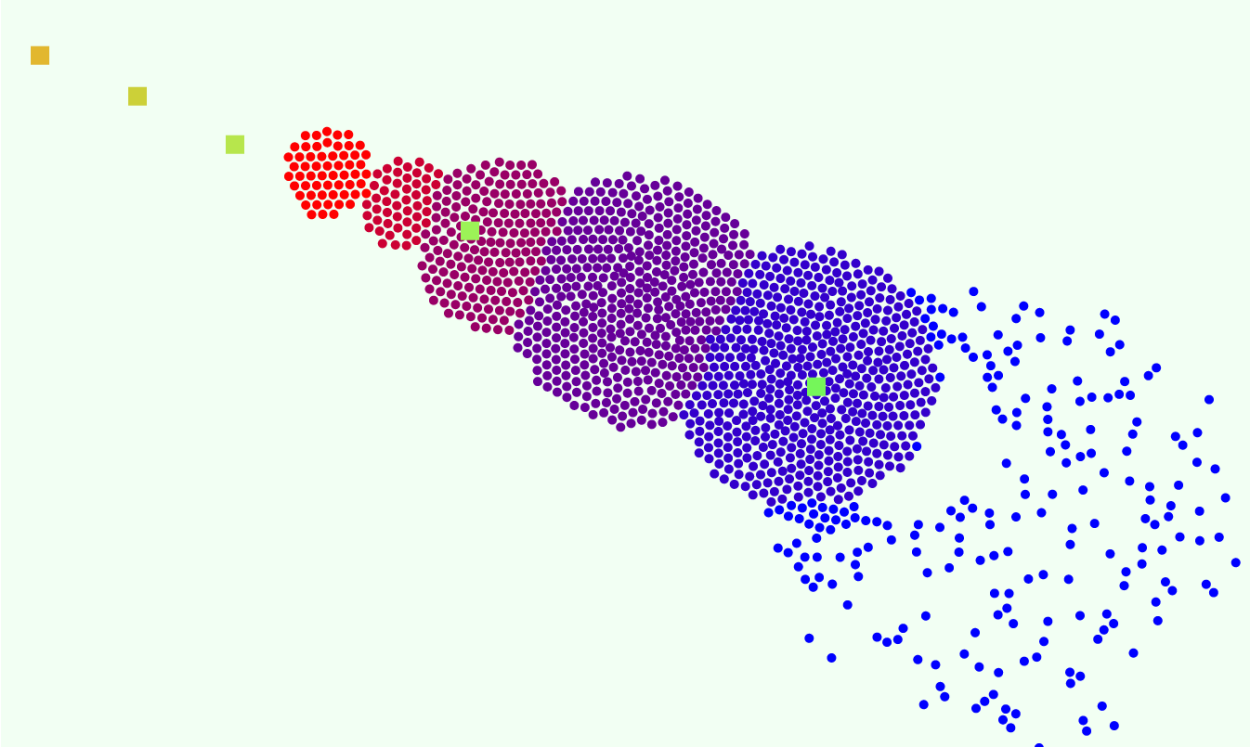
Moreover, the users can choose one of the views depending on their own taste. It makes the whole design even more flexible as one might prefer the word cloud view while the other will use the magnet view.

Finally, our design is easy to understand and manipulate so that it is usable by the general public.

The example below demonstrates how the magnets can be used to display a movie gradient depending on the budget. In the original design, this was supposed to be doable

using only one gradient magnet but that wasn't possible because of implementation specificities. It can still be done using several binary magnets.





However, our design is not built to find really precise statistical information. That's why it wouldn't be suited for a professional use or for users that are interested in statistics (for instance, comparing the entire database on one specified variable). Though our target users are not professionals so it was a choice not to show statistical information. Another bad point in our design is that it can't show the entire database. While the magnet view can display around 1000 points, the word cloud view can only show around 100 titles. Thus, the user can only have access to a restricted overview of the data.

For the word cloud view another drawback is that the sizes and colors of the titles are hard to compare. When there's too many filters the gaps between two sizes and two colors are really small. Moreover the clustering is not as clear as in the magnet view.

Indeed, if you have 3 filters the movies which fulfill the criterias 1 and 2 will be displayed the same way the movies fulfilling criterias 2 and 3 are displayed.

For the magnet view, one drawback is that focusing on the number of data displayed on the board led to a harder time to retrieve a specific movie. In fact, as all movies are points it's hard for the user to know which point corresponds to which movie and compare specific movies. Also, while the word cloud displayed the most interesting movies for given criterias in the ENTIRE database, the magnet view filters only on the smaller database you selected at the beginning. Thus, it's not possible to study the entire database.

A second limitation in the magnet view case comes with the different encoding choices. The magnets are represented by colored squares (using a color palette between yellow and blue to avoid confusion with the nodes color palette). This encoding using the color for the magnets does not lead to a rapid identification of what filter does each magnet correspond to. This issue is due to the intrinsic property of colors as the human eye can't really select a color without proper processing and also to the necessity to go through a second step and read from the magnet table above the playground. This issue could not be avoided in the context of this project because adding a details on demand feature to the magnet will cause issues with the one implemented for the nodes.

Even though our design has limitations you can observe that most of the drawbacks we encounter in one of the views aren't present in the other view: filter gaps problems corrected using clusters in the magnets view, restriction in the database of the magnet view not present in the word cloud view,... Thus, both views are complementary and merged together gives a strong design where the user can switch views depending on what drawbacks he considered to be more problematic.