PROCESS BOOK - TEAM JS

Hit Artist Analyser

Baudoin von Sury d'Aspremont
Jonathan Besomi
Julien Salomon

May 28, 2020

Contents

1 Introduction

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. As part of the class data visualization, we had to **select a dataset** and then **propose a visualisation** showing **insightful information** from it. This Process book presents our project development taking into account all the different steps. It shows the choices we had to take, the challenges we faced and the successes we accomplished.

2 Project Genesis

Looking at the available datasets, our team was from the beginning interested in working on musical data. No dataset or Kaggle proposed issues that inspired us at first. Then, looking online we found the project *RapGenius 2.0* presenting a fun mock-up of a platform that takes rap artists and analyzes the musicality and the lyrics of an artist (Eminem in the example). The **in-depth analysis of one artist's data pleased us**. Since we, as students, are used to working with large datasets, we liked the idea of extracting information from a small-scale pool.
The initial idea was thus to take **all the songs recorded by Michael Jackson** (MJ during the rest of this report), and analyze **their musicality** and **their lyrics** to get an in-depth visual description of MJ's career: how did his music evolve through time, what did his songs talk about, how are the lyrics and the musical features of his songs are related?

3 Data Scrapping

Given the specificity of our project, we had to **scrap the necessary data**. Here are a few lines to describe the tools and the methods used to scrap the data.

- **The Spotify API:** using the well-documented *Spotify API*, and its Python client *Spotipy*, we were able to get all the songs and their musical features (more on that in 4.1) from his or her albums in a csv file. The "*spotify/scraper.py*" file in our GitHub repository can generate the csv file for any artist. You can use this file by filling the ARTISTS variable at the beginning of the file. It is a list, and the result will be one csv file per artist given in the list.
- **Genius API:** using the Genius API, we were able to get, given an artist, all the songs and their lyrics in a csv file. The "*lyrics/scraper.py*" file in the GitHub can generate the csv file for any artist. You can use this file by filling the ARTIST_NAME and the ARTIST_ID field. (The ARTIST_ID is the artist's name with a capital letter at the beginning else all lower case, with a "-" instead of space. For example, Michael Jackson's id is "Michael-jackson").

Merging the two datasets was a complicated task, as it was difficult to find a match between the titles. We generated a new column in both datasets called "*clean_title*", where **the titles of the songs were simplified to augment the likelihood of a match**: special characters were removed, the titles were set to lowercase, anything between parentheses or behind a "-" was deleted as it was almost always the mention of a *Remaster* or a *Remix*. We then were able to join the two Dataframes on this "*clean_title*" column.

We noticed some lyrics missing from the lyrics Dataframe, and some albums missing from the Spotify Dataframe, without any reasons. Some more updates can be done on the scrapping to ensure that all songs have been pulled, but as it was not the emphasis of the project, we added the missing songs manually and kept on going.

It is worth mentioning that the data scraping and cleaning took us a long time, and, while we were aware that this aspect of the project will not affect our final grade, we are glad to have pulled through, as we are proud of our final result.

4 Data Analysis

Once the data was pulled, an **exploratory data analysis** was done on both datasets.

4.1 Audio features

Here are the available audio features from the Spotify API:

- **Key**: estimated overall key of the track.
- **Mode**: determines if a track is in minor or major.
- **Acousticness**: score between 0 and 1, defines how much a track sounds acoustic (not electronic).
- **Danceability**: score between 0 and 1, based on tempo, rythm stability, beat strength and overall regularity, defines how suitable a track is for dancing.
- **Energy**: intensity and activity of the song; the songs with high energy feel fast, loud and noisy.
- **Instrumentalness**: score between 0 and 1; close to 1.0 if there are no vocals in songs.
- **Liveness**: score between 0 and 1; close to 1.0 if there seems to have an audience (live recording).
- **Loudness**: the average of the track in dB.
- **Speechiness**: score between 0 and 1; indicates the proportion of spoken words in a track (compared to sung word); can indicate that the track is an interview/podcast or even presence of a narrative in a track.
- **Valence**: score between 0 and 1; indicates the musical positiveness conveyed by a track.
- **Tempo**: the overall tempo of a track in bpm.

4.2 Lyrical features

The lyrics were processed using the Python library *Texthero*, created by our own Jonathan Besomi (please add a Star!). This library allows any user to pre-process a text in a few lines of code, using techniques such as stemming, removing stop words, blanks, punctuation... From this, multiple features were extracted:

- Information on an artist's **vocabulary**: words per song, vocabulary size, top spoken words in terms of count...

- Information on **the sentiment analysis** of each lyric of the songs (score between -1 and 1, indicating how positive/negative the lyrics of a song are).
- Information on t**hemes of each lyrics** has been tested using the *Empath* library. This library is not very efficient when analyzing large corpora. This has thus been dropped.

4.3 Clusters

To compare the features with each other, **clusters** were made in the following manner:

- Cluster the tracks **given all their musical features**. Then, for each cluster obtained, observe their lyrical features.
- Cluster the tracks gave **their lyrical features**, and observe each cluster's musical features.

For each cluster, no truly interesting information was observed. We thus decided to drop this idea.

5 Story

After having explored the data, we realized that using only MJ's tracks, it would be difficult to tell an interesting data story. We thus decided to select 5 artists, each iconic, having marked their music style: **Frank Sinatra** and his jazz influence, **The Beatles** and their British rock, **Michael Jackson** the king of pop, **Eminem** incarnating modern rap music, and **Rihanna** the typical pop star of the 2000s.

We were thus interested in **visualizing the features of the discography** of each of these artists. What do they sound like? How do they compare to one another musically or lyrically? Are these artist's linked somehow? We thus wished to show how different music can be and what links all these musical artists.

6 Visualisation

6.1 Initial Plan

A brainstorming session was organized, where we tried to imagine as many visualizations as we could. We took inspiration from previous work done on music analysis (thepudding.com), but also from different websites regrouping visualization (d3-graph-gallery.com and public.tableau.com for example), as well as the slides of this course (especially for the text visualization).
After 3 meetings, we ended up with the following precise visualization ideas:

1. With Spotify, we have multiple audio features described in part 3.1. Due to the number of features, we decided that **a radar chart** for each artist, representing the median value of selected features for each of their songs.
2. To compare the artist's audio feature, a **larger radio chart** would show the mean (or the median) of each feature for each artist. Buttons listing the name of the artist would be clickable to show or not a specific artist, allowing a user to compare many combinations or artists.

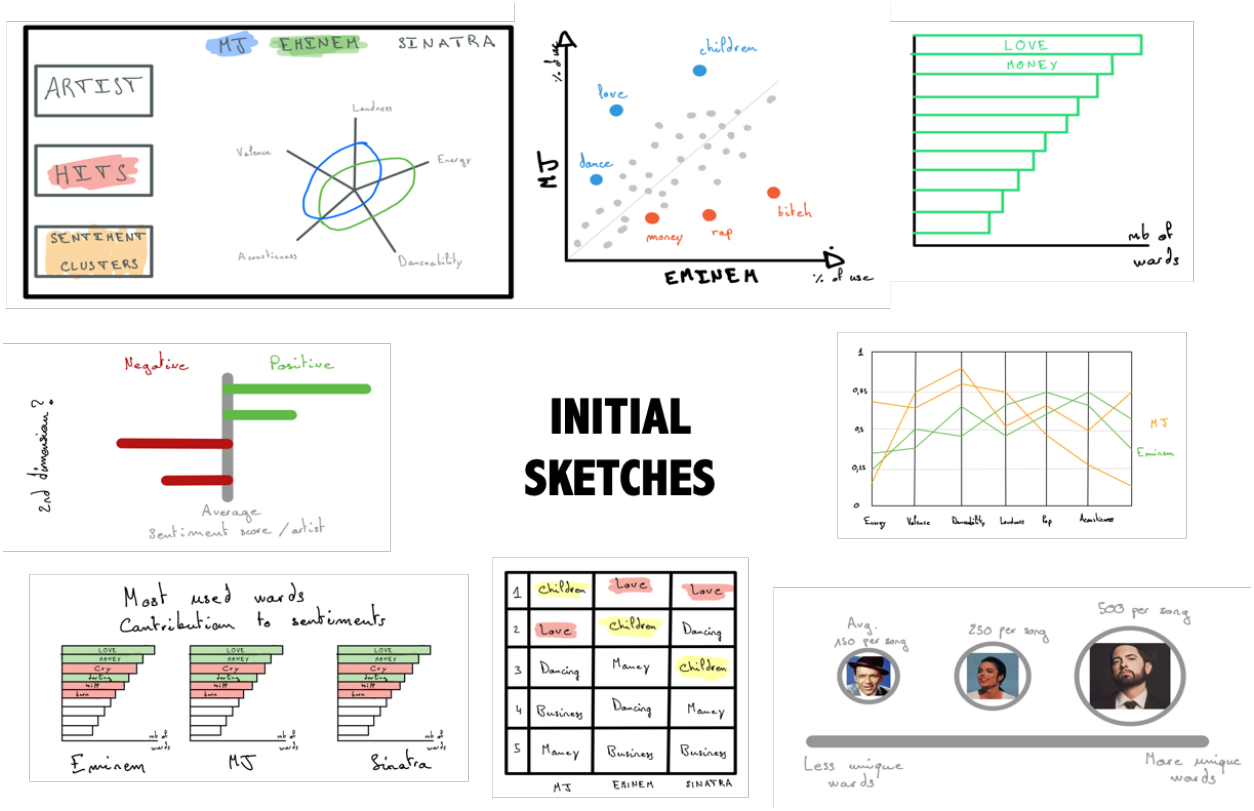

Figure 1: Example of few sketches after brainstorming

3. After visualizing and comparing audio features, artists would be visualized by their lyrical features. For each artist, themes and general mood of their songs would be shown: **word clouds** seemed to be appropriate here, with positive words highlighted in green, negative in red.

4. Vocabulary statistics would be visualized with **simple bar plots**, highlighting the size of the lyrics as well as the vocabulary of each artist. The most used words would be used to compare the artists' top songs and see how representative their hits are to their general lyrics.

5. Artists would be lyrically compared by **scattering their most representative words against each other**. A simple manner would be to select top words of each artist, and top words that both artists say, against each other, 2 by 2.

6. All tracks would be **clustered** given the audio features mentioned. Each cluster would be labelled given **its representative audio features** (example: high danceability, low acoustics). The features of the labels will be plotted using a bar plot showing the median of the value of each feature for each cluster.

7. A 2D representation of the songs in a **latent space**. The idea was to plot and observe how similar songs are related in the vector space. We would have used a scatter plot where each point will be a song (lyrics) and we will display the change in the lyrics between the songs.

8. Sinatra has multiple tracks featuring Quincy Jones. Quincy Jones produced most of Michael Jackson's hit. We wanted to show how correlated are the lyrical and audio features of Sinatra's songs featuring Quincy Jones, and MJ's songs produced by Quincy Jones. This would be highlighted plotting a **heat map** highlighting the correlation (or not) between all those features.

9. Sinatra sings 2 Beatles songs: *Yesterday* and *Something*. How do these songs compare musically? Re-using the **radio chart** from the first points seemed to be appropriate here.
10. Eminem and Rihanna have multiple songs featuring each other: *Love The Way You Lie (Part 1 and 2)*, *The Monster* and *Numb*. Musically, who's influenced is sensed more (**radio chart**)?
11. Rihanna samples MJ's song *Don't Stop 'Til You Get Enough* in *Please Don't Stop The Music*. How does the original song influence Rihanna's song (**radio chart** as well)?

6.2 Chosen Visualisation

Once we had all those ideas, we had to select a few of them to create our story. To select which visualization we will implement, we based our reflection on the following criteria :

- **Show insightful information** of our dataset
- Original visualization
- Interesting to build a story around them
- **Feedback of Milestone 2**

We then ended up with three final visualization :

1. A **Cleveland dot plot** to show for each artist **the median of the audio features** with an **interactive interface** where the reader can click on buttons to switch artists.
2. A multi-column visualisation **to show the sentiment of each song** and the top words associated with an interactive brush to select specific songs. We were inspired, for a final result, by Tableau community visualization of the Beatles' songs
3. A line chart with:

   (a) On the x-axis, the **selected audio features**: acousticness, danceability, energy, loudness, tempo, valence. The value of each feature was normalized, so all of them can be contained between 0 and 1.

   (b) On the y-axis, values ranging from 0 to 1.

   (c) **Confidence intervals of the mean** of each feature of all the song for a specific artist.

   (d) **Lines** representing the feature values **for a specific song** are added on top of these confidence intervals. This will allow us to **situate the audio features** of a specific song over the values of one or two artist's career.

   Three charts of this type are used to show the points **9, 11** and **12** of the visualisation ideas section (6.1).

4. **Word clouds** for each artist where the size of the words is proportional to the number of words in their vocabulary.

6.3 Implementation

1. For the **Cleveland dot plot**, we used d3 to implement it. Basing our implementation on the Cleveland dot plot of the d3-gallery website, and on the *update* function of the animated lollipop chart of the same gallery. To be able to select different artists to compare.

2. For the **multi-column scatterplot**, we based our implementation of the simple scatterplot from the bl.oks website. On this scatterplot, the songs were sorted along the y-axis representing the positiveness/negativeness of the lyrics of a song, based on the sentiment analysis done previously. An **interactive brush** was then implemented for this scatter plot, to select specific songs. A **multi column list** is created from scratch and is linked to the songs selected by the brush, and **prints the top words** of the lyrics of the selected songs, and the **number of time they appear**.

3. For the line chart, we based our implementation on the d3-gallery line chart with confidence interval. From there we added the confidence intervals we needed to plot, added the **songs we wanted to situate** in respect to the confidence intervals, and **deleted the mean for each confidence interval** as these lines crowded the graph. Each of these plots was adapted given the three visualizations we wanted.

4. For the word clouds, we created different them using top-words for each artist using the tool wordart.com.

7 Website

7.1 Design

It is important to have a good design for the website itself, **to package our visualizations**. We decided to spend some time to **create a mock-up** to have a clear idea about design elements and the structure.
To create the mock-up, we first focused our efforts on **the structure of the website**. Where the different blocks takes place? Do we include a banner, a menu?

Once we had an idea of the structure of the website, **we focused on the design**. We all agreed to design a **dark theme** website as it is usually nice and fits well with the music universe.

We first looked for documentation about how to design a dark theme website and then we started to create the different elements of the website.

We also selected a **color palette** for the entire website. We selected five different colours for the five artists. Putting all of this together, **we created a full mock-up** using PowerPoint.

7.2 Website Implementation

To implement the website and to code it efficiently, we first developed the banner, the comment block, the buttons and **the different data-visualization widget independently** using d3, Javascript, HTML and CSS and finally merged it to obtain the final result.

Some interesting implementation about the website:
The banner : As our project is focused on music analysis, we wanted to implement in our website a way to play the different songs of the different artists. We, therefore, created a banner which integrates a platform where the reader can **play samples of the top songs** of the different artists we are highlighting (using the Spotify API).

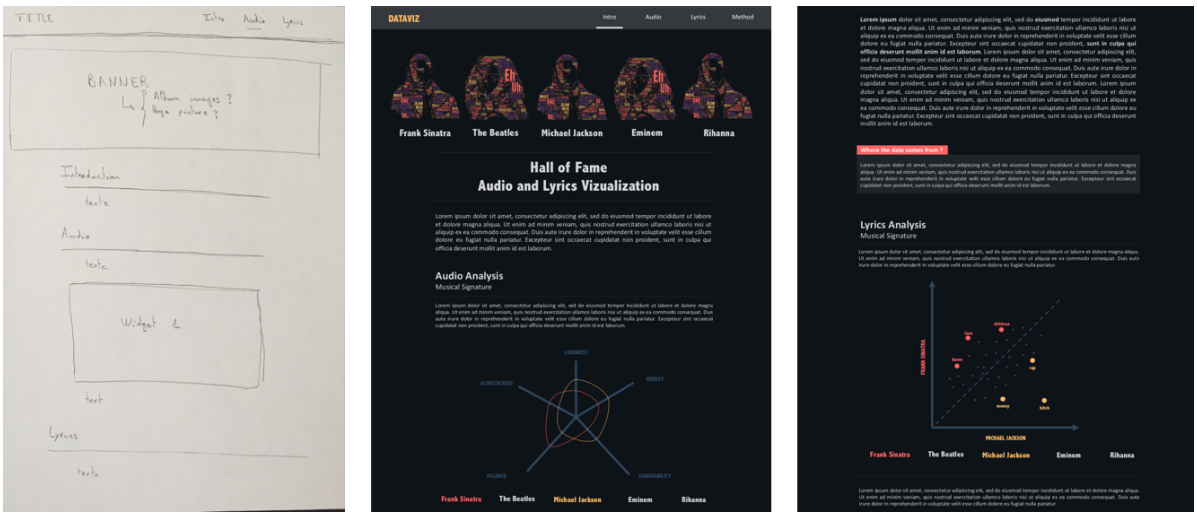

Figure 2: Example of few sketches after brainstorming

8 Peer Assessment

- **Brainstorming and Processbook** Everybody
- **Data scraping** Julien and Jonathan
- **Lyrics EDA** Jonathan
- **Spotify EDA** Baudoin and Julien
- **Data Story** Julien
- **Website structure and design** Baudoin
- **Visualisation using D3** Baudoin (Cleveland dot plot), Jonathan (Multi column scatterplot and list), Julien (Line chart)

We thus gave each other these roles in the project:

- **Jonathan Besomi**: NLP and JavaScript expert, our real text hero.
- **Baudoin von Sury d'Aspremont**: Web designer and Mock-up artist, the eye of our group.
- **Julien Salomon**: Story teller and analysis leader, the creative mind of the group.

9 Conclusion

The final result is our beautiful website ***hit-artist-analyzer.now.sh***. The visualizations we managed to implement allow us to show insightful information that corresponds to our original goal.

Our main **self-criticism** comes from the fact that we had some trouble linking the textual and audio analysis. Our clustering idea explained in 4.3 was good, but the focus on visualization did not give us the time to analyze this as deep as we could have.

Nevertheless, the audio and lyrical visualization complement themselves quite nicely, and the fact that most of our plots are interactive render our final product very easy and, we hope, **fun to navigate**.

Thank you for reading our process book.