Visual Analytics project: an interface for the recommendations of songs

Alessio Parmeggiani, Alessio Sfregola

October 16, 2022

Abstract

One of the most important characteristics of today's most popular apps are the powerful recommendation algorithms that let users find out new stuff they may like. May it be news, videos, images or what their friends post. The same happens for music, and Spotify, one of the most popular apps between music lovers, is no exception. The problem is that these recommendation systems are black boxes, they just take users' behavior as input, and output what they think they may like. This certainly makes the experience much easier for the user, but also doesn't them exploit all the potential of such complex systems. Our goal with this project is to create a visual interface for such an algorithm, for users to interact with, which let them explicitly express their preferences and explore the recommended results. We decided to use PCA as a very simple recommendation system. We use it to reduce the dimensions of the data and compute distances between data items, which allows us to plot them through scatter plots, and compute the distances between them. This distance can be interpreted as a similarity function, so that users can find the songs or artists most similar those they already like. Other than that, the interface includes a search bar, for users to find their favorite artists, a filtering system, to let them express what characteristics they like in a song, and two additional plots (a box plot and a radial plot), to display the values of the attributes for the recommended songs.

1 Introduction

Nowadays almost everyone listens to music in their everyday life, and one of the most popular apps to do so is Spotify. It often happens that after listening to a certain song that we like a lot, we can't find a similar song from the same artist and searching for songs of the same genre does not help.

We then decided to use a dataset the consists of hundreds of thousands of songs from Spotify labelled with many categories that represents more informations about a song, rather than only a genre. These categories are:

One of the main features that make Spotify one of the most popular apps in the world is its powerful recommendation system, which frees users from the need of choosing the music they want to listen to, and lets them discover their next favorite artist. The only problem with this wonderful mechanism is that it is completely transparent to the users, so there is no way for them to interact with it and customize it to their own taste. This is where our project comes into play. Our goal was to build a interactive tool for users to discover new music, without the powerful black box mechanism used by Spotify, that could also let them grasp new insights on the most popular music at the moment.

Dataset For this project we decided to use a dataset[1] containing data about more than 1.2 million songs extracted using the Spotify API. This data consists of technical information about the songs (such as title, artist name, album name...) but also some interesting numerical data computed autonomously by Spotify. This data consists of values that typically range between 0 and 1 for each of the following categories:

- **Tempo**: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
- **Danceability**: describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

[1] https://www.kaggle.com/datasets/rodolfofigueroa/spotify-12m-songs

- **Liveness**: detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

- **Energy**: is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.

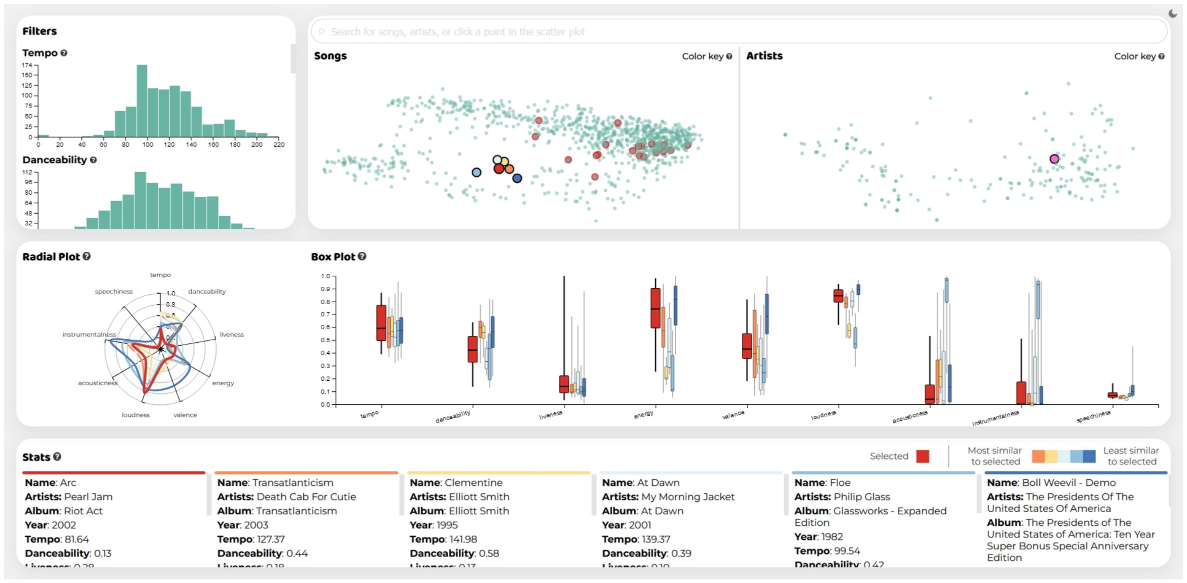
- **Valence**: a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

- **Loudness** of a track in decibels (dB): the values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.

- **Acousticness**: is a confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

- **Instrumentalness**: predicts whether a track contains no vocals. Öohänd äahs̈ounds are treated as instrumental in this context. Rap or spoken word tracks are clearly v̈ocal:. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

- **Speechiness**: detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.



Name: Arc
Artists: Pearl Jam
Album: Riot Act
Year: 2002
Tempo: 81.64
Danceability: 0.13

Name: Transatlanticism
Artists: Death Cab For Cutie
Album: Transatlanticism
Year: 2003
Tempo: 127.37
Danceability: 0.44

Name: Clementine
Artists: Elliott Smith
Album: Elliott Smith
Year: 1995
Tempo: 141.98
Danceability: 0.58

Name: At Dawn
Artists: My Morning Jacket
Album: At Dawn
Year: 2001
Tempo: 139.37
Danceability: 0.39

Name: Floe
Artists: Philip Glass
Album: Glassworks - Expanded Edition
Year: 1982
Tempo: 99.54

Name: Boll Weevil - Demo
Artists: The Presidents Of The United States Of America
Album: The Presidents of The United States of America: Ten Year Super Bonus Special Anniversary Edition

Figure 1: A screenshot of our final interface.

This is the data we use to let users discover new tracks similar to those they already like.

The dataset has too many songs for our visualizations so we decided to select only a subset of them in our visualization in order to not slowdown the software. By selecting only some songs at random, we could encounter the problem of having most of the songs and artists unknown to most users, so we selected, using a list of the top 1000 artists in the Spotify platform, only the tracks belonging to artists in this list and we ended with a lower number of songs. We still selected only a subset of these songs to have a faster visualization.

To do so, we reduce the dimensionality of this data using Principal Component Analysis (PCA), compute the distance between the resulting data elements and find those closest to the users' input. We also give users the ability to fine tune their queries by allowing them to filter out songs that have values for some of the above categories that fall out of user defined intervals. An user of our system can search for its favourite songs and find similar songs based on these categories. The user can also identify artist that are similar to its favourite artist, and this similarity can be analyzed using all the available categories, or only the ones chosen by the user. Our complete interface can be seen in Figure 1.

2 Related works

[MHJV18] performed a study on which type of controls are preferred by the user in an application similar to ours; in fact they also used as study case the recommendation system of Spotify and made available two different type of controls for the user to choose the best recommended track, these are a radar chart and some sliders. The authors found out that the radar chart has been more useful than the sliders in the selection of the preferred parameters so this was also one of the reasons we added a radar chart. We included an improved version of the sliders to try to increase their effectiveness, instead of selecting a single value for a category, we select a range, and the distribution of data for the category is always shown.

A similar approach was also used by [MHCV19] with the goal of finding a correlation between the personal characteristic of a user and the songs or artists they choose. While their goal is different from ours the interface that was developed can provide some inspiration. The authors used sliders like in the other paper but also a scatter plot with few data to compare one attribute with respect to another, the two chosen attributes are used in the two axes and the tracks are plotted using their values.

The choice of a correct type of different representation for our data was not trivial and we encountered many possible designs. The analysys of many available designs for a radial plot studied in [DLR09] was useful to select the best design to represent all the values related to each category of a song, while the study in [BW14] was really useful to determine when it's better to select a radial visualization.

3 Implementation

Tools&libraries Javascript hmtl css d3js libreria PCA online pandas In the making of this project we used several different tools. We used three main programming languages: *Javascript*, *HTML*, *CSS*; moreover we used also *Python* to preprocess the dataset. Regarding libraries, we made large use of *pandas* in Python for the preliminary elaboration of data, *d3.js* for the visualization of data, a library called *PCA-js* for Principal Component Analysis and *autocomplete.js* for the autocompletion of songs and artists in the search bar.

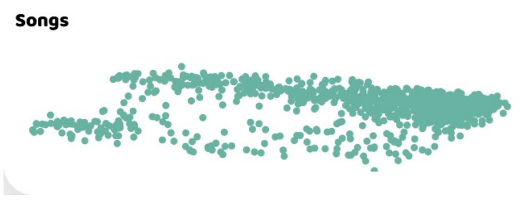
Artists and co_artists On the dataset we noticed that for each track there could be more than one artists, we decided to consider the first artist that appears as the main artist, and the next one as co-artists. In this way we can consider two songs as made by the same artist even if just one of their artists or co-artists are the same.

The disadvantage of this approach is that for some songs there are some co-artists that only contributed to that song so in the artist scatter plot they will appear in the same spot. To overcome the problem of having too many "irrelevant" artists we increased the radius of the dots for the artists based on how many songs an artist made, in this way the user can immediately see which are the most important artists, but the data about the other artists are still fully available.

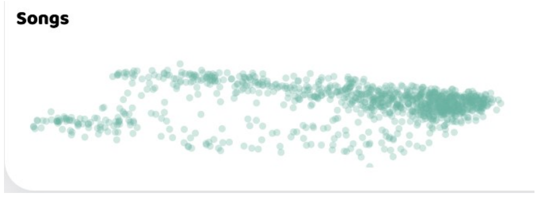
Dimensionality reduction To represent our multidimensional data in 2 dimensions we used the PCA algorithm, in this way the distance between the points is preserved and this algorithm is better suited for our type of data. In fact while another algorithm like multidimensional scaling preserves the similarity between the elements, this similarity is more useful when there are data of different nature that needs to be combined and the similarity function can be custom defined. In our case this is not needed, in fact our data is entirely numerical. Moreover, to get consistent results and avoiding weighting some categories more than others, we normalized all the data using the z-score to scale all the values between 0 and 1 and preserve the closeness of the data, something that could be lost if we used min-max to normalize.

For the artists instead, since the dataset we used contains only data from the songs, we used the mean of the different categories for all the songs made by an artist as the data on which to apply the PCA algorithm.

Scatter plots The most important thing we thought was needed in our project is having an accessible and easy to comprehend interface to view in an instant all the songs and artists. This is why we created two scatter plots to visualize all the data items used in our analysis. The first one displays a dot for each song in the dataset, the second one does the same for every artist in the dataset. Given the large amount of songs we need to represent, we used small and slightly transparent dots to represent a point so that the user can perceive better the distribution of the data. In addition to this, we also scale the size of the dots that correspond to artists proportionally to the number of songs they made.



(a) No opacity applied on the dots.



(b) The dot are slightly transparent.

Figure 2: The importance of using transparent dots for the scatter plot to show the distribution of the data.

The user can select an element by clicking on it. After the selection many information can be extracted by looking at the different ways the points are highlighted. To better show different categories of points after a selection we change some characteristics of a point, such as the radius,the color and the contour, moreover they are brought in the foreground in an order based on the different level of importance given to each highlighted element. The colors are chosen in a way that each of them is clearly distinct from the others. To do this, we took inspiration from color schemes found in https://colorbrewer2.org . The meaning of each of them is clearly explained through a tooltip when hovering the *Color Key* label on top of the scatter plots.

When clicking on a song, the clicked track and the 5 more similar tracks based on the distance on the plot are highlighted. The selected track is colored in red, and the similar songs are colored with using a sequence of color that gets gradually further from red and closer to blue, in order to encode the fact that the selected songs are not equally distant to the chosen one; moreover we also highlight the songs made by the same artist, and in the scatter plot related to the artists we highlight the artist and co-artist that made that song.

Instead when the user selects an artist the interface highlight, as in the interaction on the songs, the selected element and the most similar ones, and in the scatter plot with the songs are highlighted the songs of which the selected artist is author or co-author.

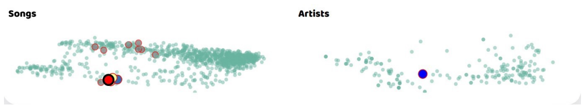


Figure 3: 4 different type of selections.

To better navigate in the scatter plots we added a way to zoom the view, so that the user can better visualize a clusters of point. By dragging the mouse on the scatter plot the brushed area is enlarged. This feature is particularly useful when the similar elements are too close.

Radial plot We needed also to display the relevant information of an artist or song so we added a radial plot updated with the values of all the relevant categories of a song and instead when an artist is selected the plot is updated with the mean of all the values of their songs; so that the user can grasp immediately what are some typical features of an artist.

Box Plot A simple mean of the songs was not enough to represent all the relevant data of an artist so we added a boxplot to show more details for each artist and the similar ones.

Filters To allow the user to choose which category is more relevant we added a filter system for each category. The user can select the desired interval of values for one or more categories and all the song which data is outside those values is deleted from the scatter plot. If an artist does not have any song in the other scatter plot is in turn deleted. A filter is visualized as an histogram so that the user can tell at a glance which values are more common between all the tracks.

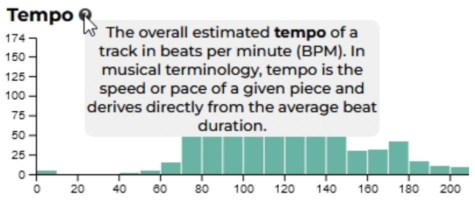
After every filter selection the system updates all its data and selections in the scatter plot, so that the songs most similar to the selected one are shown between only the tracks that are still visible.

Search bar An important thing that we added is a way to search for a specific track or artist at the start, so that an user can start interacting with our environment and search for a song of their liking by starting from a song they know or heard instead than searching for it in the scatter plot.

Other data Finally we added some data with mainly textual information so that the user can select the text of its preferred artist or song and search their song on the Spotify app; here it's also possible to examine the exact values plotted into the radial plot.

Help sections With all these specific terms and informations in the interface an user could fell lost. To avoid this problem we added some help buttons that display useful informations such as the meaning of a specific category or what a panel is doing.

We also added a tooltip when hovering over a dot in the scatter plot that shows the name of an artist or a song, and in the latter case, also its artist.


The overall estimated **tempo** of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.

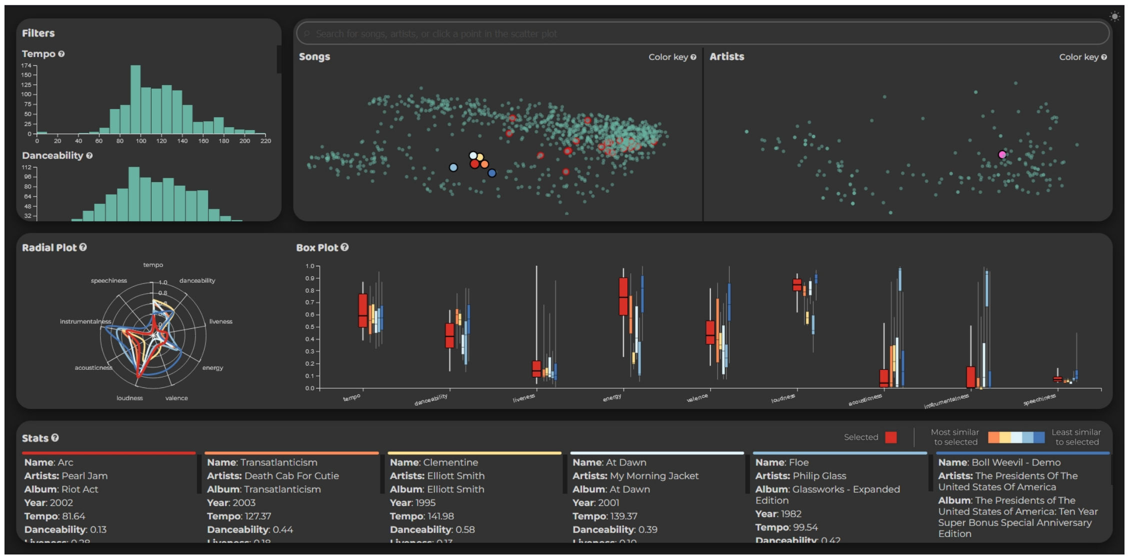
(a) Tooltip when hovering over the help button of the filter section.

Song: Aunt May Packs
Artist: Danny Elfman

(b) Tooltip when hovering on the scatter plot.

Figure 4: Some tooltips that appears on different elements.

Dark mode We noticed that our interface contains a lot of bright colors and could not be ideal when in dark environments, so we added a button to switch between light mode and dark mode, that uses mostly dark colours.



Name: Arc
Artists: Pearl Jam
Album: Riot Act
Year: 2002
Tempo: 81.64
Danceability: 0.13

Name: Transatlanticism
Artists: Death Cab For Cutie
Album: Transatlanticism
Year: 2003
Tempo: 127.37
Danceability: 0.44

Name: Clementine
Artists: Elliott Smith
Album: Elliott Smith
Year: 1995
Tempo: 141.98
Danceability: 0.58

Name: At Dawn
Artists: My Morning Jacket
Album: At Dawn
Year: 2001
Tempo: 139.37
Danceability: 0.39

Name: Floe
Artists: Philip Glass
Album: Glassworks - Expanded Edition
Year: 1982
Tempo: 99.54

Name: Boll Weevil - Demo
Artists: The Presidents Of The United States Of America
Album: The Presidents of The United States of America: Ten Year Super Bonus Special Anniversary Edition

Figure 5: A screenshot of our interface with the dark mode toggled on.

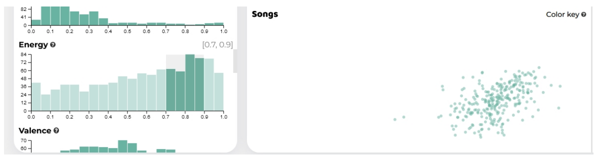
4 Discovered insights

First of all we noticed,especially when using a lot of data, that songs and artists are grouped into distinct clusters: on the left side of the plot there is mainly classical music that shifts into modern pop music toward the center of the plot; on the right side there are examples of more "extreme" music, hard rock or metal on the bottom and rap on the top.

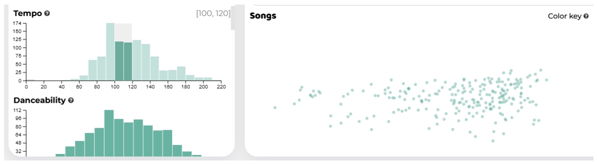
This is mainly due to the fact that the higher variance is preserved by the liveness,acousticness and instrumentalness categories. Almost always to an high value of liveness correspond a low value of acousticness and instrumentalness; these three categories can represent a good indicator of the composition year of the track.

When plotting the artists instead, we noticed that the main factor that account for the most variance is the liveness combined with the instrumentalness and acousticness.

Thanks to the filter section we can discover at a glance many insights on the distribution of data and we noticed how some categories are a lot more important than other in the classification of a song. For example in Figure 6 it's possible to notice how selecting only a subset of values for the *energy* category identify a specific cluster of songs while the same is not valid for the *tempo* category.



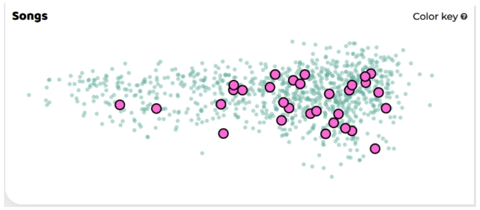
(a) Filter applied on "energy".



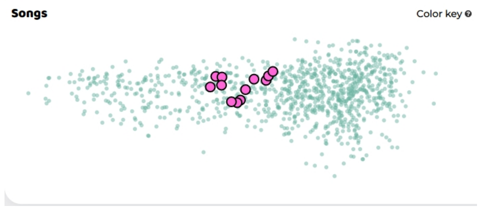
(b) Filter applied on "tempo".

Figure 6: Results after filtering different categories.

Something else that can be discovered at a glance is if an artist often make tracks of a different genre or if they prefer to remain on a specific type of music, these type of insight can be analyzed by simply looking at the distribution of the songs on the scatter plot after clicking on an artist as shown in figure 7



(a) An artist with songs that span across multiple genres.



(b) An artist more focused on a specific genre.

Figure 7: An example of versatility of two artists, in pink the songs made by each artist.

5 Conclusions and future works

A future iterations of this project could see the addition of an integration of the tags of each tracks in the computation of the similarity between the different songs.

Something more advanced that could be included in the future is a way to let the user hear a sample of a song of their choice. This feature could be implemented in a form of a link to a video or to the page of Spotify; something interesting would be to use a more advanced dataset like the one created and analyzed in [mil] that contains also the file audio of each tracks. This dataset contains also other metadata such as the complete text of a song or similarity data between the artists, so it could also be used to create new categories based on the text content and find new ways to recommend similar artists.

References

[BW14] Michael Burch and Daniel Weiskopf. *On the Benefits and Drawbacks of Radial Diagrams*, pages 429–451. Springer New York, New York, NY, 2014.

[DLR09] Geoffrey M. Draper, Yarden Livnat, and Richard F. Riesenfeld. A survey of radial methods for information visualization. *IEEE Transactions on Vi-*

sualization and Computer Graphics, 15(5):759–776, 2009.

[MHCV19] Martijn Millecamp, Nyi Nyi Htun, Cristina Conati, and Katrien Verbert. To explain or not to explain: The effects of personal characteristics when explaining music recommendations. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 397–407, New York, NY, USA, 2019. Association for Computing Machinery.

[MHJV18] Martijn Millecamp, Nyi Nyi Htun, Yucheng Jin, and Katrien Verbert. Controlling spotify recommendations: Effects of personal characteristics on music recommender user interfaces. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, UMAP '18, page 101–109, New York, NY, USA, 2018. Association for Computing Machinery.

[mil]