1 Introduction

Spotify has positioned itself as one of the most formidable online music streaming applications available on multiple platforms [2]. It boasts a large collection of songs from all corners of the globe including podcasts, create-your-own-playlists etc. However, an important feature of the application, that is more useful to Data Scientists, is that it qualifies each of its songs and artists with a set of attributes. Therefore, each song is identified with a quantified attribute that best describes its musical composition and style. This in turn makes Spotify an interesting collection of data to visualize and correlate the attributes to identify patterns and trends. This paper describes the attempted visualization with reference to the data selected, the tasks that it aims to achieve and the technical approach adopted.

2 Data

The Spotify data selected to create this visualization has been retrieved from Kaggle [8]. The data set contains the best songs as quantified by Spotify in the last decade, that is from 2010-2019. A total of 100 songs have been collected in the set. The data set columns can be described as follows. The categorical attributes present are:

- Title: Song title.
- Artist: Song artist.
- Genre: The genre that best describes the song.
- Year: Year of release.

The following quantitative attributes are present in the data:

- BPM: The tempo or speed of the song abbreviated as BPM or beats per minute.
- Energy: A value between 0 to 100 that denotes the level of activity in the track.
- Dance: A value between 0 to 100 that denotes how suitable the track is for dancing.
- dB: The decibel level or the loudness of the track.
- Live: A value between 0 to 100 that indicates the probability of the track being performed live.
- Valence: A value between 0 to 100 denoting the musical positivity of the track.
- Duration: The length of the track.
- Acoustic: A value between 0 to 100 denoting how acoustic the track is.
- Speech: A value between 0 to 100 denoting the presence of spoken words.
- Popularity: A value between 0 to 100 denoting the track's popularity during its time of release.

The data set can be visualized as a single flat table with the title as the unique identifier for each row. The quantitative attributes together describe a song's composition style, which can be visualized in an attempt to relate them with each other. This not only removes the complexity of the data being on different scales (e.g loudness and duration) but also facilitates a summarized view of a song which can be interpreted as its composition style.

3 Tasks

The aim of the visualization is mainly ***exploratory***. It aims to visualize attribute correlations of the songs,especially with the valence of a song. Additionally artist trends like popularity and valence across the decade are also visualized. Also, the interactive features of the visualization aims to help users to find tracks closest to the set attribute values. Some of the most important tasks the visualization achieves are as follows:

- Present the top three songs based on popularity for each year.
- Present and compare the attributes of the top three songs of each year.
- Present the relevance of genres for each year and how they vary across the decade.
- Derive the song with attribute values closest to the combination of the attribute values as set by the user.
- Analyse the trends of popularity of artists across the decade.
- Query the data set to fetch songs with the set single attribute value for a particular year and also present this song's attributes.

Some of the visual encoding channels that are used to create this visualization are as follows:

- Radial Graphs: Radial polar graphs are constructed using slider and knob controllers to display the attribute values for each song. It also serves the task of interactivity and aiding users to change the

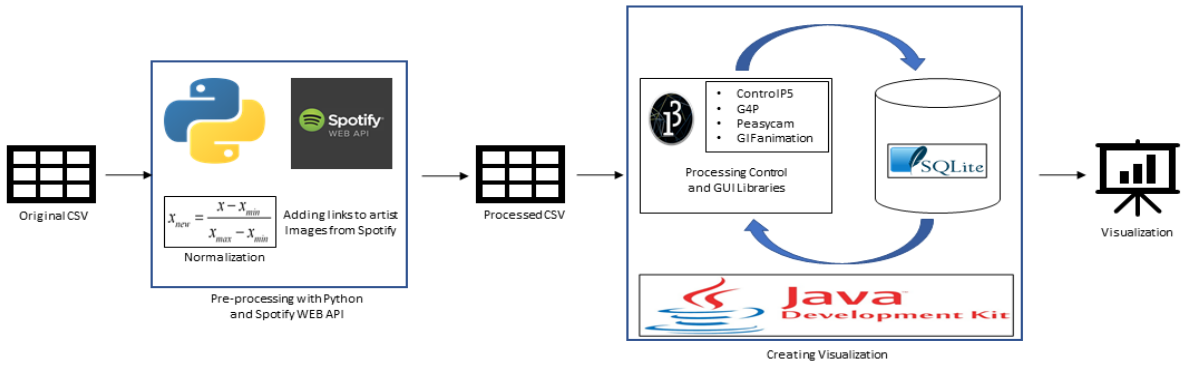

Figure 1: Technical Design

slider values to update attribute levels.

- Size: The genre counts for each year are displayed as rectangles with varying sizes.
- Colour: The year slider is set with a colour gradient to separate the years.
- Lines: Represent the popularity, valence and energy trends for artists across the decade.
- Sliders: Set the attribute levels.
- 2D-slider: Set single attribute levels for particular years.

4 Approach and Technical Design

The design for this visualization includes two broad categories. Firstly, the raw data was processed to fit the features planned for the visualization. The data was cleaned and normalized using python with in built packages like pandas [4] and numpy [3]. Furthermore, an additional column was added to the existing data set which includes links to the artist images. This was fetched using the Spotify Web API [6]. The Spotify Web API takes the artist ID as input and fetches all relevant data in the form of json. After the raw data has been processed and finalized, the data is loaded onto a sample Database using SQLite JDBC driver available in Java [7]. A single table consisting of all the processed data is available in this database, which facilitates SQL querying over them, making the communication between the visualization application and database robust and simple. Finally, the application is built on processing [10] with the help of 2 GUI control libraries: ControlP5 [11] and G4P [5] used to create the interactive features within the visualization. Additionally, Peasycam [9] was used to create the 3D bars and GIFAnimation [1] library was used to create the simple animation pointers in the help mode. An overview of the technical design is depicted in Figure 1.

5 Conclusion

The application developed has been able to provide a visual summary of a song taking into account its various attributes as quantified by Spotify. To achieve this, the correlations between these attributes have been visualized as polar radial graphs, which are also slider controllers themselves. This in turn it facilitates the interactive aspect of the application enabling users to find song closest to the set levels of attributes. On analysing the attributes for the top three songs of each year in the past decade, it can be concluded that songs with higher energy and 'danceability' levels tend to have higher valence levels. However, this is not exclusive. Future work to analyse all data for the decade could help indicate more intricate patterns and correlations between the data. While this application focuses on controlling the visualization to a particular year based on the user's input, to study the trends more intricately, a single view with all the data could prove beneficial.

6 Source Code and Repository

The source code, executable file for the application and instructions on running the application can be found here: `https://github.com/scrntnstrnglr/SpotifyVisualization`. The README file contains instructions on running and using the application. Also, presentation and instructional video on the visualization can be found in this video `https://youtu.be/1kM5a_h0gJ0`

References

[1] Gif-animation. http://extrapixel.github.io/gif-animation/. [Online; accessed 18-April-2020].

[2] Music for everyone. http://https://www.spotify.com/us//. [Online; accessed 18-April-2020].

[3] Numpy¶. https://numpy.org/. [Online; accessed 18-April-2020].

[4] pandas. https://pandas.pydata.org/. [Online; accessed 18-April-2020].

[5] Quarks place. http://www.lagers.org.uk/g4p/. [Online; accessed 18-April-2020].

[6] Spotify web api. https://developer.spotify.com/documentation/web-api/. [Online; accessed 18-April-2020].

[7] Sqlite jdbc. https://bitbucket.org/xerial/sqlite-jdbc/src/default/. [Online; accessed 18-April-2020].

[8] Nicolas Carbone. Spotify past decades songs attributes. https://www.kaggle.com/cnic92/spotify-past-decades-songs-50s10s#2010.csv, Jan 2020. [Online; accessed 18-April-2020].

[9] Jonathan Feinberg. peasycam v302. http://mrfeinberg.com/peasycam/. [Online; accessed 18-April-2020].

[10] Processing Foundation. Reference. processing was designed to be a flexible software sketchbook. https://processing.org/reference/, journal=Back to the Processing cover. [Online; accessed 18-April-2020].

[11] Andreas Schlegel. controlp5. http://www.sojamo.de/libraries/controlP5/. [Online; accessed 18-April-2020].