

Basic Info

Popular Viral Song Explorer for DJs

Team members:

Kasidy Fernandes	u0690696
Kyle Woods	u0890696

Project Reposisty: <https://github.com/kasidyffernandes/CS6630Project>

Background and Motivation

As a DJ it is important to stay in touch with popular music and ongoing trends in order to entertain and engage your audience. While there are numerous resources available for DJ's to track such trends, they often require a paid subscription (like [BPM Supreme](#)), or are for very specific, niche-genres (like [Beatport - charts](#)). Additionally, these resources consist of fairly static lists of songs and do not offer much additional information or interactivity for further exploration of the songs that are popular.

Available datasets include many interesting variables for each song including fundamental traits like the tempo, key, duration and loudness as well as perceived traits such as energy, danceability, valence, and speechiness of a song. With the number of variables available there is an opportunity to encode these values in much more interesting and possibly enlightening visualizations other than the lists/tables that are readily available.

A major indicator of song popularity can also be found on social media platforms, especially TikTok, and by how viral songs are. Data surrounding trends with music on the TikTok platform have been the topic of some research ([Vox - We tracked what happens after TikTok songs go viral](#)) which is an influence on this project. Users of the platform know how often the same songs will get used and how quickly new trends will spread. By injecting a viral song into a DJ set, you can capitalize on the audience's associations with the song for some fun audience engagement to break up your DJ set.

Planning out a DJ set in order to create a cohesive and interesting set of songs can take lots of preparation and planning to arrange tracks together for smooth transitions while two songs are overlapping. The major factors for what songs will mesh well are the tempo or beats per minute (BPM) and the key that songs are in. It's generally a best practice to mix songs of similar BPMs and with keys that are harmonically similar to avoid clashing sounds. Sometimes, putting together a mix can feel like putting together a jig-saw puzzle and you are hunting for a certain piece to fit in order to work your way through a list of songs. A motivating factor for this project is to provide a tool that would help indicate some popular, trending songs that any DJ could slot into a set to grab the audience's attention and break up a mix of music.

Project Objectives

For this project we are taking an exploratory approach to analyze common trends in popular music. The datasets used for this project contain the audio characteristics (BPM, key, danceability, energy, etc) of the top charted songs from Spotify and TikTok. We aim to represent this data in a meaningful way to discover common patterns in popular songs as well as comparing how the top songs vary between the

two platforms. By representing important traits, like BPM and key, and allowing for filtering and sorting by these traits, this project could also serve as a tool to help add popular songs to DJ mixes.

Data

The data we will be working with contains the most popular songs from Spotify and TikTok. The following attributes will be used to build the visualizations that are considered attributes of the song:

- Song Information – Artist, genre, album, artist followers
- Danceability – based on a combination of elements including tempo, rhythm, and stability.
- Energy – represents a perceptual measure of intensity and activity.
- Key - Standard Pitch Class Notation.
- Loudness- overall amplitude of decibels (dB).
- Speechiness – presence of spoken words in a track.
- Acousticness – a confidence measure of whether a track is acoustic.
- Liveness – detects the presence of an audience.
- Valence – measuring the positiveness of a track.
- Tempo – overall estimate of a track in beats per minute.
- Duration- length of track in milliseconds.
- Genre - the genre or genres the song belongs to

There are also values available to show the popularity of the song these include:

- Streams - The total number of streams of a song
- Highest Charting Position - the highest rank achieved on the [Spotify Charts](#) (1-200)
- Number of Weeks Charted - the number of times the song appeared on the Spotify Charts
- Week of Highest Charting - week date range when the song was the most popular

We will be using Kaggle to acquire most of our datasets as well as Spotify API to provide any additional data types we may need.

- [Spotify Top 200 Charts \(2020-2021\) | Kaggle](#)
- [Spotify Top Chart Songs 2022](#)
- [TikTok Trending Tracks](#)
- [TikTok Trending Songs 2019-2022](#)

Data Processing

For this project we will need to supplement the datasets with additional attributes. Adding the release date and genres for the songs will help with our visualizations. We are also looking to get more time specific data to provide a more detailed visualization. Most of these additions can be found on Spotify and Kaggle.

Since multiple datasets will be used we will need to aggregate the data which will be done using the Python/Pandas libraries.

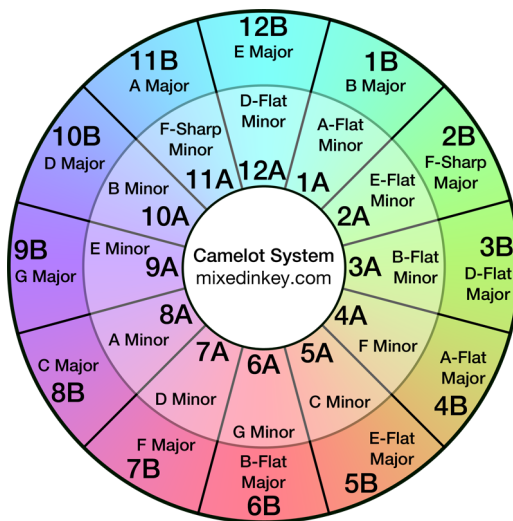
- Looking up date of song release
- Looking up genre from artist name
- Looking up what weeks the songs were on the Spotify Charts list

There will be some care that needs to be taken when handling strings for the artist's names or song names as these will often include special characters that can get turned into odd characters when imported or passed around.

Visualization Design

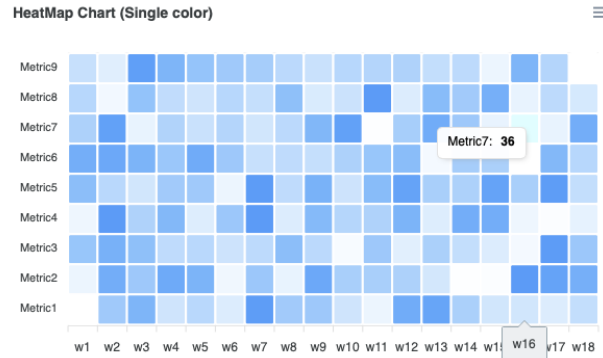
“Main Graph” - We will have a scatterplot that will encode songs by their number of streams (x-axis) and via selectable attribute (y-axis) from the list in the Data section. The data points will be color coded based on if they come from the Spotify dataset, the TikTok dataset or appear in both. This plot will have brushing available to filter the other visualizations.

“Camelot wheel” - This wheel (left) is a method used by DJs to categorize songs by key and mode and is often used in DJ-ing software like Serato or Rekordbox. We plan on implementing a circular bar chart (right) to represent the frequency of each key of popular songs. The number of songs in each key will be encoded by the height of the bar in each section. We will use the same colors for the keys since those are standardized for Camelot notation and are used in DJ software.



Based on the brushed selection from the main graph, or defaulting to the entire dataset, the last two visual elements will change to show the top 10 songs and their attributes.

“Heatmap” - we will encode the various attributes of the top ten songs in a heatmap similar to the one shown below:

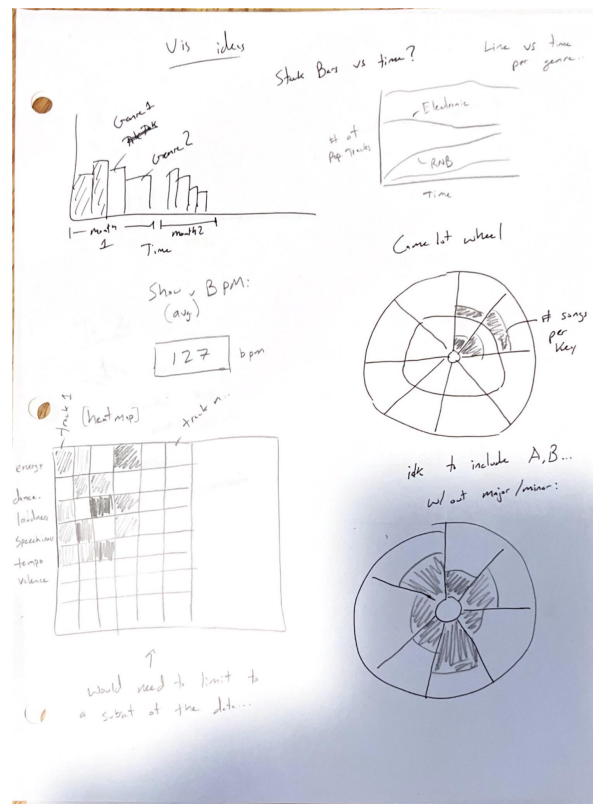


“Top 10 List” - We will have a table showing the top 10 songs in a selection with the album artwork, artist name and song name.

We arrived at these main visualizations after doing some brainstorming on what charts would be possible. We were trying to figure out if we can arrange one chronologically by when the song was most popular (the weeks it was on the spotify list). This was a struggle to conceptualize how the time a song was popular would be easily visualized with other attributes especially considering a large size data set.

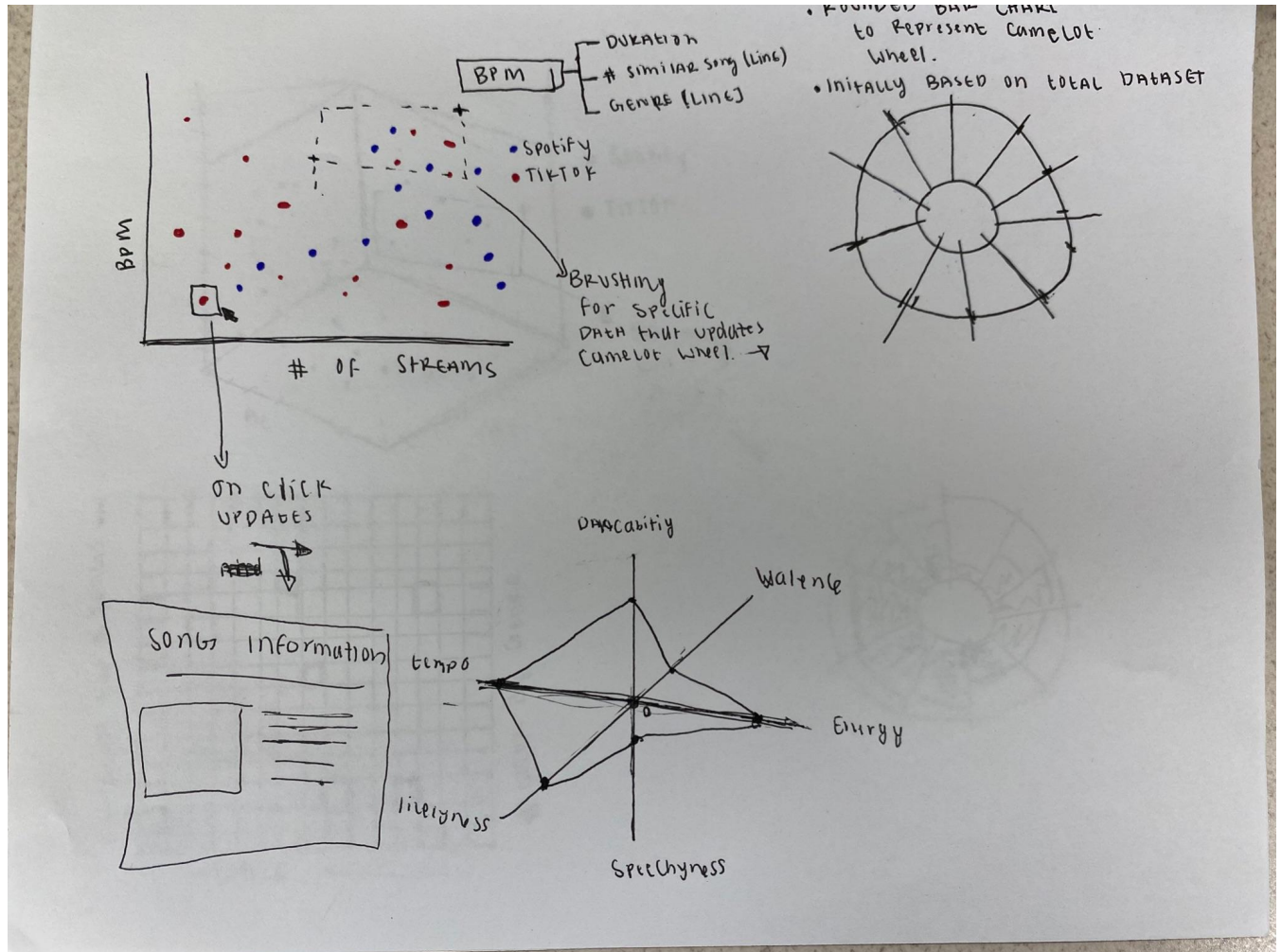
Brainstorm

Trying to show counts of different genres over time via bars or lines. Showing the average BPM overall. Heatmap and Camelot Ideas roughed out:



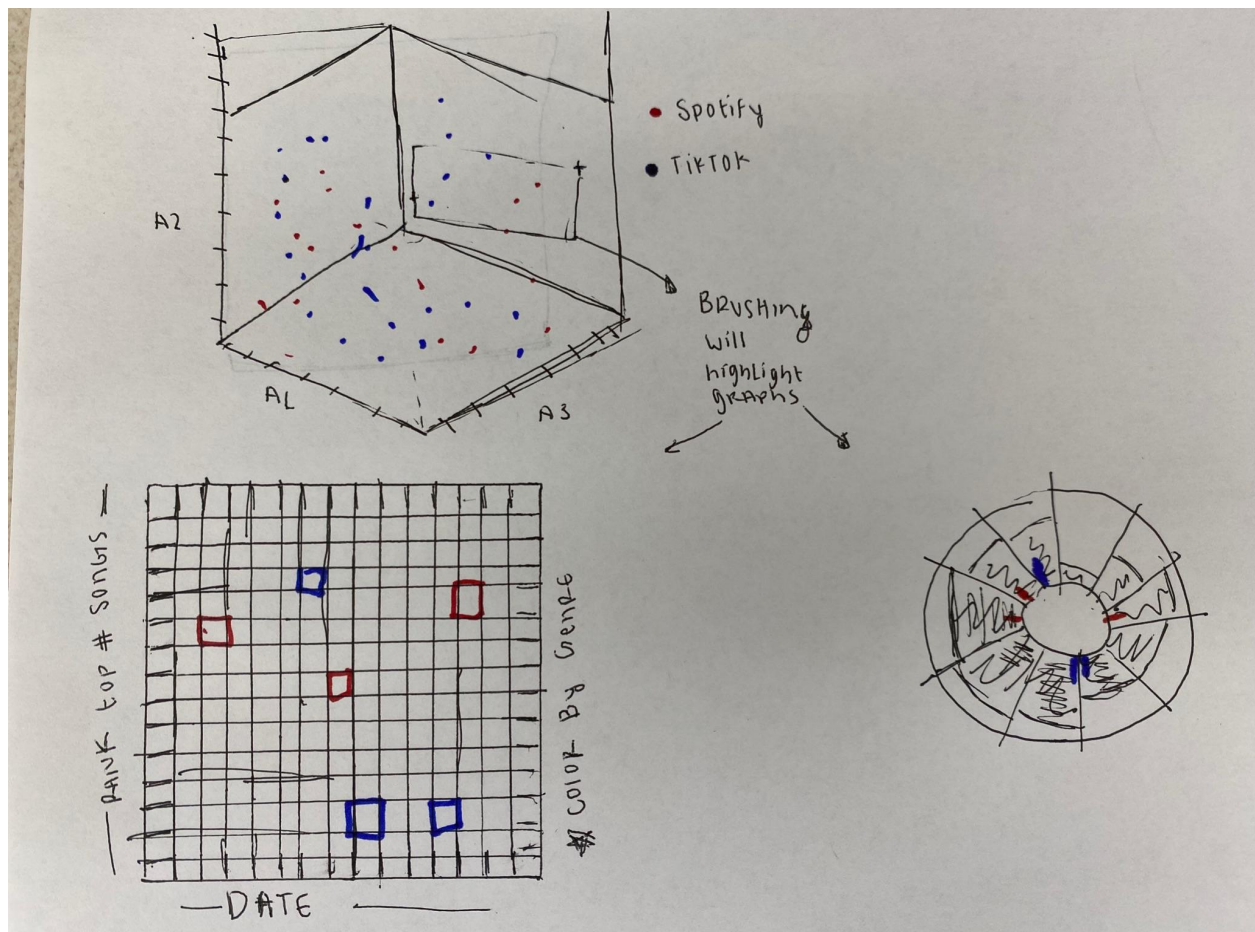
Prototype 1

The scatter plot compares two different attributes (EX: Tempo, # of Streams), these attributes can be changed via a dropdown menu. The color of the encodes whether the data is from tiktok or spotify. Selecting an individual scatterplot node will execute a tooltip with all of the song/artist information and also trigger a spatial graph that represents the nodes' audio characteristics.

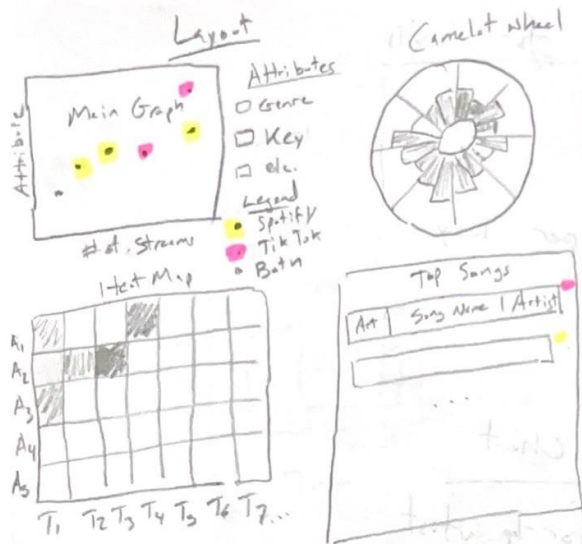


Prototype 2

Contains a 3D scatter plot that compares three different attributes. Similarly to prototype 1 the marks encode which dataset the variables are from. There is a heat map that contains the highest ranked songs within a timeframe the cells will be color coded to represent the genre type. We want to implement a brush that will update the surrounding graphs with the contained data variables.



Final Design with Notes



Main Graph - Shows selected attribute vs # of streams

→ has brushing to limit the selection and update the other charts

Camelot Wheel

Shows # of songs in each key for selection, defaults to all songs in data set

Heat Map / Top Songs

- Top 10 songs for selection by # of streams
- have album art, song name
- artist name and # of streams
- shows if from spotify or tik tok

Heat map displays attribute values for the top songs in selection

- Optional Features
- Play song when datapoint is clicked
 - Show more info / tooltip on song when it is clicked:

Top Songs

	Art	Track	Artist
1			
2			

	Art	Track	Artist
3			

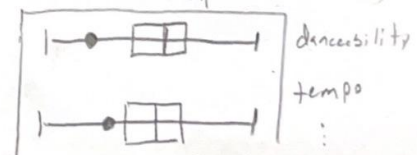
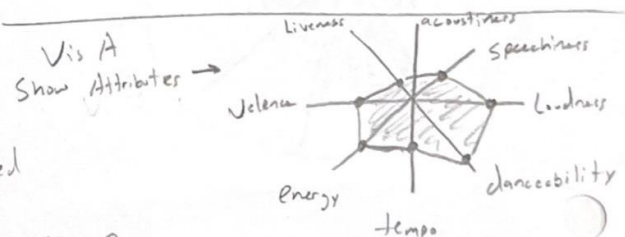
Chart dates, Release Date, Credits

Vis A

Vis B

← Selected Track

Vis B compared to Selection stats



Must-have features

This project will include the following features

- An interactive scatterplot with selectable attributes to plot and brushing that will update the remaining visual elements
- A visual representation of how many songs are in each key in the selection shown on a Camelot wheel
- Top 10 List of songs in the selection based on number of streams

Optional Features

One feature that we hope to implement into our visualization is connecting to Spotify API to display additional information about the featured song. The information could include album artwork, song credits, artist information. We would also like to implement a function that allows the user to play the song from Spotify when a data point is clicked.

Another possible feature could be to expand on a song in the Top 10 list in order to show a detailed view with some additional visualizations comparing its attributes to the others in the list.

Project Schedule

- **Week 9**
 - Finish Proposal
 - Plan the project schedule, decide on layout/features etc.
- **Week 10**
 - Data acquisition/wrangling
 - Rough outline in code with data
- **Week 11**
 - Create a prototype
 - Implement working views
- **Week 12**
 - Work on interaction aspects of the project
 - Milestone due
- **Week 13**
 - Finalize Process book
 - Create working demo
- **Week 14**
 - Finish project screencast of a working demo
 - Finalize final project
- **Week 15**
 - Project due.