

1. Introduction

The relationship between social movements and music has a long history, with songs and artists often serving as a reflection of their time's political and social conditions. Examples of this can be seen in songs such as "Blowin' In the Wind" by Bob Dylan, which was related to the civil rights movement and anti-war sentiments, and "A Change is Gonna Come" by Sam Cooke, which became an anthem for the civil rights movement. Similarly, the Italian folk song "Ciao Bella" has been used in leftist movements in Europe. More recently, the Black Lives Matter movement, which began in 2013 in response to the acquittal of Trayvon Martin's murderer, has been closely tied to music, particularly with songs such as "Alright" by Kendrick Lamar becoming an anthem for the movement due to its hopeful tone.

The digital age has brought about many changes to our daily practices, including how we listen to music. With the rise of streaming platforms like Spotify, more and more people are turning to these platforms to listen to their favorite songs. As a result, these platforms are able to collect a wealth of data on users' listening habits and preferences. Spotify, in particular, has developed a recommendation engine that uses various audio features to score and recommend songs to users. These features include liveness, loudness, speechiness, and danceability, but the exact definitions and scoring criteria for these features are not publicly available.

In this project Top Charts of the Spotify Weekly in the United States between 2016 and 2023 is collected from Chartmetric Data Analytics platform, then using the Spotify API the collected data is merged with audio features. This data will then be used to study the audio features of songs in the top charts during the Black Lives Matter movement period. This project aims to provide a comprehensive dataset that can be used to further research on the relationship between social movements and music.

2. Spotify API

The Spotify Web API is a platform that allows developers to access the Spotify music library and data. This includes the ability to search for songs, albums, and artists, as well as access information

about individual tracks and playlists. The Spotify Web API also provides access to audio features of songs, such as tempo, key, and energy. The audio feature analysis endpoint returns a JSON object containing various audio features of the song. This feature is extremely useful for music-related projects as it provides access to a vast amount of data about songs and allows for the creation of personalized playlists, music recommendations and so on.

3. Chartmetric Data Analytics Platform

Chartmetric is a data analytics platform that is tailored to the music industry. It offers a wide range of data and insights that allow users to track and analyze the performance of songs, albums, and artists on various platforms. With real-time data, historical data, and predictive analytics, Chartmetric provides a comprehensive view of the music industry and can be used to gain a competitive edge. It is a valuable tool for record labels, music managers, and independent artists to track and analyze the performance of their music on streaming platforms and social media.

4. Script

The script named `spotifyAPI.py` is for getting audio feature analysis for songs. It uses the Spotify Web API to get access to the Spotify data. The script first gets the token from Spotify by providing the `client_id` and `client_secret`, which is used to authenticate all the subsequent requests. The script also extracts data from the 'Top Charts Data' directory which contains the weekly top charts CSV files and collects all the data in a pandas dataframe called `analysis_df`. Then it uses the token to search for a song by name, to get its ID, which is used to get the audio features of the song. The audio features of all songs are collected and stored in a pandas dataframe. Also, it reads and uses a csv file "audiofeatures.csv" to store all the features of the songs. It then calls the `finalDfCreator` function to get the audio features of the songs and merge it with the extracted pandas dataframe. Further details of methods used in the script are explained.

1. `getToken()`: This method is used to get the token from Spotify by providing the `client_id` and `client_secret`. It encodes the `client_id` and `client_secret` and sends a POST request to Spotify's token API to get the token.

2. `getAuthHeader(token)`: This method is used to create the headers for the subsequent requests. It takes the token as an argument and returns the headers with the token.
3. `searchForSong(token, song_name)`: This method is used to search for a song by name. It takes the token and the name of the song as an argument and sends a GET request to Spotify's search API with the song name and type as 'track' and limit as 1. It then returns the ID of the first song in the search results.
4. `getAudioFeatures(token,song_name)`: This method is used to get the audio features of a song. It takes the token and the name of the song as an argument, calls the `searchForSong` method to get the song's ID and then sends a GET request to Spotify's audio features API with the song ID. It then returns the audio feature analysis of the song as a JSON object.
5. `chartmetricToAnalysis()`: This method is used to extract data from the 'Top Charts Data' directory and create a pandas dataframe with that data. It creates an empty dataframe with the given columns, then reads the csv files in the directory and adds the data to the dataframe, with the date of the file added as the first column. It then returns the dataframe and a list of songs by removing the duplicates.
6. `finalDfCreator(token, songs, analysis_df)`: This method is used to get the audio feature analysis of the songs and merge it with the extracted dataframe. It takes the token, a list of songs, and the extracted dataframe as arguments. It first reads an existing 'audiofeatures.csv' to check if the audio features of the songs have already been collected and stored. If it does not exist, it creates an empty dictionary to store the audio features. It then iterates through the list of songs and gets the audio features of each song using the `getAudioFeatures` method, and stores it in the dictionary. Finally, it creates a new dataframe from the dictionary, merges it with the extracted dataframe, and returns the final dataframe.