Ava Yip
DS 210

Music Trends: How Similar are Songs that Go Viral on TikTok?

Objective

The focus of this project is to determine if there are certain features in songs that make them more likely to go viral on TikTok—in other words, whether or not TikTok-viral songs are more similar to each other than generally popular songs are. This is done by evaluating various attributes of viral songs on TikTok and songs from Spotify Top Charts, namely danceability, energy, speechiness (amount of spoken words in a track), valence (musical positivity), and tempo. Average distances in terms of these attributes are calculated between each node from a stratified random sample of TikTok-viral songs from the years 2019-2022, and the same is done for a sample of songs from Spotify Top Charts 2000-2019, before TikTok became popular and began influencing the charts. These averages are then compared to determine if there is a significant difference that suggests that there is a certain "formula" of features that make songs more likely to go viral on TikTok.

Code

The song module implements a Song struct with the traits title, danceability, energy, speechiness, valence, and tempo.

The function create_attribute_nodes() takes in a vector of Songs, creates a vector of the five attribute values in each song, and returns a vector of all of these vectors. These are the attribute nodes that will be used when calculating distances.

The function select_random_sample() takes in a vector of attribute nodes and a number of samples and selects a random sample of the given size from the given vector.

The function distance() calculates the distance between two nodes.

The function average_distance() calculates the distance between all pairs of a given list of nodes and returns the average of these distances.

The function max_distance() returns the largest distance that exists between two nodes of a given list of nodes. It also returns the coordinates of these two nodes.

The function get_song_title() takes in one node and a vector of Songs, matches the attributes of the node to the correct song in the list, and returns the corresponding song title. If no match is found, the function returns "NA".

In the main module, there are two file reader functions: one personalized to the four TikTok datasets, and one personalized to the Spotify dataset. They each read in tsv files and collect the wanted attributes. The attributes are normalized so that all of the values are between 0 and 1 in order to ensure no single attribute will have a significantly greater impact on the calculated distances. The information is parsed into and returned as a vector of Songs.

In the main function, for the TikTok songs from each year from 2019-2022, a random sample of 50 songs is chosen, and the average distance between these songs is calculated. These four averages are then averaged, finding the overall average distance of a stratified random sample of 200 TikTok songs. Then, a random sample of 200 Spotify Top Charts songs is collected, the average distance between these songs is calculated, and the two averages are compared to determine if there is a significant difference.

To further visualize this data, the largest existing distance between two nodes is also calculated for each dataset. This is calculated using the vectors containing all of the data (250+ nodes for each TikTok year and 1000+ nodes for Spotify) rather than the random sample in order to find the true maximum distance.

Results

After running the code multiple times, I found that a majority of the random samples produced a TikTok song average distance that was smaller than the Spotify song average distance, demonstrating that TikTok songs are, on average, more similar than Spotify Top Charts songs. However, there are some random samples where the Spotify average distances are smaller than the TikTok average distances. The distances are also very small, typically ranging from 0.01-0.09. Additionally, when comparing the maximum distances of the TikTok datasets from each year to the Spotify dataset, the Spotify maximum distance lies in between the TikTok maximum distances. Therefore, I feel that I cannot conclude with certainty that there is a certain "formula" involving these attributes that make a song more likely to go viral on TikTok. TikTok songs do seem to be more similar to each other than generally popular songs are, but the difference is very small.

Below is a sample output of my code:

2019 TIKTOK

avg distance between random sample of 50 songs: 0.43167260017222137

max distance between two songs: 1.2131809788222037

most different songs: u U U U u U did i mfken stutter?; Kill the Director

2020 TIKTOK

avg distance between random sample of 50 songs: 0.4568031158018683

max distance between two songs: 1.3895585731360878

most different songs: Knock at the Door; Lucky

2021 TIKTOK

avg distance between random sample of 50 songs: 0.40837637424352946

max distance between two songs: 1.0931482640081354

most different songs: Caroline; Love Tonight - David Guetta Remix Edit

2022 TIKTOK

avg distance between random sample of 50 songs: 0.4161464724706799

max distance between two songs: 1.1120985341236629

most different songs: Down Under (feat. Colin Hay); Build a Bitch

2000-2019 SPOTIFY

avg distance between random sample of 200 songs: 0.4372800831147022

max distance between two songs: 1.3106120162061692

most different songs: My Immortal; Beat Again - Radio Edit

AVERAGE DISTANCE COMPARISON

tiktok avg distance: 0.4282496406720747
spotify avg distance: 0.4372800831147022
TikTok viral songs are, on average, more similar than Spotify Top Charts songs.