

## Changes in Music Sentimental Analysis and Explicitness Over Time

### Project Overview

My parents always tell me and my siblings that “our music is more explicit and negative” than theirs. My goal for this project, was to determine whether or not that was the case by using NLTK Sentiment Analysis and the MusixMatch Lyrics API to analyze songs. Using these tools, I created a program that could analyze any songs where the lyrics were available through the MusixMatch API and then compare them with a different set of songs, typically from a different year.

### Implementation

The data for the songs is stored in a dictionary with the key being the song name and the value being the artist name. I did this because it is less likely to have a repeat of the song name than the artist. (For the data tested, neither were repeated.) Both the artist name and the song name were needed to plug into the url to get the url for the song with the lyrics. The lyrics are then retrieved as a json response and automatically the unnecessary data is removed from the lyrics. Once the lyrics are compiled, the sentimental analysis is completed and then saved to a json file.

I chose to store this data in a json file because it allows for key-value pairs, which include whether or not the song is explicit, the positive and negativity scores, and more. These are all linked to the songs so that users can see the individual songs' score which could make outliers in the data more obvious. This key-value pair data type is why I chose to use a dictionary, mostly over a list because it makes it easy to request specific sets of information.

Finally, the data sentimental analysis and explicitness are compared between the first dictionary and the second. Here, the scores are stored as floats while the explicitness is an int because it is a 0 or 1. I chose to store them as ints for two reasons, 1) the API passed them as a 0 or 1 response and 2) because it is the simplest solution for adding up the amount of explicit songs when doing the analysis.

### Results

While the results output data for negativity, neutrality, positivity, a compound result, and explicitness, my analysis is mostly focused on negativity, positivity, and explicitness. The compound score is useful for looking at a single metric which at this exploratory stage is not very useful.

The negativity score of all the new songs is 0.366 more than the older songs which means that my parents' statement, “that new songs are more negative than older

songs," is correct. Connected to this negativity, two of the five new songs were also explicit compared to zero of the older songs. While the new songs are more negative according to NLTK's analysis, the new songs also contain more positivity (0.493 old, 0.625 new) than the older songs. Some of this could be due to the nuances of music where the context and how the words are said impacts whether or not a sentence over all is positive or negative.

Looking at individual songs yields interesting result because so much is classified as neutral. The results also show that newer music is more explicit in general than older music.

## Reflection

I enjoyed completing this project overall; however, given more time, there are a few changes I would make. First, I would likely combine the two dictionaries into one; from there, the keys would be the years and within there the artist (not the songs) would be the keys. And then the songs would be a list within the key of artist. This would allow multiple songs with the same name and allow for multiple songs by the same artist. I would also like to use all the song lyrics instead of just the first 30% (which is a limitation MusixMatch has on their free plan). I also think that by comparing more songs, maybe the top 50 instead of 5, I would have a better view into the analysis (however MusixMatch has an undocumented limit on the amount of requests can be made on the free plan). Finally, with this new system, I believe I could use fewer lines of code even with the increased complexity.

For demonstration and testing purposes, I chose to use the top 5 songs from the Billboard Top 5 rankings for each year spaced out by 30 years. This may not be a perfect reflection of all the songs and also may not be totally indicative of what my parents are saying since neither I nor my parents' taste may match the preferences top music.

I think the project was appropriately scoped as it used some of the more complex things we have learned and a lot of data could be garnered from what I created. I also had a good plan for testing as I was able to cover scenarios not seen in the data such as the API not responding with lyrics. From what I learned, I believe future projects will be able to use less code by using multiple functions and it would prevent the duplicated lists that I used. Before starting, I wish I knew to use nested dictionaries instead of using multiple to be able to display the years associated with the songs and data. I also think it would have been more effective to use more functions but if I restructured the code, as outlined earlier, more functions would not be necessary.