CSC111 Project Proposal: Music Genre Proximity Tree

David Wu, Kevin Hu

Tuesday, March 16, 2021

Problem Description and Research Question

To many of us, music is a large part of our lives, helping us get through the day. According to the International Federation of the Phonographic Industry (IFPI), the average person spent 18.4 hours a week listening to music in 2021. With the sheer number of music available on modern streaming platforms such as Spotify or Apple Music, it can be extremely difficult and time consuming for the casual listener to find new music that they enjoy, especially since listeners themselves may not know what they like. A listener may know that they typically enjoy Rock music but "Rock" is a broad term that encompasses many distinct subgenres, perhaps they are into the raw, distorted sound of Noise Rock or maybe they prefer the more technical, complicated time signatures of Math Rock. This is not to mention that different albums within a subgenre could also be further categorized by different descriptors such as warm, aggressive or depressing. The casual listener is simply unaware of these distinctions when trying to find something they like on the Rock section of Spotify. We believe that good music recommendations should be fast and easy, **our goal is to provide users with the best possible song recommendations, given the current preferences of the user in either the form of a preferred genre or a favourite album,** letting users spend less time looking for music and more time listening. We would also provide an interactive visualization that shows how our top recommendations fit within the larger tree of music genres and albums, allowing users to explore a specific musical pathway on their own and potentially discover new songs and albums they can enjoy, without having to search through individual songs on music streaming sites.

Computational Plan

Our project will use a tree to represent the albums in our dataset in a hierarchical manner. Starting from the root, the tree will branch into broad genres (pop, rock, hip-hop, etc.) with each genre then branching into more specific subgenres (ex. the genre 'Rock' may branch into subgenres like 'Noise Rock', 'Pop Rock' or 'Psychedelic Rock'). Subgenres may also have their own subgenres (ex. 'Hip-Hop' may branch into 'Trap' which branches into 'Drill' which branches into 'UK Drill'). This tree will be visually displayed on a graphical interface and users will be able to interact and explore the tree on their own by clicking deeper into the tree. To prevent the tree from becoming too large in the visualization, we will only display a tree with a maximum depth of 2 on the interface each time, whenever a user clicks on a genre, it will update the interface to display a new tree with that genre being the new root of the tree.

In terms of computations, we'll be filtering our data set by genres and by album descriptors. We'll first create a simple tree that branches off by genre(i.e. rock can branch off into classic rock, punk rock, etc), which the user can explore on their own, or after they get a recommendation. This tree can be created by sorting by keywords(i.e. finding the keyword 'rock' for one subtree) and adding subtrees by these keywords. The other tree we will be creating is our recommendation tree, which will have two different settings. The user can decide to input a genre they like or an album they like. If the user first inputs an album, the descriptors of that album will be taken and compared to every other album. Albums with the most matches would then be added as subtrees of the recommendation tree. Then, with each recommended album, the descriptors of those albums will be taken and compared, to then create another line of recommendation. To make sure the same albums aren't recommended over and over, albums already existing in the upper levels of the tree will not be in the comparison at lower levels of the tree. If the user first inputs a genre, the top albums of that genre will first be displayed(a top album being the most popular album, measured by the number of interactions with that album in our data set). Within those recommended albums, the descriptors of those albums will be taken, and the recommendation algorithm will then switch to matching the descriptors with other albums not already in the tree. To display our program, we will first use the tkinter library to display the genre tree, which the user can interact with to explore the different genres. We will then use the plotly library to display

the recommendation trees on a separate window. To reduce the potential size of one tree, the recommendation tree will be restricted to a certain depth. If the user wants to explore a certain subtree more, they can click that subtree, which will then generate a new tree with the selected album as the new root. Previous recommendation trees will also be saved, so the user can have to option to go back if needed.

References

https://www.ifpi.org/wp-content/uploads/2021/10/IFPI-Engaging-with-Music-report.pdf TODO