Spotify Artist Network Project Summary

Project Overview

The Spotify Artist Network project is a Python-based interactive program that leverages Spotify playlist data to build a graph representing artist connections. Users can explore artist relationships, popular songs, and playlists where artists co-occur. The goal of this project is for users to gain insights into an artist's popularity and receive recommendations for related artists who share commonalities with their favorite artists.

My graph is designed to store relationships between artists, where each artist is represented as a node in the adjacency list. A node is a tuple containing the artist's ID, name, and Spotify URL. Edges between nodes represent their co-occurrence in playlists, with weights indicating the number of playlists where the two artists appear together. Artists that frequently co-occur in playlists have stronger connections (higher weights).

To optimize program performance and reduce processing time during execution, I focused on storing only the artist relationships in the adjacency list and excluded detailed playlist and track information from the graph. Instead, playlists and tracks are stored separately and accessed directly from the file when needed.

Interactions and Findings

1. **Discover Popular Artists:** Users can find the most connected artists based on co-occurrence in playlists. Artists such as The Weeknd and Post Malone frequently co-occur in playlists, reflecting their prominence and collaborations.
2. **Identify Strong Relationships:** The program identifies the pair of artists that co-occur the most in playlists. This indicates shared audiences and frequent collaborations. In my dataset, there are multiple pairs that share the highest co-occurrence count of 29.
3. **Explore Artist Information:** Users can explore an artist's top songs and associated albums. This provides users with more detailed information about the artist they are interested in.
4. **Get Related Artists Recommendation:** Recommendations of related artists based on co-occurrence are provided.
5. **Search Common Playlists:** Users can explore playlists containing multiple selected artists and obtain a Spotify link to these playlists for further exploration. By exploring the graph, users can observe patterns in artist collaborations.

Challenges

1. After reviewing the Spotify Web API, I noticed that the "Get Available Genre Seeds/categories" endpoint is deprecated and returns an empty array, making the data inaccessible. As a result, I replaced the interaction mode in my original plan (genre-based recommendation) with the ability to get a playlist that contains two artists inputted by the user.
2. Since calls like "Get Category's Playlists" are also deprecated, and the "Search for Playlist" endpoint only supports keyword searches, I had to search for playlists using the keyword "2024", which is not a real year filter. This means inevitably many playlists have similar titles like **"Top Hits 2024"**. In terms of data interpretation, it is difficult to tell if there is a common pattern/style between two artists who co-exist in these playlists.
3. I investigated platforms like Kaggle and other resources to find playlists organized by genre but was unsuccessful. As a result, I continued using the current dataset for this project while planning to explore genre-specific playlists in the future.

Conclusion

The Spotify Artist Network project demonstrates how graph-based analysis can be used to uncover meaningful patterns in music data. It provides an interactive way for users to explore connections, discover new music, and understand artist relationships. With potential extensions, this project could further enhance music recommendation systems or explore genre-based analysis.