

Create a temporal network of genres:

In this function, a directed graph (`nx.DiGraph()`) is created and populated using the NetworkX library. For each artist in the provided artist list, their genres and associated releases are obtained. The releases are sorted chronologically by release date. Then, the releases are iterated in order, and the genres of the current artist and the next artist in the list are obtained. For each genre of the current artist, a node is added to the graph. Connections are established between the genres of the current artist and the genres of the next artist in the list. If the connection already exists, the weight of the edge is increased by 1. If it doesn't exist, a new edge is added with an initial weight of 1. Visualize the temporal network of genres:

The resulting graph is a directed graph where the nodes represent music genres and the edges represent transitions between genres over time. The graph nodes are unique for each genre, and the edges represent transitions between genres. The weight of an edge indicates how many times that transition between genres has been observed in the provided list of artists.

In the provided code, the only attribute used is the weight of the edges, which represents the frequency of transition between genres.

The list of artists is used in the code to analyze the evolution of music genres over time. Each artist in the list represents an individual entity contributing to the temporal genre network. By having a list of artists, the code gathers information about the genres associated with each artist and their releases (albums). By sorting the releases by release date, the sequence of genre transitions over time can be analyzed.

The idea behind this is to track how music genres have evolved over time by observing genre transitions between artists and their releases. With a list of artists, it is possible to analyze how genres have influenced and changed over time, identify trends, observe the emergence of new genres, and explore the influence of past genres on current ones.

REPORT

1. Define the objectives of your study. What question or objectives do you aim to address?

The objectives of the study are to analyze the evolution of music genres over time and to identify genre trends, the emergence of new genres, and the influence of past genres on current ones. The study aims to answer questions such as how genres have evolved, which genres have gained popularity or declined over time, and how genres have influenced each other in the music industry.

2. Describe the process of data acquisition and the obtained data.

The data acquisition process involves using the Spotify API to gather information about artists' genres and their release dates. The Spotify API provides endpoints that allow us to search for artists, retrieve their genre information, and obtain details about their releases.

To obtain the data, the following steps were followed:

1. A list of artist names was provided as input.
2. For each artist name, a search was performed using `search_artist` function to find the corresponding artist ID.
3. The artist ID was then used to retrieve the genres associated with the artist.
4. Additionally, the releases (albums) of the artist were obtained, including their release dates.
5. The releases were sorted chronologically based on their release dates.

The data obtained includes information about the genres of each artist and the release dates of their albums. This data is essential for analyzing the evolution of music genres over time and constructing the temporal network of genres.

(b) Which API endpoints did you use to acquire the data?

To acquire the data, the following Spotify API endpoints were used:

· Search for an artist: This endpoint allows searching for an artist based on a given query. The endpoint used is:

It returns a list of artists that match the search query, including their artist ID.

· Get artist information: This endpoint retrieves detailed information about a specific artist using their artist ID. The endpoint used is:

It returns information such as the artist's name, genres, and other relevant details.

· Get artist's releases: This endpoint retrieves the releases (albums) of a specific artist. The endpoint used is:

It returns a list of albums associated with the artist, including their release dates and other album details.

Note: Those functions were taken from the internet and modified according to our needs, just few modifications to acquire what we wanted to, but the structure is not homemade.

(c) Describe the obtained data. For graphs, describe the type of graph used, what the nodes and edges represent, and their attributes. For tabular data, describe the collected entities and their attributes.

The obtained data consists of a temporal network of genres represented by a directed graph where the nodes represent music genres and edges represent transitions between genres over time.

The edges in the graph have a single attribute called "weight". The weight represents the frequency or count of transitions between two genres. It indicates how many times a transition from one genre to another has occurred in the analyzed data.

No tabular data.

(d) Indicate the volume of collected data. For graphs, provide the size and order.

The final graph order and size:

```
Order network: 37
Size network: 37
```

e) Detail the format to store the data.

Graph of Networkx library.

3. Describe de data preprocessing tasks

No data preprocessing tasks at all except extracting what we needed from the API.

4. Describe the data analysis tasks.

(a) What network analysis algorithms did you use?

The code does not explicitly mention the use of network analysis algorithms. However, once the temporal network of genres is constructed, various network analysis algorithms can be applied to gain insights into the evolution of music genres.

5. Describe the visualization tasks.

(a) What tools did you use to generate the visualizations?

The last result we decided to keep is just the graph visualization with a spring layout used from the networkx library, we tried gephi visualization based on metrics but as the graph is relatively simple we did not achieved better results.

Temporal Genre Network

