Instruction and Data Description

The description of data:

1. influence_data.csv
2. full_music_data.csv
3. data_by_artist.csv
4. data_by_year.csv

Data Descriptions

1. influence_data.csv (Data is encoded in utf-8 to allow for handling of special characters):
- influencer_id: A unique identification number given to the person listed as influencer. (string of digits)
- influencer_name: The name of the influencing artist as given by the follower or industry experts. (string)
- influencer_main_genre: The genre that best describes the bulk of the music produced by the influencing artist. (if available) (string)
- influencer_active_start: The decade that the influencing artist began their music career. (integer)
- follower_id: A unique identification number given to the artist listed as follower. (string of digits)
- follower_name: The name of the artist following an influencing artist. (string)
- follower_main_genre: The genre that best describes the bulk of the music produced by the following artist. (if available) (string)
- follower_active_start: The decade that the following artist began their music career. (integer)

2. full_music_data.csv 3. data_by_artist.csv 4. data_by_year.csv
Spotify audio features from the “full_music_data”, “data_by_artist”, “data_by_year”:
- artist_name: The artist who performed the track. (array)
- artist_id: The same unique identification number given in the influence_data.csv file. (string of digits) Characteristics of the music:
- danceability: A measure of how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable. (float)
- energy: A measure representing a perception of intensity and activity. A value of 0.0 is least intense/energetic and 1.0 is most intense/energetic. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. (float)

- valence: A measure describing the musical positiveness conveyed by a track. A value of 0.0 is most negative and 1.0 is most positive. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). (float)
- tempo: The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. (float)
- loudness: The overall loudness of a track in decibels (dB). Values typical range between -60 and 0 db. Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). (float)
- mode: An indication of modality (major or minor), the type of scale from which its melodic content is derived, of a track. Major is represented by 1 and minor is 0.
- key: The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value for key is -1. (integer) Type of vocals: - acousticness: A confidence measure of whether the track is acoustic (without technology enhancements or electrical amplification). A value of 1.0 represents high confidence the track is acoustic. (float)
- instrumentalness: Predicts whether a track contains no vocals. “Ooh” and “aah” sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly “vocal”. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0. (float)
- liveness: Detects the presence of an audience in a track. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live. (float)
- speechiness: Detects the presence of spoken words in a track. The more exclusively speechlike the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks. (float)
- explicit: Detects explicit lyrics in a track (true (1) = yes it does; false (0) = no it does not OR unknown). (Boolean) Description:
- duration_ms: The duration of the track in milliseconds. (integer) - popularity: The popularity of the track. The value will be between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played more frequently now will have a higher popularity than songs that were played more frequently in the

past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity are derived mathematically from track popularity. (integer)
- year: The year of release of a track. (integer from 1921 to 2020)
- release_date: The calendar date of release of a track mostly in yyyy-mm-dd format, however precision of date may vary and some just given as yyyy.
- song_title (censored): The name of the track. (string) Software was run to remove any potential explicit words in the song title.
- count: The number of songs a particular artist is represented in the full_music_data.csv file. (integer)

Topic: 1. There is a shift/change in the most important **features** of popularity over time. 2. What features of music make the artist more popular?

Instructions:

For 1:

- Divide the time (from 1950 to 2010) with decades interval, conduct a feature importance analysis by XGboosting or other related machine learning algorithms for each period time
- Visualize the results

For 2:

- Will only focus on the timeframe from 1990 - 2020.
- By saying the popularity of the artist we will be looking at the social media accounts and the streaming of songs of the artist. To be more specific looking at their number of followers on social media accounts (focusing on Instagram) and streaming amount of the artists' songs ().
- Web scrape this information. Use all the artist names from the Full_mustic_Data.
- Determine which features of music the artists use the most in their music. Also available in Full_music_Data.
- For each year, provide the results of the top 20 artists for song streaming and social media followers. Also list the features of music the specific artists use.

Topic:

What does the musician(artists) social network look like?

- Firstly, how will we measure individual artists' influence? We use followers number in the data set (influence_data)
- Then, we calculate and average the influence of each song of individual artists, and incorporate the song influence with the followers number together as the general influence of artists

- For now, we do a dictionary, where each key represents an influencer, and each value represents that influencer’s followers.
- Using that, we can calculate influence by implementing a breadth-first search algorithm (BFS) and rank artists based on score.
- The direction of the point connection in this social network will be determined by the who follow who
- Visualize the subnetwork with a depth of 10

Topic:
How are the results of popularity analysis correlated with each other?

- Danceability vs Loudness vs Energy vs Valence, now we need to run a correlation / regression analysis to see the relationship of these four features
- Some related keywords performance on Google Trend (dance practice; music festival; electronic music; dj ; hit song)

Topic : Find how the **genres** influence each other over the years. For example, when pop increases in popularity, what genre loses the most popularity?
Instructions:

- There are 3 datasets with popularity as a variable: Data_by_artist, Data_by_year, Full_mustic_Data.
- If our aim is to find how genre influences each other with popularity as the measurement, we need to use multiple datasets as the dataset with the genre, influence_data does not include popularity as a variable.
- Therefore how should we measure genre popularity? We take Influence_data and take every artist in there. The dataset maps every artist to a genre.
- So we would have a list of artists, each mapped to a genre. Then we find the same artists who are mapped to a genre in data_by_artist. Data_by_artist has a popularity variable.
- So now we have a list of artists who are mapped to a genre AND has a popularity value.
- Now for time-based data, we need to go into full_music_data. Find the artists in our list and match it with the artists in full_music_data. This should have the songs they released, the year, and the popularity.
- Now to reiterate what we have so far. We have songs by artists, with release year and popularity of those songs. We also have the artists’ own popularity, their genre, and also their active start year.

- Now we have two charts. On one chart, we plot an artist's popularity as y value. The x value is time. And then we group all the artists by genre. We would have pop, rock, country, and more genres. Now the chart would look like plots of genre going through the year, measured by popularity.
- On the second chart, we do the exact same thing, but with songs. We group songs into similar genres, plot them on a chart with popularity as y and time as x. We know the genre of the songs since they are connected to artists, whose genre we know. Note that if a song has two different genres, it shows up twice as two genres. For instance it is plotted once as pop, and another as rock.
- So to reiterate, now we have two charts. One is the artists grouped by genre, their popularity through the years. The second chart are the songs grouped by genre, their popularity through the years.
- Now I want tests to be done which would result in us knowing which genre’s change affects the other genre’s change the most. For instance, if pop increases by 10% in popularity, what other genres are affected? Which genres decrease the most as a result of that?
- Models could be XGboosting, OLS, but I want to use machine learning models like tensorflow.