

## Background and Motivation:

Music streaming platforms generate vast amounts of user interaction data. This enables researchers to analyze how social structures influence music preferences. One such dataset is provided by Deezer, which is a European music streaming service. The dataset contains both friendship connections, which represent direct social links between users, and user-artist interactions, which capture the artists that users have engaged with, such as liking or listening to their music. By studying these networks, we can gain insights into how users' social environments shape their listening behaviors. Understanding these aspects of how people connect based on shared interests can be useful for applications such as music recommendation systems, targeted marketing, and community formation studies. These insights can help improve algorithms that personalize music recommendations, optimize promotional strategies for artists, and enhance user engagement on streaming platforms.

## Research Questions:

This project aims to investigate the relationship between social network structures and user music preferences. Specifically, we explore:

1. How do the structural properties of the Friendship Network and the Co-Liking Network compare when analyzed against random baseline networks?
  - *Rather than directly comparing the two networks, we construct random baseline networks with the same number of nodes and edges and evaluate how the observed networks deviate from expected random structures.*
2. How do communities in the Co-Liking Network, as identified through modularity-based clustering, shape user preferences and influence the concentration of likes on specific artists?
  - *To examine how user preferences are distributed within detected communities, we analyze the most liked artists in each community and assess whether preferences are concentrated around a few dominant artists or spread across a diverse range of musicians.*
3. How does network clustering influence the diversity of music preferences among users in the Co-Liking Network?
  - *To investigate this, we analyze the number of unique artists liked per user across different levels of clustering coefficient. This helps determine whether users in highly clustered groups exhibit narrower music preferences compared to those in loosely connected groups, or if social structure plays a role in broadening exposure to diverse music.*

4. How does a user's position in the Co-Liking Network, as measured by clustering coefficient deciles, influence the concentration of their music preferences around a small set of dominant artists?
  - *To investigate this, we analyze whether users in highly clustered social groups tend to like a smaller set of dominant artists compared to those in loosely connected networks.*
  - *By analyzing the share of total likes received by the top 5 artists per clustering coefficient decile, we quantify how artist concentration varies across different levels of network clustering, which ensures a fair comparison across equally sized user groups.*

## **Dataset Overview:**

The study utilizes the Deezer Social Network Dataset, which provides user connections and music preferences from the Deezer streaming platform.

### Data Sources

- `deezer_europe_edges.csv` (Friendship Network): Contains edges representing mutual friendships between users.
- `deezer_europe_features.json` (User-Artist Preferences): Provides a list of artists liked by each user, allowing us to construct music-based relationships.

### Dataset Characteristics

- Users are represented as nodes, while friendships and co-liking relationships form edges.
- The Friendship Network is unweighted and undirected, meaning connections indicate mutual friendships without additional numerical values.
- The Co-Liking Network is weighted, where edge weights represent the number of common artists liked by two users.

### Importance of This Dataset

- It provides a rich social graph that enables the study of both direct (friendship) and indirect (music preference) relationships.
- It allows us to analyze how social connectivity influences shared music tastes and how user preferences shape social clustering.

## Types of Networks Analyzed

This project examines two types of networks derived from the dataset:

- Friendship Network:
  - An undirected, unweighted network where edges represent mutual friendships.
  - Extracted from `deezer_europe_edges.csv`.
  - Represents direct social connections between users.
- Co-Liking Network:
  - A user-user similarity network constructed from `deezer_europe_features.json`.
  - Nodes represent users.
  - Edges exist when two users like the same artists, forming a weighted network where the edge weight corresponds to the number of common artists liked by both users.

### *Why Compare These Networks?*

- Friendship connections are explicit social ties, whereas co-liking relationships emerge from shared interests.
- The Co-Liking Network may form densely connected communities around popular artists, which can reveal patterns that are different from direct social connections.
- Comparing these networks against randomly generated baselines helps us understand whether music preferences influence network structure differently than direct friendships.
- Random baseline networks allow us to determine whether observed structural properties are meaningfully different from what would be expected by chance. This ensures that any patterns found in the Friendship or Co-Liking Network are not simply artifacts of network size or density.

### *Sampling the Friendship Network*

Since the original friendship network contained around 28k nodes, working with the full dataset for analysis would have been computationally expensive. To make the network more manageable while preserving key structural properties, we performed random sampling using the following approach:

- Randomly selecting a subset of users: We sampled 5,000 users from the original dataset.
- Extracting relevant connections: Once the subset of users was selected, we filtered the edges to only include friendships between sampled users while preserving their connectivity structure.
- Saving for further analysis: The sampled friendship network was exported as a CSV file.

This sampled friendship network maintained the general characteristics of the full network while making it feasible for detailed analysis.

### *Constructing the Co-Liking Network*

The Co-Liking Network was built to analyze user connectivity based on shared music preferences rather than direct social ties. This process involved several steps which are outlined as follows:

#### Step 1. Extracting User-Artist Preferences

To build the network, we used the `deezer_europe_features.json` file. Each user in this dataset has an associated list of artists they have liked. The JSON file was loaded and parsed using the `json` module in Python. The dataset was structured in the JSON file as a dictionary, where:

- Keys represent user IDs.
- Values contain lists of artists liked by each user.

This structure was used to construct connections between users based on shared artist preferences.

#### Step 2. Constructing the Edge List

The network was constructed by identifying users who shared music preferences. This involved iterating over all possible user pairs in the dataset and establishing an edge if they liked at least one artist in common. Edge weights were assigned based on the number of shared artists. A higher weight indicated that two users liked more of the same artists.

#### Step 3. Constructing the Co-Liking Network Graph

Once the edge list was generated, we proceeded with constructing the actual network representation. We used the `networkx` library to represent the Co-Liking Network as a weighted, undirected graph. The edge list, which contained user pairs and their corresponding weights, was loaded into `networkx` using the `from_pandas_edgelist()` function.

In this graph:

- Nodes represent users (individual listeners).
- Edges represent connections between users based on shared artist preferences.
- Edge Weights indicate the number of shared artists between two users.

This way we obtained a well-defined social network that connects users based on shared music preferences. This provided an alternative representation of social ties beyond direct friendships

and revealed new insights into how users can connect through common interests rather than explicit social connections.

#### Step 4. Exporting the Network for Analysis

To facilitate further analysis and visualization, we exported the constructed co-liking network. The network was saved as a CSV file containing the edge list, including the source user, target user, and edge weight. This exported file was then imported into Gephi, to analyze the structural patterns and visualize connectivity among users.

#### *Generating the Random Baseline Networks*

To better understand the structure of both the Friendship and Co-Liking Networks, we generated random baseline networks for comparison. These networks preserved key structural properties but had randomized connections to serve as a baseline.

##### Random Friendship Network:

To create a random version of the Friendship Network, we used the Erdős–Rényi model (ER model) with networkx:

- Preserving key properties:
  - The number of nodes and number of edges were kept identical to the observed friendship network.
- Randomizing connections:
  - Instead of real-world social patterns, edges were randomly assigned between users, which simulates a network where friendships form randomly rather than organically.

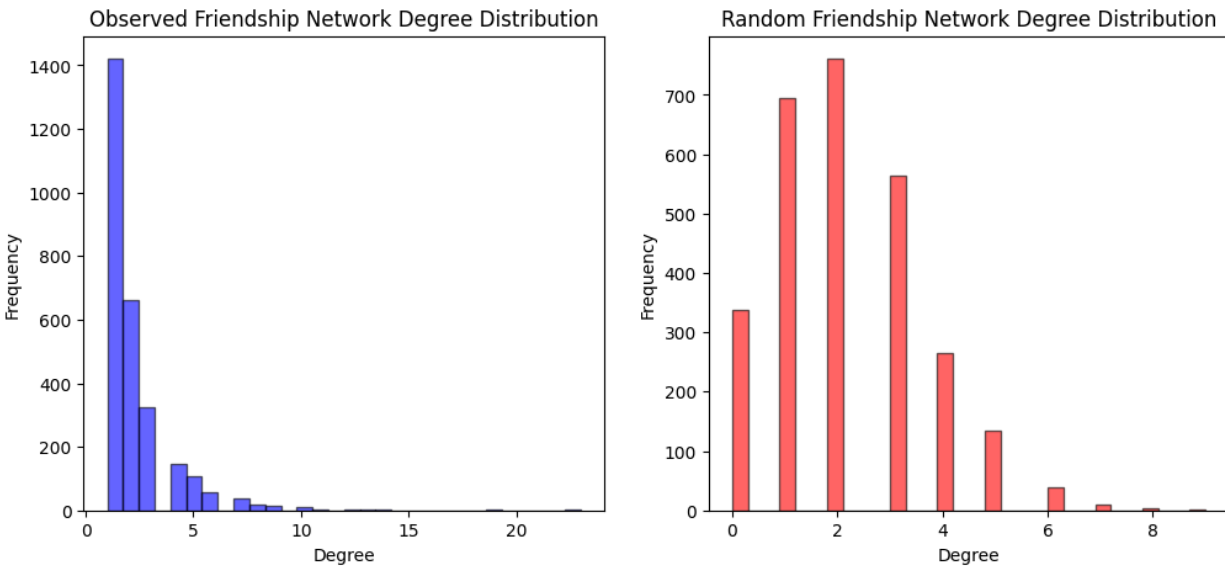
##### Random Co-Liking Network:

For the randomized version of the Co-Liking Network, we used the Erdős–Rényi model to randomly assign edges between users while preserving the same number of connections. This approach randomized user-to-user connections in a way that maintained:

- The same number of nodes (users)
- The same number of edges (co-liking relationships between users)

Unlike the real co-liking network, which is highly clustered around popular artists, the randomized co-liking network lacks this structure, which leads to a more uniform degree distribution.

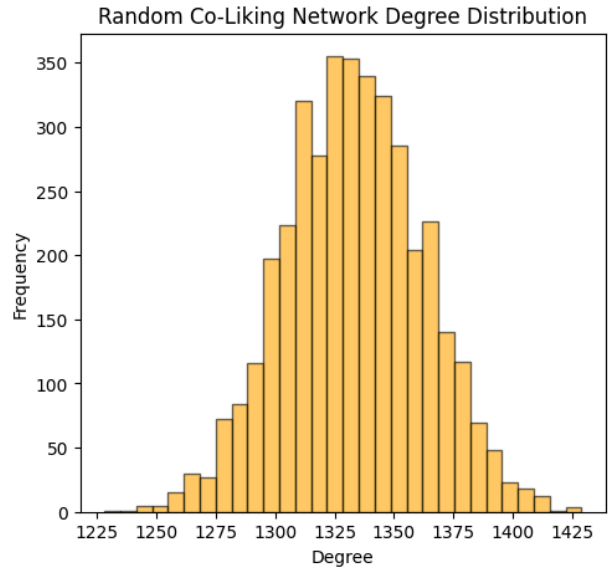
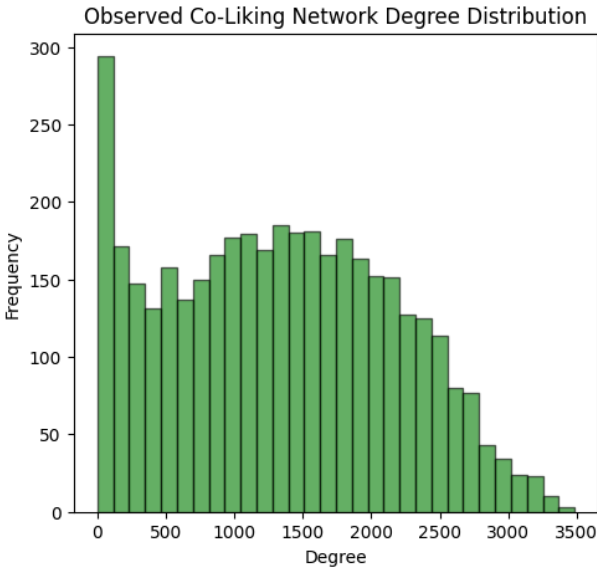
## Comparing the Friendship Network and Co-Liking Network with Their Random Baselines



### Friendship Network Degree Distributions

- **Observed Friendship Network**  
The degree distribution of the observed friendship network is highly skewed and right-tailed, with the majority of users having very few connections. A large proportion of nodes have a degree of 1 or 2, while only a handful of users are highly connected. This pattern reflects the way friendships usually form, with mutual connections within small and tightly knit groups.
- **Random Friendship Network**  
In contrast, the random friendship network follows a more balanced distribution, which is a characteristic of an Erdős-Rényi random graph. Here, most users have moderate connectivity, and there is less skewness compared to the observed network. This difference highlights that real-world friendships are far from random as they exhibit clear structural patterns by forming within localized and highly clustered communities rather than being evenly distributed across the network.

From this we can conclude that the friendship network is significantly more localized and structured than a random network which reinforces the idea that friendships emerge within small and interconnected groups rather than forming randomly.



### Co-Liking Network Degree Distributions

- Observed Co-Liking Network

The degree distribution of the observed Co-Liking Network exhibits a combination of skewness and semi-uniform connectivity. Some users have exceptionally high degrees, likely due to their preference for popular artists that link them to thousands of other users. So overall this network is more connected than the friendship network which illustrates how shared music preferences naturally form larger and more interconnected communities.

- Random Co-Liking Network

The random baseline for the Co-Liking Network follows a classic bell-curve distribution, with most nodes having similar degrees and fewer extreme outliers. This suggests that if co-liking relationships were purely random, users would have a much more balanced number of connections, and we would not observe highly influential users or strong clustering patterns. The absence of significant deviations in the random network reinforces that the organic structure of the observed Co-Liking Network is not accidental but emerges from user preferences and behavioral patterns.

From this we can draw the conclusion that the Co-Liking Network is highly structured and clustered, much more than its random counterpart. This confirms that music preferences naturally lead to community formation, where users are strongly connected based on their shared interests, rather than being evenly distributed in a random manner. The presence of high-degree nodes in the observed network highlights the central role of popular artists in shaping user interactions which create dense connectivity patterns that are absent in the random baseline.

## KL Divergence Analysis: Comparing Network Structure to Random Baselines

To quantify the differences between observed and random networks, we computed the Kullback-Leibler (KL) Divergence for each network's degree distribution. The KL Divergence measures how much the observed degree distribution deviates from the randomly generated counterpart.

The process involved:

1. Extracting degree sequences from both the observed and random networks.
2. Creating probability distributions by constructing histograms of degree frequencies.
3. Applying KL Divergence formula from `scipy.stats.entropy()`, which calculates how different the observed degree distribution is from the random baseline.

This method ensures a statistically rigorous comparison of structural patterns between real-world networks and their randomized equivalents.

```
import numpy as np
from scipy.stats import entropy

def compute_kl_divergence(degree_sequence_observed, degree_sequence_random):

    # Computing histogram (probability distributions)
    observed_hist, bin_edges = np.histogram(degree_sequence_observed, bins=50, density=True)
    random_hist, _ = np.histogram(degree_sequence_random, bins=bin_edges, density=True)

    # Adding small value to avoid division by zero
    observed_hist += 1e-10
    random_hist += 1e-10

    # Computing KL Divergence
    kl_div = entropy(observed_hist, random_hist)
    return kl_div

# Friendship Network KL Divergence
friendship_observed_degrees = [deg for _, deg in G_friendship.degree()]
friendship_random_degrees = [deg for _, deg in G_random_friendship.degree()]

kl_div_friendship = compute_kl_divergence(friendship_observed_degrees, friendship_random_degrees)

# Co-Liking Network KL Divergence
coliking_observed_degrees = [deg for _, deg in G_co_liking.degree()]
coliking_random_degrees = [deg for _, deg in G_random_co_liking.degree()]

kl_div_coliking = compute_kl_divergence(coliking_observed_degrees, coliking_random_degrees)

# Displaying the KL Divergence results
print(f"KL Divergence for Friendship Network: {kl_div_friendship:.4f}")
print(f"KL Divergence for Co-Liking Network: {kl_div_coliking:.4f}")
```

KL Divergence for Friendship Network: 0.2782  
KL Divergence for Co-Liking Network: 13.2643

The results we obtained are outlined below:

- Friendship Network KL Divergence: 0.2782
  - A relatively low value indicates that friendship connections are more evenly distributed and not drastically different from a random network. This suggests that

while friendship connections exhibit social clustering, they still retain an element of randomness and decentralization.

- Co-Liking Network KL Divergence: 13.2643
  - A much higher KL Divergence suggests that the observed Co-Liking Network is highly structured and deviates significantly from randomness. Users do not form connections randomly but rather cluster tightly around shared music interests, likely due to popular artists acting as central hubs.

From this it can be inferred that friendships tend to be more flexible and decentralized, which allow diverse social formations without strict clustering around specific individuals. In contrast, music preferences create highly structured clusters, where users are strongly connected through shared engagement with popular artists. The comparison with random baselines reinforces that while friendships are more dispersed and organic, co-liking behavior is significantly more structured. These findings confirm that friendship networks emerge more naturally, whereas music preference networks form tight-knit communities driven by the widespread appeal of certain artists.

## **Community Detection in the Co-Liking Network and Artist Preferences**

To understand how users cluster based on shared music preferences, we performed community detection on the Co-Liking Network using the Louvain algorithm in Gephi. This process allowed us to identify groups of users with strong co-liking ties, which revealed patterns in music taste.

### **Step 1: Running Louvain Community Detection**

- The Co-Liking Network was imported into Gephi for analysis.
- The Louvain community detection algorithm was applied by navigating to:
  - Statistics → Modularity and clicking Run.
  - Gephi assigned Community IDs to each user based on their music preference-based connectivity.

### **Step 2: Exporting Community Assignments**

- Once communities were detected, we exported the Node Table containing User IDs and Assigned Community IDs
- This file was saved as a CSV file and later uploaded to Google Colab for further processing.

### Step 3: Data Processing in Colab

After importing the community assignment CSV into Google Colab, we performed several data-processing steps to integrate community structure with user-artist preferences and analyze music preferences within each detected community.

#### 1. Loading the Community Assignments

- The CSV file containing user-to-community assignments was read into a Pandas DataFrame.
- The "Id" column was renamed to "User" for consistency across datasets.

#### 2. Merging Community Data with User-Artist Preferences

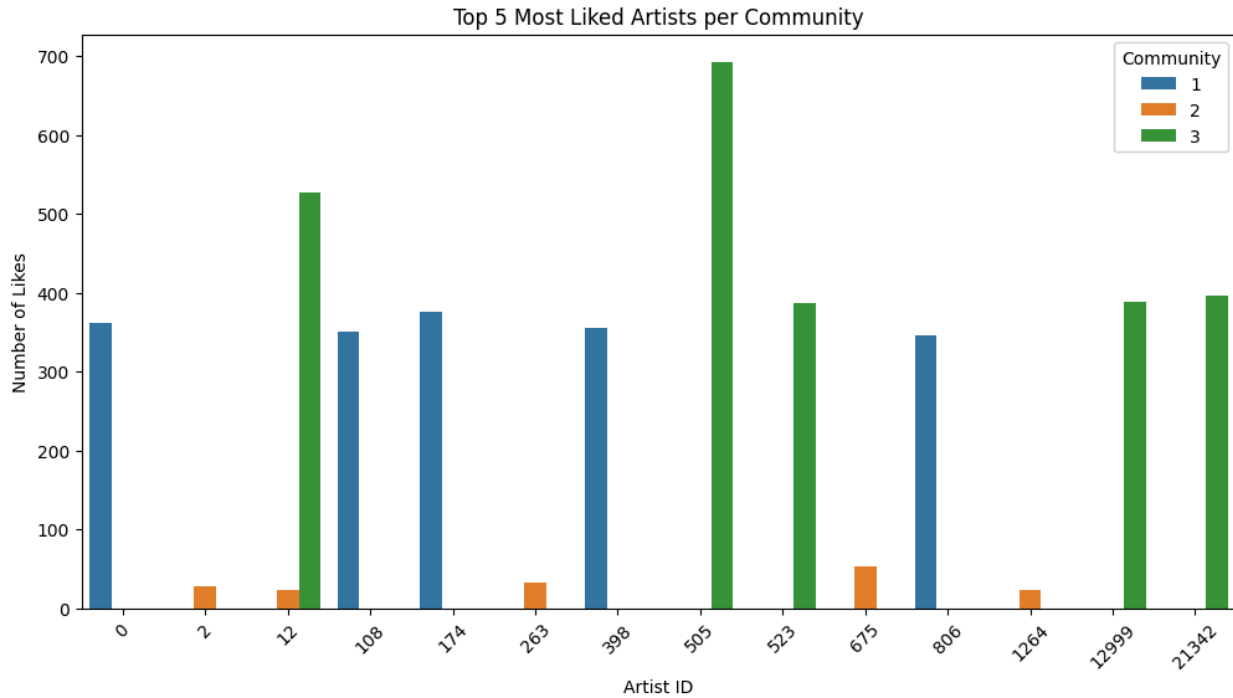
- Since we needed to analyze music preferences within each community, we merged:
  - Community Assignments (User → Community mapping).
  - User-Artist Data (User → List of Liked Artists).
- This inner merge ensured that only users with both community labels and artist preferences were included.

#### 3. Extracting and Analyzing Artist Preferences within Communities

To understand the music preferences within each detected Louvain Community, we performed the following steps:

- Exploding Artist Lists:
  - Since each user liked multiple artists, the dataset was exploded, so each row represented a (User, Community, Artist) triplet.
- Counting Likes Per Artist Per Community:
  - The number of likes for each artist-community combination was counted.
- Calculating the Share of Likes per Artist in Each Community:
  - The proportion of total likes each artist contributed in a given community was calculated.
  - This allowed us to determine whether artist preferences were highly concentrated (few dominant artists) or evenly distributed (diverse preferences).
- Extracting the Top 5 Most Liked Artists Per Community:
  - To identify leading preferences in each community, we extracted the top 5 artists per community based on their like share.
- Visualizing Artist Preferences by Community:
  - Finally, a bar plot was generated to compare artist preference patterns across different communities.

## Findings



### 1. Community 1 (Blue Bars)

- The top artists in this community have relatively even distribution in terms of likes.
- There is no single dominant artist, suggesting that users in this community have diverse music preferences.
- The relatively high like counts across multiple artists indicate that users share common interests but without extreme concentration on a few specific artists.

### 2. Community 2 (Orange Bars)

- The number of likes for the top artists is significantly lower compared to other communities.
- This suggests that users in this community have a more varied or individually unique taste in music, rather than collectively favoring a small set of artists.
- This aligns with the idea that certain communities do not form tightly around popular artists but instead exhibit a broad spectrum of preferences.

### 3. Community 3 (Green Bars)

- In contrast to Community 1, this community shows a strong concentration of likes on a few specific artists.
- One particular artist (ID 505) stands out with nearly 700 likes, which is the highest across all communities.
- This suggests that this community forms around a niche or a highly popular artist, creating a strong preference pattern.

The analysis reveals that community structure significantly shapes user preferences, with some groups forming around a few dominant artists while others exhibit more diverse musical engagement. Communities with high artist concentration suggest stronger peer influence, where shared interests reinforce collective listening habits. In contrast, communities with broader distributions of artist preferences indicate a more individualized musical landscape, where social ties do not strictly dictate taste. These findings highlight that music-based communities are not solely shaped by direct friendships but also by shared cultural influences, which reinforces the idea that network structure impacts how musical preferences form and evolve.

## **Clustering Coefficient and Music Diversity Analysis**

*(Focuses on whether highly connected users like a broader or narrower range of artists.)*

To examine the relationship between network clustering and individual music diversity, we performed the following steps:

### Computing Clustering Coefficients

- We calculated the clustering coefficient for each user in the Friendship Network using NetworkX's `nx.clustering()` function.
- The clustering coefficient measures the extent to which a user's friends are also friends with each other, capturing local connectivity within the network.
- The computed values were stored in a DataFrame for further processing.

### Binning Users by Clustering Coefficient

- Users were grouped into 10 fixed clustering coefficient bins using `pd.cut()`, with each bin covering an equal range of clustering coefficient values (e.g., 0.0–0.1, 0.1–0.2, etc.).
- This ensured that users were categorized into fixed intervals rather than equal-sized groups, and allowed for a structured comparison of music diversity across different levels of network clustering.

### Extracting User-Artist Preferences

- We loaded the `deezer_europe_features.json` file, which contains a list of artists liked by each user.
- This data was structured as a dictionary, where the keys were user IDs and the values were lists of liked artists.

### Merging Network and Preference Data

- To analyze the connection between clustering coefficients and music diversity, we merged the user clustering data with the user-artist preference data using Pandas.
- This allowed us to examine how the number of unique artists liked per user varied across different clustering coefficient bins.

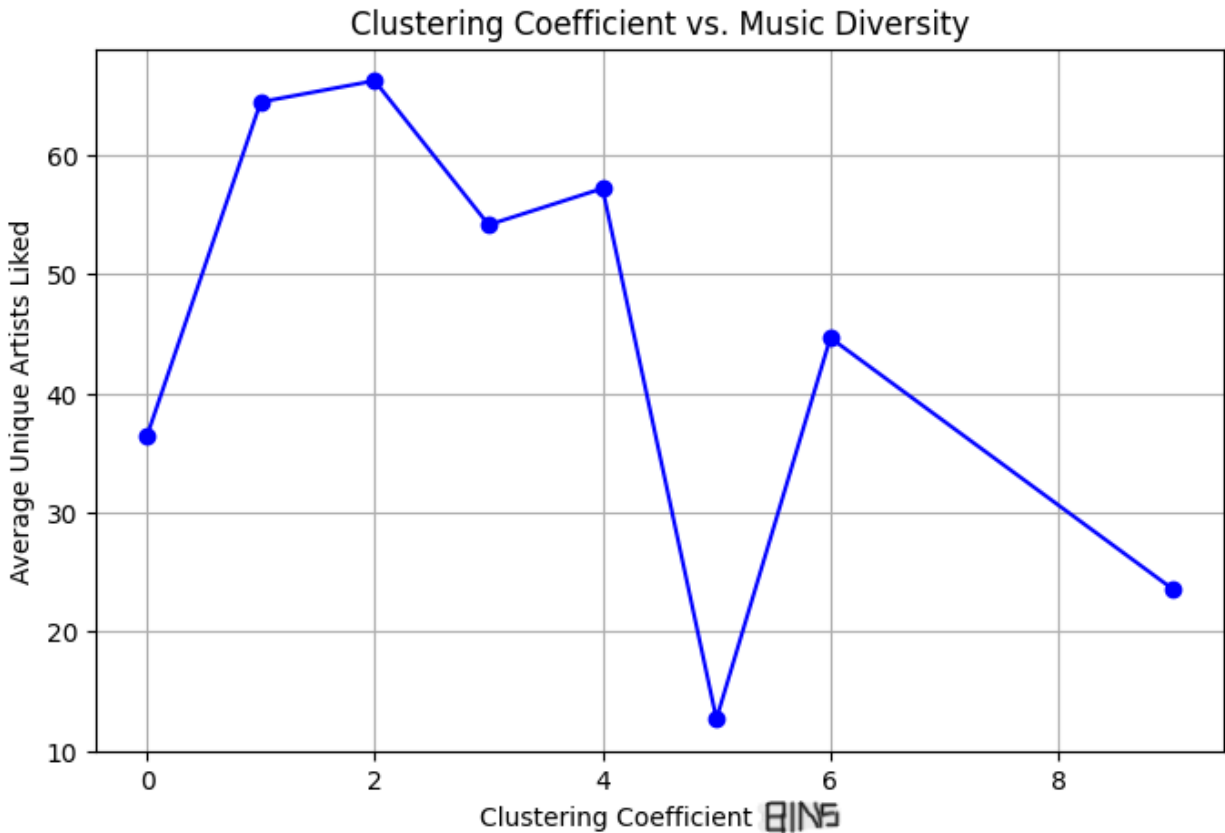
### Computing Music Diversity

- For each user, we counted the number of unique artists they liked.
- We then computed the average number of unique artists liked per clustering coefficient bin to analyze the trend.

### Visualization

- A line graph was plotted to illustrate how music diversity varies with clustering coefficient bins.

### *Findings*



1. Users with very low clustering coefficients (Bins 0-2) exhibit high music diversity.

- The average number of unique artists liked increases steeply, with a peak around Bin 2.
- This suggests that individuals in loosely connected networks are exposed to a broader variety of artists, likely due to diverse social influences.

2. Users in mid-range clustering coefficients (Bins 3-4) maintain relatively high but fluctuating music diversity.

- The diversity slightly declines but remains stable. This indicates that users in moderately connected friendship groups retain a mix of both broad and localized music preferences.

3. A sharp decline in diversity occurs at Bin 5.

- The average number of unique artists drops drastically and reaches its lowest point.
- This suggests that individuals in this clustering range experience strong social reinforcement, where peer influence highly restricts musical variety.

4. Users with high clustering coefficients (Bin 6) experience a sudden recovery in diversity.

- Unlike Bin 5, these users show a significant increase in the number of unique artists liked.
- This suggests that some tightly-knit groups break away from extreme preference concentration and regain exposure to a broader set of artists.

5. Users with extremely high clustering coefficients (Bins 7-9) experience a gradual decline in diversity.

- While some high-clustering users sustain moderate diversity, the overall trend suggests that as network connectivity tightens, users increasingly share musical preferences and reduce exposure to diverse artists.

These findings confirm that social network structure plays a crucial role in shaping music diversity. Users in loosely connected networks explore a broader variety of music, while those in moderate to high clustering environments show fluctuating diversity patterns, possibly influenced by a mix of peer influence and external exposure.

The sharp drop at Bin 5 suggests a threshold where social ties begin to significantly restrict diversity, while the partial recovery at Bin 6 implies that not all tightly-knit groups conform strictly to shared preferences. Overall, the declining trend at the highest clustering coefficient bins reinforces the idea that highly interconnected communities tend to form strong musical identities, which limits the variety of artists users engage with.

## **The Influence of Network Clustering on Artist Concentration in Music Preferences**

*(Focuses on whether highly connected users concentrate their likes on fewer popular artists. In other words, do tightly connected users like the same artists?)*

To examine the relationship between network clustering and artist concentration, we analyzed how the share of total likes received by the top 5 most-liked artists varied across users with different clustering coefficients. Instead of using fixed bins (e.g., 0.0-0.1, 0.1-0.2), we adopted a decile-based binning approach, grouping users into ten equal-sized bins based on their clustering coefficient.

### **Step 1: Binning Users by Clustering Coefficient**

- Users were grouped into deciles (10 bins) using `qcut()`, which ensured a balanced distribution across different clustering coefficient levels.

## Step 2: Computing Artist Concentration per Decile

- The dataset was exploded so that each user-artist pair appeared as a separate row.
- The total number of likes per artist and total likes per decile were computed to normalize artist concentration.
- The like share of each artist within a decile was computed as:

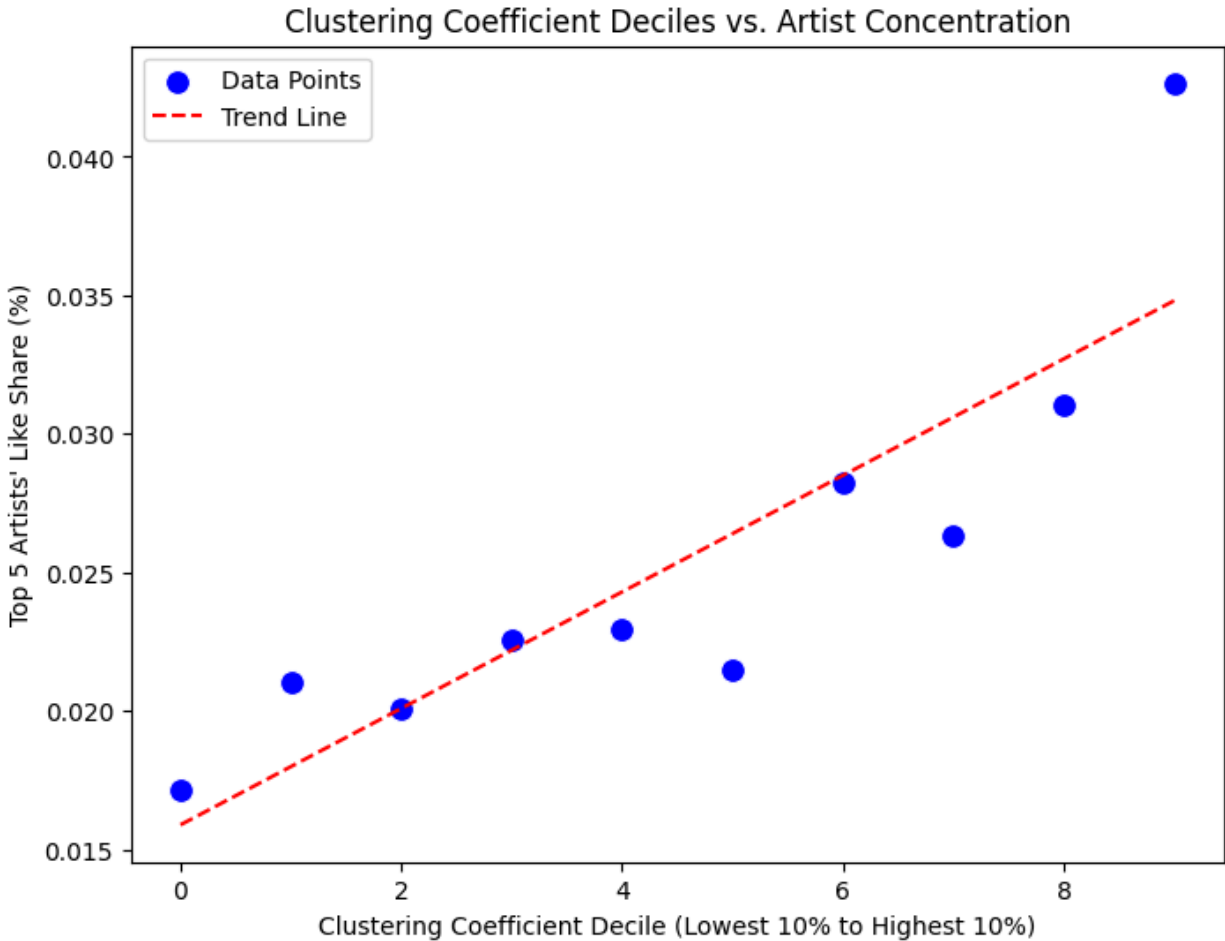
$$\text{Like Share} = \frac{\text{Artist's Like Count in Decile}}{\text{Total Likes in Decile}}$$

- To measure artist concentration, we extracted the top 5 most-liked artists per decile and summed their total like shares. It represents the extent to which music preferences within a decile are dominated by a few popular artists.

## Step 3: Visualizing Trends in Artist Concentration

- A scatter plot was generated, with the clustering coefficient decile on the x-axis and the total like share of the top 5 artists on the y-axis.
- A linear trend line was fitted to highlight the overall relationship.

*Findings:*



Users in highly clustered communities focus their likes on a small set of dominant artists.

- As the clustering coefficient increases (moving right on the X-axis), the share of total likes received by the top 5 artists also increases.
- The top 10% (Decile 9) has the highest concentration (~4%), meaning a small set of artists dominates the music preferences within these tightly connected groups.
- This suggests that strong social clustering reinforces shared listening habits, reducing overall diversity in artist choices.

Lower clustering users exhibit more diverse musical preferences.

- Users in the bottom deciles (0-2) show lower artist concentration (~1.5% to 2%).
- This suggests that low-clustering users engage with a broader range of artists, likely due to exposure to more varied social influences.

These findings confirm that network clustering plays a crucial role in shaping music preferences. Users in loosely connected social structures explore a wider variety of music, whereas those in highly clustered networks converge around a smaller number of dominant artists. The clear positive trend in the scatter plot reinforces the idea that social connectivity strengthens shared musical interests, and reduces individual diversity in artist preferences.

## **Future Directions and Implications**

While this project provides valuable insights into the relationship between social structures and music preferences, several avenues for future research remain open. Expanding upon these findings could lead to deeper understandings of how user behaviors are shaped by social and cultural factors.

### **1. Refining the Network Models**

- **Temporal Analysis:** Our study considers a static snapshot of user interactions. Future work could incorporate temporal aspects. This would allow for studying how users' social circles and preferences shift, how communities form and dissolve, and whether long-term friendships drive music taste homogenization.
- **Genre-Level Analysis:** Instead of aggregating music preferences at the artist level, future research could explore whether clustering impacts users' preferences for specific genres rather than individual artists. This could provide a deeper understanding of how different music categories spread through social ties.

### **2. Implications for Music Recommendation Systems**

- **Personalized Recommendations Based on Social Structure:** Given that users in highly clustered groups tend to favor a smaller set of dominant artists, recommendation algorithms could incorporate network clustering as a feature to balance between reinforcing existing tastes and introducing diverse music options.
- **Community-Based Recommendation Models:** Instead of recommending music solely based on individual preferences, models could be trained to predict user preferences based on community structures by leveraging insights from co-liking behaviors to make group-aware recommendations.

### **3. Broader Applications Beyond Music**

- **Extending to Other Domains:** The methodology used in this study, which is comparing direct social ties with interest-based networks, can be applied to other domains such as

movie preferences, book recommendations, and online content engagement to analyze whether similar clustering effects exist in different cultural consumption patterns.

- **Impact on Marketing Strategies:** Music labels and artists could leverage these insights to tailor marketing campaigns based on social clustering. Highly clustered communities may respond well to exclusive artist-focused content, while loosely connected users might be more receptive to discovery-driven music promotions.

## **Conclusion:**

This study explored the relationship between social network structures and music preferences and focused on how friendship ties and shared music interests shape user interactions. By analyzing the Friendship Network and Co-Liking Network, we demonstrated that social connectivity significantly influences listening behavior. Our comparison with random baseline networks revealed that both networks exhibit high structural organization, with the Co-Liking Network forming stronger and more clustered communities around shared musical interests.

Our findings show that highly clustered users tend to concentrate their likes on a smaller set of dominant artists, while low-clustering users engage with a broader range of artists. This suggests that tight-knit social groups reinforce shared listening habits, which lead to less diverse music preferences, whereas loosely connected users maintain greater diversity in their music choices. The KL Divergence analysis further confirmed that the Co-Liking Network is far more structured than a random equivalent. This reinforces the idea that music preferences naturally lead to community formation.

Through modularity-based community detection, we identified that different user groups exhibit varying degrees of artist concentration, with some communities displaying strong preferences for a few key artists, while others maintain more evenly distributed musical interests. Additionally, our clustering coefficient analysis revealed that there is a clear positive trend between social clustering and artist concentration. This further highlights the role of network structure in shaping listening and liking behavior.

These insights have important implications for music recommendation systems, targeted marketing, and social influence research. Future work could extend this analysis by incorporating temporal trends, genre-based preferences, and personalized recommendation models that account for social clustering effects. Overall, this study highlights the interplay between social structures and cultural consumption, and demonstrates how shared music preferences shape and are shaped by user connectivity in online streaming platforms.

## References

We used several online resources to implement network analysis using NetworkX and Pandas, understand and compute KL Divergence, and interpret findings in the context of social network and music preference studies. These references provided essential guidance on data manipulation, statistical analysis, and theoretical insights, as outlined below:

1. Dataset and Related Paper: Stanford Network Analysis Project (SNAP)
  - *Feather-Dataset: Deezer Europe Social Network and Music Preferences*  
<https://snap.stanford.edu/data/feather-deezer-social.html>
2. Calculating Kullback-Leibler (KL) Divergence in Python
  - *Machine Learning Mastery: "How to Calculate the KL Divergence for Machine Learning"*  
[machinelearningmastery.com](https://machinelearningmastery.com)
  - *Statology: "How to Calculate KL Divergence in Python (Including Example)"*  
[statology.org](https://statology.org)
3. Utilizing NetworkX for Network Analysis
  - *NetworkX Official Documentation: "NetworkX: Network Analysis in Python"*  
<https://networkx.github.io/documentation/stable/>
  - *Towards Data Science: "Network Analysis Made Simple"*  
<https://towardsdatascience.com/network-analysis-made-simple-cc5c1be0f0ea>
4. Employing Pandas for Data Manipulation
  - *Pandas Official Documentation: "pandas: Python Data Analysis Library"*  
<https://pandas.pydata.org/docs/>
  - *Real Python: "Pandas Tutorial: DataFrames in Python"*  
<https://realpython.com/pandas-dataframe/>
5. Interpreting Findings in Social Network and Music Preference Studies
  - *ResearchGate: "Mining the Dynamics of Music Preferences from a Social Networking Site"*  
[researchgate.net](https://researchgate.net)
  - *SAGE Journals: "Music Preference, Social Identity, and Collective Self-Esteem"*  
[journals.sagepub.com](https://journals.sagepub.com)