

Network Analysis of Spotify Playlists Using Artist Co-Occurrence

Massimo Rondelli

massimo.rondelli@studio.unibo.it

Department of Computer Science and Engineering, University of Bologna

Nadia Farokhpay

nadia.farokhpay@studio.unibo.it

Department of Computer Science and Engineering, University of Bologna

Michele Dinelli

michele.dinelli5@studio.unibo.it

Department of Computer Science and Engineering, University of Bologna

Youssef Hanna

youssefawni.hanna@studio.unibo.it

Department of Computer Science and Engineering, University of Bologna

1 Introduction

Our individual musical tastes act as a distinctive identifier, a unique fingerprint reflecting our preferences. While the amount of engagement with music varies, everyone must have a personal musical taste that is unique. An expression of music taste is creating music playlists. These are not simply random lists of songs, they are curated collections where genres, artists, and even musical periods are often mixed, forming a reflection of an individual's unique listening habits. When we look at these playlists, it's clear that if two artists show up together a lot, there's some kind of connection, maybe their music just fits well, or fans who like one also tend to like the other.

This report explores connections between artists on Spotify [1]. We're specifically looking at the relationships that pop up when artists are found together in playlists made by users. By studying how often artists appear together, we can build a network of these artists. This helps us analyze how artists are linked and understand how people's music choices connect different musicians.

2 Problem and Motivation

Streaming services like Spotify changed how we listen to music and show off our tastes through playlists. Even though each playlist is personal, together they reveal patterns. Artists often appear together, hinting at latent associations based on genre, mood, or how people actually use music. Unlike other recommendation methods, playlists show us how users genuinely connect artists.

Our motivation comes from a key observation: while artists are interconnected through various means, a significant source of valuable information about these relationships is often skipped or doesn't get the attention it deserves. We're referring to the important connections established by users through their personal music playlists, they aren't analyzed nor used to extract knowledge while may represent a reflection of how listeners perceive artistic proximity [2].

In Section 3 we describe the data source. In Section 4 we analyze the validity and reliability of this study. In Section 5.1 we analyze centrality measures to understand which are most influential nodes and understand the network structure. In Section 5.2, Section 5.3, Section 5.4

and Section 5.8 we analyze network properties such as scale-free property and small world effect to determine node roles and network robustness. In Section 5.5 and Section 5.7 we investigate the possibility to use network analysis to build artists recommendation heuristics. In Section 5.6 we analyze network periphery. In Section 6 we sum up our study with a conclusion section that recaps our findings. Finally we conclude with Section 7 which is a critique on this study.

3 Dataset

The dataset used is the continuation of the RecSys Challenge 2018 [3] and it contains 1,000,000 music playlists, created by users on the Spotify platform between January 2010 and October 2017. The dataset's original task is automatic playlist continuation: given a seed playlist title and/or an initial set of tracks in a playlist, predict the next tracks. The dataset is publicly available [4] but it's not user-friendly because it's divided into multiple slices formatted as .json files, and it's huge (5GB). We randomly sampled 10 dataset slices in order to drastically reduce the dimension of the upcoming network. This left us with 10,000 playlists. Since we cannot (computationally) analyze such a network, we computed a subgraph i by reducing the number of playlists and keeping connected components only.

The result is a monomodal, undirected and weighted co-occurrence network with 1,139 nodes and 17,908 edges. Nodes represent artists, edges represent the co-occurrence of artists in playlists and edge weights are the cumulative sum of the artists' co-occurrence frequencies (see Appendix A Figure 19).

4 Validity and Reliability

This analysis uses a publicly available dataset [3] that is frequently employed by researchers for investigating artist relationships in user-generated playlists. It is also used in music recommendation challenges, with a total of over 15,000 users and numerous submissions from almost 300 different teams. Although only a subset of 10,000 playlists was used due to computational constraints, the random sampling strategy ensures representativeness of the larger dataset and maintains the generalizability of our findings. The entire methodology of this study is designed for transparency and reproducibility. All code and resources are open-source [5], [6], allowing easy verification and replication of our findings. Any elements of randomness introduced are controlled through seeded generators.

The technical implementation of the study relies on established standards de facto within the scientific community. Network analysis is performed using NetworkX [7]. Visualizations are generated with the well-known Matplotlib and Seaborn libraries [8], [9].

5 Measures and Results

Computing network measures helps one to summarize and extract insight from complex network data in a manageable and interpretable way. Analyzing the raw network directly is impractical due to its size and complexity. Music playlists, like many real-world networks, form intricate webs of relationships. By applying mathematical measures that capture interesting features of network structure quantitatively, it's easier to extract important information such as structural characteristics, patterns, node importances and network properties [10].

5.1 Centrality Measures

Network centrality metrics quantify the influence or importance of a node in a network; however, there is no generalized definition of centrality [11], as a result, many centrality measures exist [12].

5.1.1 Degree Centrality

Degree centrality is applied to identify artists who serve as central figures in user playlists, providing a first glimpse into influence and popularity across the network. The simplest centrality metric is degree centrality. In an undirected graph, it's defined by the number of edges attached to a node. Defining this in terms of the network's adjacency matrix $W \in R^{N \times N}$ gives the following

$$c_i = \sum_{j=1}^N W_{ij} \quad (1)$$

Artists with a high degree of centrality appear alongside many different artists in playlists, which means they are broadly connected across listener tastes (likely popular). Artists with a high co-occurrence rate are more likely to be frequent collaborators or playlist “glue” (these artists may be useful for playlist generation or recommendation models).

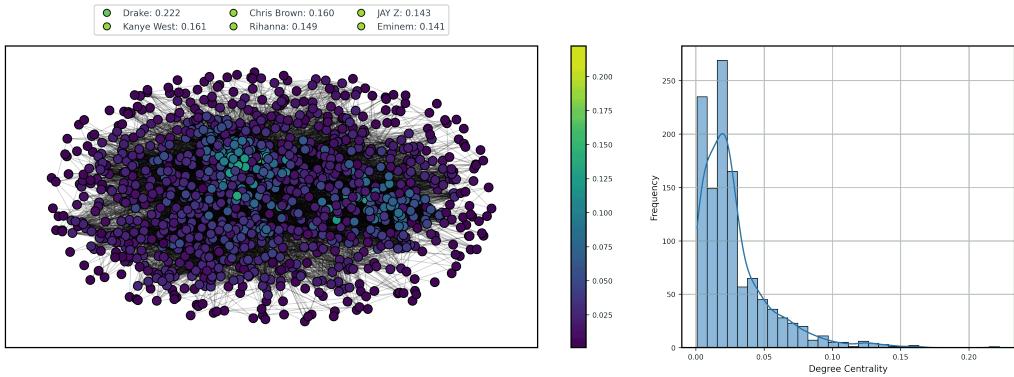


Figure 1: Degree centrality in the artist co-occurrence network and degree centrality distribution

Figure 1 shows the network where each node represents an artist and edges indicate co-occurrence in playlists. Node color is proportional to the node's degree centrality, with brighter nodes having higher centrality. On the right it's shown the distribution of degree centrality values across the network. As one would expect, the distribution is long-tailed, which means that only a few artists get the highest degree centrality scores hence representing popular nodes.

5.1.2 Weighted Degree Centrality

We apply weighted degree centrality to go beyond simple connectivity and capture the strength of artist relationships in playlists, reflecting not just presence but frequency of co-occurrence. Since the network obtained is weighted, we can further analyze degree centrality using weighted degree centrality which is an extension of degree centrality that takes into account edge weights. Given a weighted adjacency matrix $W \in R^{N \times N}$, where $W_{ij}^{(w)}$ denotes the weight of the edge between nodes i and j , the weighted degree centrality $c_i^{(w)}$ of node i is defined as

$$c_i^{(w)} = \sum_{j=1}^N W_{ij} \quad (2)$$

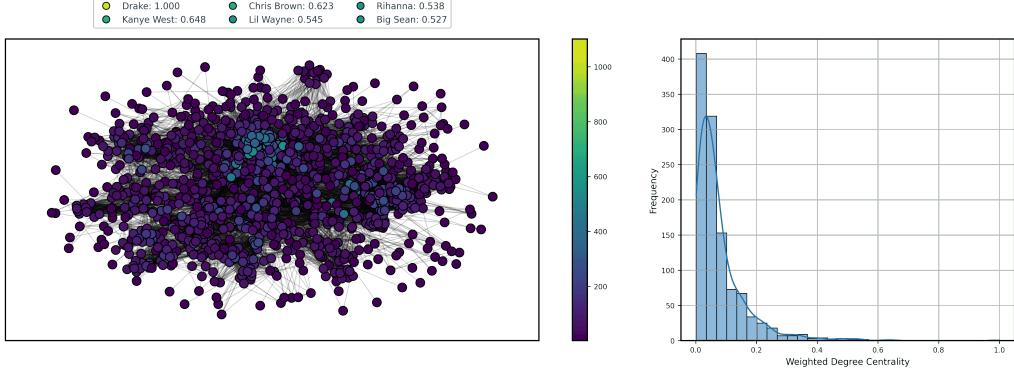


Figure 2: Weighted degree centrality in the artist co-occurrence network and weighted degree centrality distribution

Figure 2 shows the network, where nodes with higher centrality appear brighter. On the right it's shown the distribution of weighted degree centrality, showing again a highly skewed (right) distribution. This suggests that a few artists dominate playlist co-occurrences, likely due to frequent collaborations or broad popularity.

5.1.3 Eigenvector Centrality

Unlike simpler metrics that count direct links, eigenvector centrality captures influence through association with other important nodes [11]. An artist may be central not by reaching many others, but by connecting to a few highly influential ones. This recursive definition emphasizes quality over quantity of connections. In our case, since the network is undirected and analyzed as a single component, we don't apply normalization or zero-trailing adjustments [10].

Given a weighted adjacency matrix $A \in R^{N \times N}$, where A_{ij} denotes the weight of the edge between nodes i and j the eigenvector centrality of node v is proportional to the sum of the eigenvectors centralities of v 's neighbors and it's defined by the following equation

$$x_v = \frac{1}{\lambda} \sum_{j \in M(v)} A_{ij} x_j \quad (3)$$

where $1/\lambda$ is a constant and $M(v)$ is a function that returns the neighbors of the node v . With a small rearrangement, this can be rewritten in vector notation as the eigenvector equation

$$Ax = \lambda x \quad (4)$$

In general, there will be many different eigenvalues λ for which a non-zero eigenvector solution exists. However, the Perron–Frobenius theorem implies that only the greatest eigenvalue results in the desired centrality measure [10], [13].

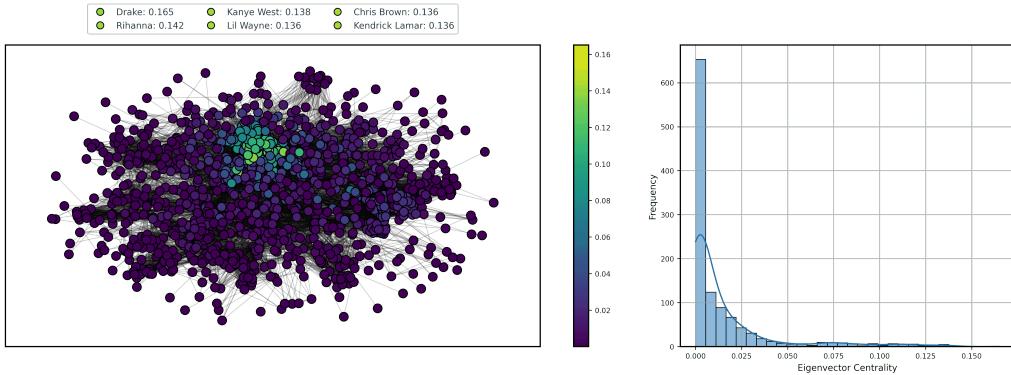


Figure 3: Eigenvector centrality in the artist co-occurrence network

Figure 3 shows eigencentrality across the network, with brighter nodes indicating higher values. The histogram on the right is right-skewed, suggesting that a few artists hold most of the level of influence within the network.

These metrics seem redundant but eigencentrality captures broader structural importance by factoring in the influence of a node’s neighbors, not just how many they are, but how connected they are.

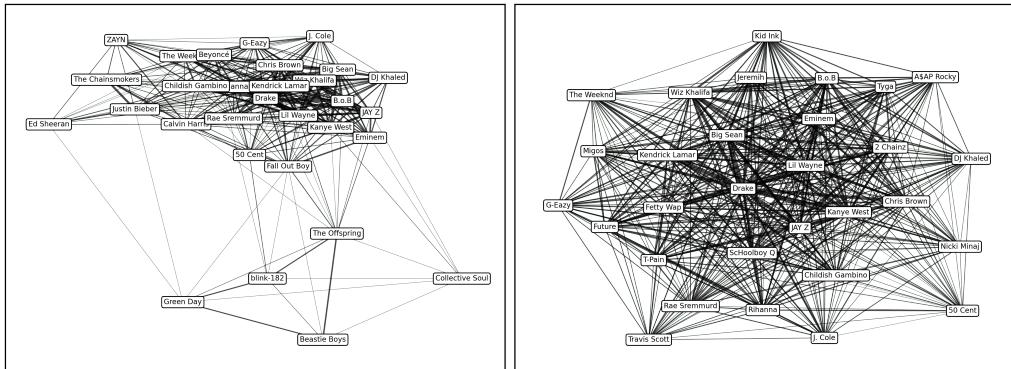


Figure 4: Top 30 artists subgraphs by degree centrality and eigencentrality

Figure 4 compares subgraphs based on different centrality measures. The left shows the top 30 by degree centrality; the right, by eigencentrality. It’s clear that the two are structurally distinct: the degree-based subgraph is less connected, with lower density (0.6759 vs 0.9977) and average degree (19.60 vs 28.93). This shows how local and global centrality capture different structural roles [14].

5.1.4 Closeness Centrality

In collaborative playlists, some artists naturally emerge as intermediaries, linking otherwise separate listening communities. Betweenness centrality helps uncover these bridges. Closeness centrality of a node u is the reciprocal of the average shortest path distance to u over all $n - 1$ reachable nodes [15]. Closeness centrality uses the notion of mean distance between a node u and other nodes in the network defined as $\ell_u = \frac{1}{n} \sum_v d(u, v)$. Closeness centrality is basically ℓ_u^{-1}

$$C(u) = \frac{n - 1}{\sum_{v \neq u} (d(u, v))} \quad (5)$$

where $d(v, u)$ is the shortest-path distance between v and u and n is the number of nodes.

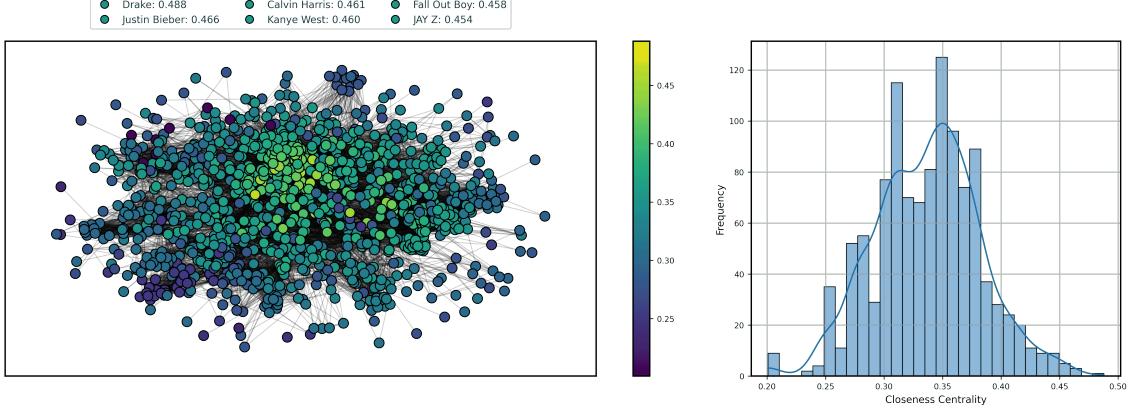


Figure 5: Closeness centrality in the artist co-occurrence network

Figure 5 shows the closeness centrality metric applied to the network. Node color is proportional to closeness centrality with brighter colors indicating higher centrality. The histogram on the right shows the closeness centrality distribution, which appears approximately normal. This is because the values of the mean distance typically have a small range, as they are limited by the diameter of the network, which is typically between 1 and $\log n$. Our network has a diameter of 7 and 1139 nodes, so bingo because $\log(1139) \simeq 7$.

Since closeness assumes full reachability, it fails in disconnected graphs [10]. But here, the network is connected, so no alternative like harmonic centrality is needed.

5.1.5 Betweenness Centrality

In a co-occurrence network of artists, some nodes act as key bridges between distant clusters, betweenness centrality helps identify these connectors. It measures how often a node lies on the shortest paths between all other nodes, highlighting its role in linking the network. Formally, for a node $n \in N$, where N is the set of all nodes, betweenness centrality is defined as:

$$C_B(n) = \sum_{s \neq n \neq d} \frac{\sigma_{sd}(n)}{\sigma_{sd}} \quad (6)$$

where σ_{sd} is the total number of paths between s and d and $\sigma_{sd}(n)$ is the number of paths between the pair that pass through node n .

This captures how much “traffic” a node handles. To account for multiple shortest paths and make values comparable, betweenness is often normalized by dividing by the total number of possible node pairs, scaling the measure between 0 and 1.

$$C_B^{\text{norm}}(n) = \frac{1}{\binom{n^2}{2}} \sum_{s \neq n \neq d} \frac{\sigma_{sd}(n)}{\sigma_{sd}} \quad (7)$$

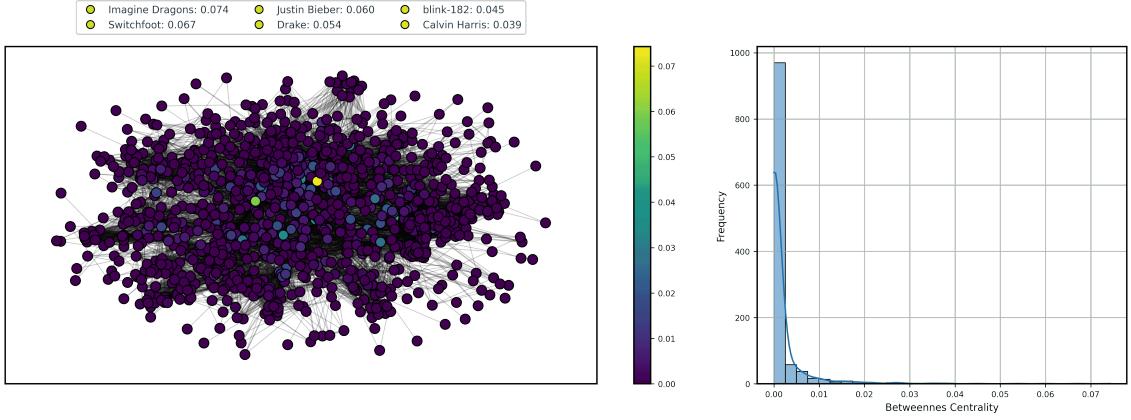


Figure 6: Betweenness centrality in the artist co-occurrence network

Figure 6 shows the betweenness centrality metric applied to the network. Node color is proportional to betweenness centrality with brighter colors indicating higher centrality. On the right there is the distribution of betweenness centrality that appears highly skewed on the right. This means that only a small number of nodes act as “gateways” gluing different parts of the networks together.

5.2 Power Laws and Scale-free Networks

Understanding whether artist centrality follows a power-law helps assess the network’s robustness and the concentration of influence, both key to analyzing real-world musical ecosystems. Networks whose degree distribution follows a power-law are known as scale-free networks. These networks are notably resilient, able to maintain overall connectivity even when a significant number of nodes fail [16]. To detect power-law behavior, one can use the cumulative distribution function (CDF), which gives the probability that a node has degree $\geq d$

$$P_d = \sum_{d' \neq d}^{\infty} p_{d'} \quad (8)$$

If the degree distribution follows a power-law, it can be described in logarithmic form as

$$p_d = -\alpha \ln(d) + c \quad (9)$$

which simplifies to

$$p_d = Cd^{-\alpha} \quad (10)$$

where $C = e^c$ is a normalization constant and α is the power-law exponent. To quantitatively assess the power-law fit, α can be estimated using

$$\alpha = 1 + n \left(\sum_i \ln \frac{d_i}{d_{\min} - \frac{1}{2}} \right)^{-1} \quad (11)$$

Empirically, power-law distributions in networks typically fall within the range $2 \leq \alpha \leq 3$.

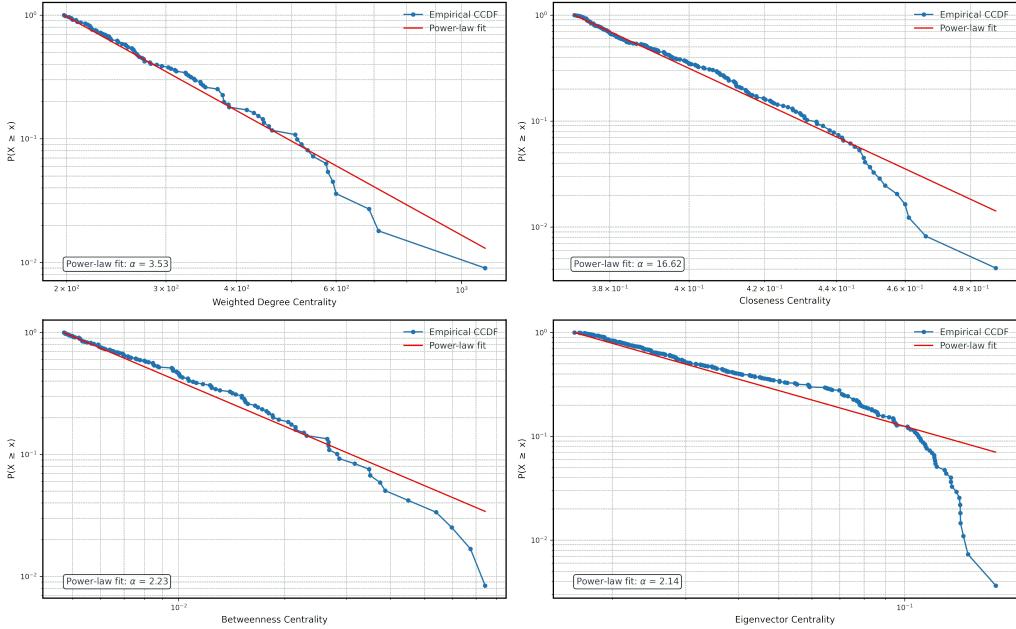


Figure 7: Power Law fit of weighted centrality, closeness centrality, betweenness centrality and eigencentrality in the artist co-occurrence network

Figure 7 shows the power-law fit for four centrality measures from Section 5.1: weighted, closeness, betweenness, and eigenvector centrality. The plots display CDF vs. metric values on a log-log scale. All follow a scale-free pattern, except closeness centrality, which is actually expected, as its values are limited by network diameter [17] (see Section 5.1.4).

5.3 Small World Effect

A small-world network is a graph characterized by a high clustering coefficient (0.527 in our network) and low distances on average (3.026). In an example of the social network, high clustering implies the high probability that two friends of one person are friends themselves [18]. Networks show small-world effects when

$$\ell \propto \log n \quad (12)$$

where ℓ is the mean distance of the whole network represented by

$$\ell = \frac{\sum_{ij} d_{ij}}{n^2} \quad (13)$$

and d_{ij} is the shortest path between node i and j .

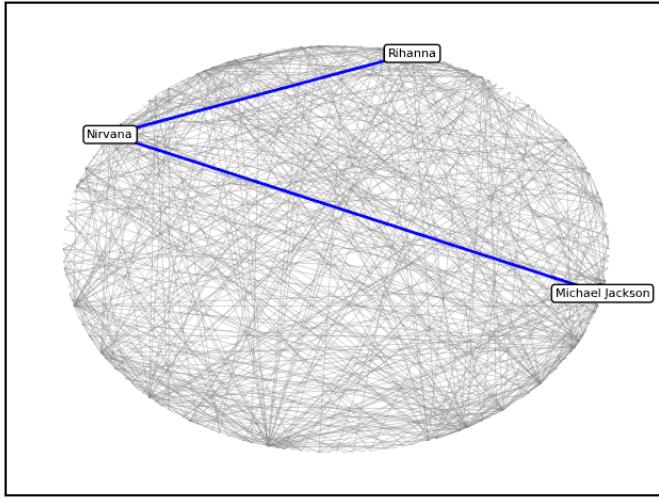


Figure 8: Small World Effect

We can safely say that the network we're considering is a “small world” since $\ell = 3.02649$ and $\log n = 7.04$ [5]. Nirvana acts as a mediator between Michael Jackson and Rihanna.

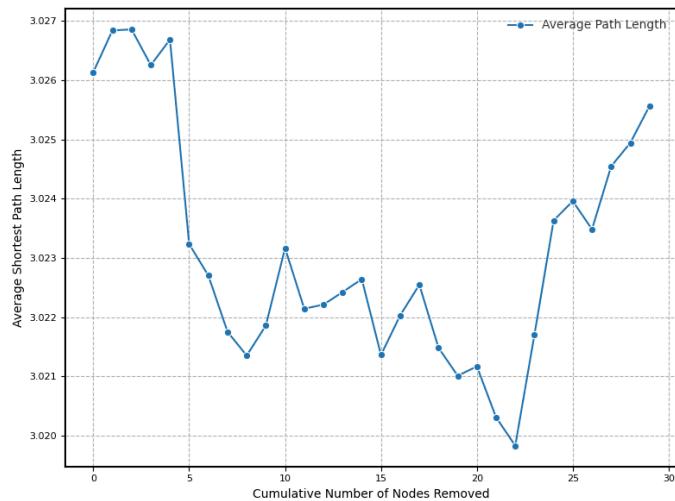


Figure 9: Network Resilience: Average Shortest Path Length vs Iterative Node Removal

Since a network that show small world effect is generally robust to random permutations we tested iterative node removal against the average shortest path length ℓ . In Figure 9 we show that this intuition is true since there is no linear nor polynomial/exponential relationship between the two axes. The ℓ variation is really small.

We formalize small world-ness results using random graph theory thus calculating ω defined as

$$\omega = \frac{L_r}{L} - \frac{C}{C_l} \quad (14)$$

where $\frac{L_r}{L}$ is the ratio between the average shortest path of the Erdős–Rényi random graph with the same size the network and the average shortest path of the network and $\frac{C}{C_l}$ is the ratio between the average clustering coefficient of the network and the average clustering coefficient

of a regular network of the same size. The network shows $\omega = -0.2904$ [5]. Graph used are showed in Appendix C Figure 22.

5.4 Average Local Clustering Coefficients vs Node Degree

While node degree is a notion already introduced in Section 5.1, Local Clustering Coefficients (LCCs) of a node i is calculated, by examining all distinct pairs of its neighbors. We count how many of these pairs are directly connected to each other and divide that by the total number of possible neighbor pairs. If the degree of node i is k_i , then the number of possible pairs is given by the binomial coefficient $\binom{k_i}{2}$. The coefficient thus represents the likelihood that two neighbors of a node are also neighbors themselves.

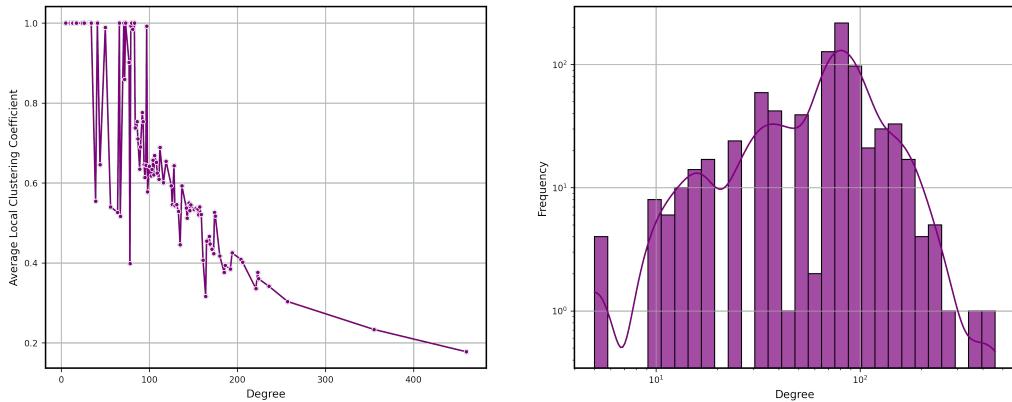


Figure 10: Average Local Clustering Coefficients vs Node Degree

In Spotify playlists, the clustering coefficient-degree relationship reveals how artists are positioned in the musical ecosystem. As shown in Figure 10, average clustering tends to decrease as degree increases.

Artists with low degrees and high clustering are often niche, they have few but tightly connected collaborators. In contrast, high-degree artists with low clustering are typically popular figures who span across diverse playlists and link different musical styles.

5.5 Groups of Nodes

In network analysis, groups of nodes are subsets of nodes that are more densely connected to each other than to the rest of the network. Structures such as cliques, k-cores and k-components help reveal hidden patterns and relationships. Considering artist co-occurrence network, they can uncover musical communities based on shared audience preferences or stylistic similarity.

5.5.1 Clique Analysis

A clique is a group of artists who all appear together in at least one playlist, meaning every artist in the group is directly connected to every other artist based on shared playlist presence. We identified 3,043 maximal cliques with an average size of ~ 12 artists.

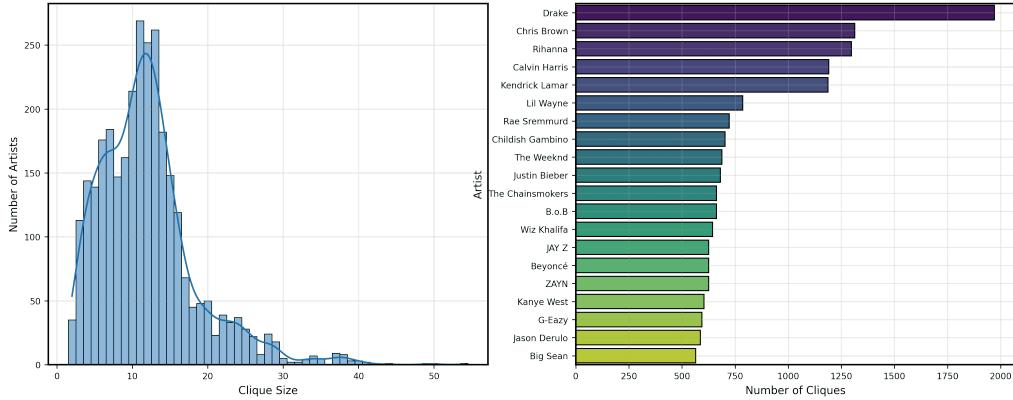


Figure 11: Combined clique analysis with distribution of clique sizes and artist frequencies in cliques

The left panel of Figure 11 shows clique sizes ranging from 2 to 54, with most between 10–14 artists. The right panel lists the top 20 artists by clique participation, Drake (1,971), Chris Brown (1,314), and Rihanna (1,298) lead, highlighting their versatility and cross-genre appeal.

We show the three largest cliques to examine the structure of the most significant artist communities.

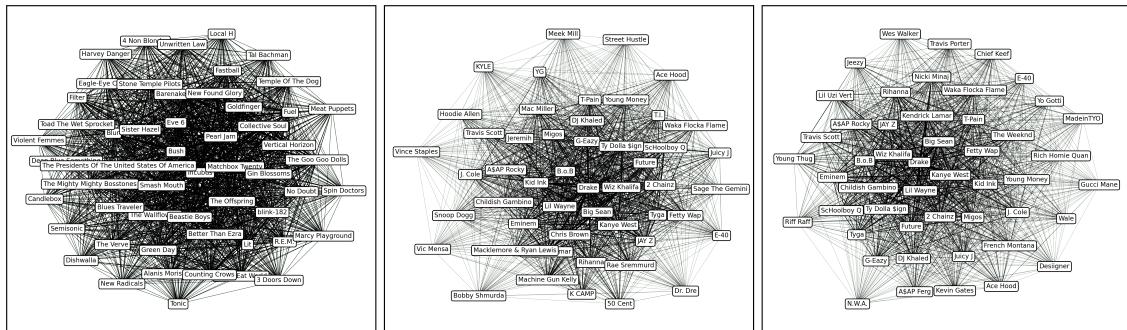


Figure 12: Visualization of the three largest clique containing 54, 50 and 49 artists respectively. Edge thickness represents the strength of co-occurrence relationships.

In Figure 12 the left panel shows the largest clique of 54 artists in a fully connected subgraph. This massive community can represent either artists from related genres (in this case predominantly rock and punk). As shown in Figure 12 center and right panel, rap and hip-hop are the predominant genres. The clique analysis captures audience-driven associations in artist groupings, revealing how listeners and curators organize music on Spotify. While many groups are genre-based, others may form around shared moods, cultural trends or recommendation algorithms, reflecting the complex nature of music consumption. Clique analysis offers a valuable approach for enhancing artist recommendations. For example if a user's playlist contains multiple artists that form a clique within our network, it strongly suggests a cohesive musical taste. In such cases, we can confidently recommend other artists belonging to the same clique who are not yet present in the user's playlist.

5.5.2 K-Cores

The K-core is a flexible way to group nodes: each node is connected to at least K others in the set, unlike cliques which require full connectivity. It's easy to compute and useful for uncovering structure. In our case, K-cores help build recommendation heuristics. Even without direct links, artists in the same K-core often share an underlying affinity. If artist A and B share neighbors within a K-core, they can be recommended to each other's listeners. This approach highlights strong indirect connections and key nodes that support network cohesion.

Our analysis found a maximum core number of 53, with 54 nodes (4.74% of network) in the maximum K-core, and an average core number of 22.12.

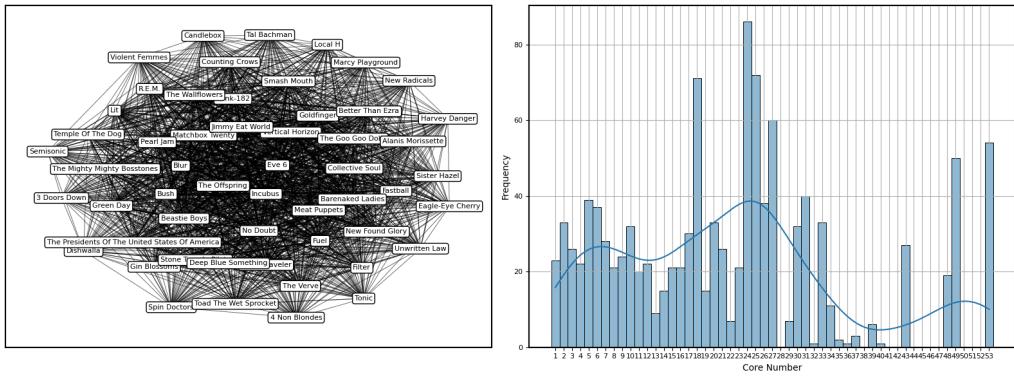


Figure 13: Visualization of the maximum K-core and core distribution across the network

The 53-core reveals a highly cohesive group where each artist connects to at least 53 others. These artists play a crucial role in holding the network together and are ideal candidates for recommendation heuristics due to their strong internal links and influence on connectivity and robustness.

To better understand this structure, in Figure 14 we visualized several K-cores, uncovering an “onion-like” layering that reflects the network’s core-periphery organization. The outer 1-core includes the entire network (1,139 nodes), while the 15-core narrows to 788 nodes, and the 30-core to 280. At the center lies the 53-core, just 54 densely connected artists with 1,431 edges, spotting the most tightly knit and influential part of the network. A well-connected network is revealed by K-core analysis, and this organized decomposition serves as a basis for network-based recommendation algorithms, greatly improving the discovery of music.

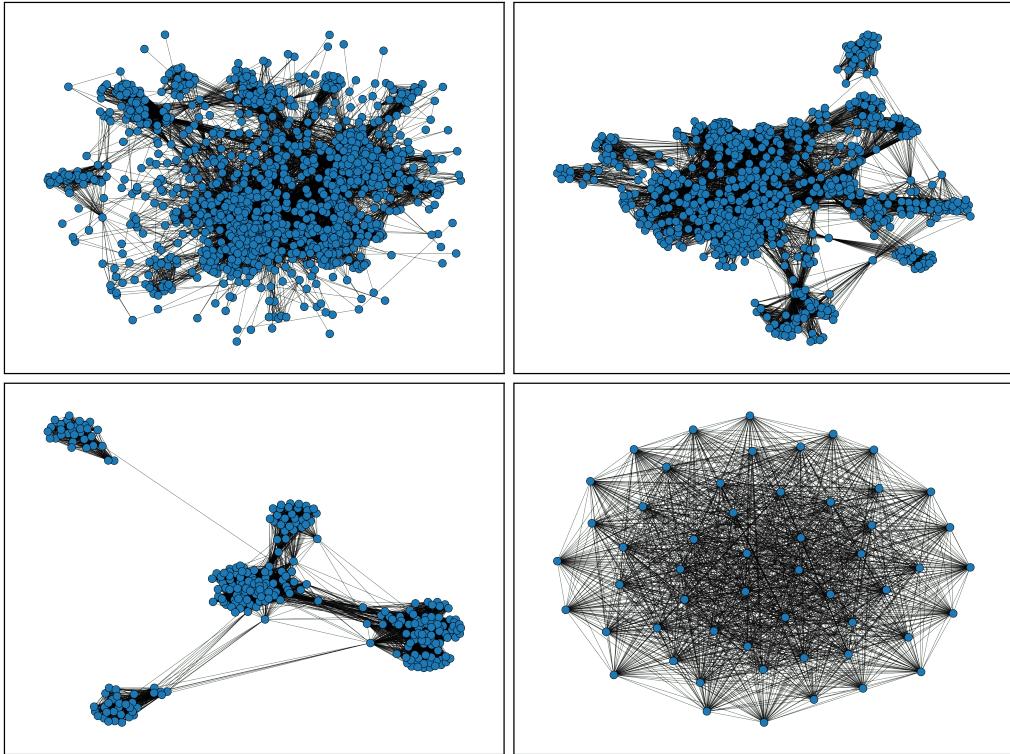


Figure 14: Visualization of several K-cores at different K values

5.6 Periphery Structure

We measured the periphery nodes to better understand the network's structure. These are the most distant nodes, with the longest shortest paths to others, typically sitting at the outer edges. We found 10 such artists, all with low degree centrality, few direct co-occurrences. As shown in Figure 15, they form a sharp peak in the low-degree region, contrasting with the network's broader long-tailed distribution. Most of these artists (9 out of 10) belong to Community 3, focused on worship and Christian music, while the remaining one, Wim Kijne, belongs to the smaller, fragmented Community 6. This suggests that niche or less mainstream genres often populate the structural boundaries of the network.

These nodes offer insight into the network's outer margins, highlighting artists with limited exposure, fewer collaborations, and genre-specific isolation.

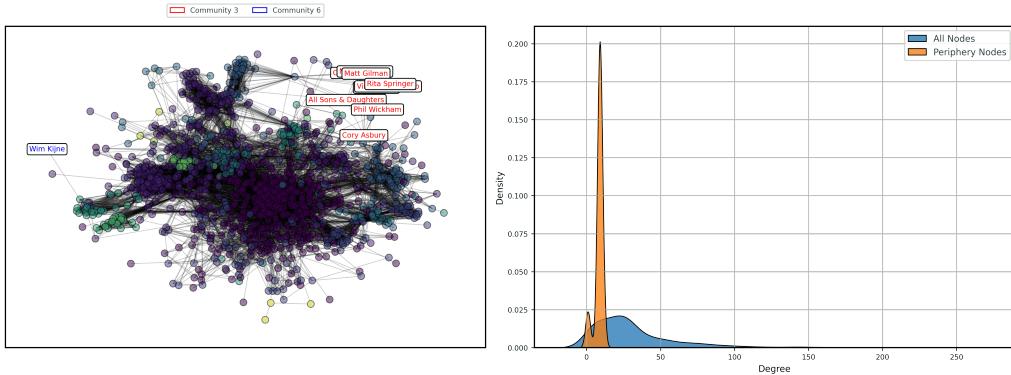


Figure 15: Periphery highlights and degree centrality distribution of periphery nodes vs all nodes

5.7 Structural Equivalence

We introduce structure equivalence analysis to focus on node similarity and apply different measures to understand how it's possible to determine Structural equivalence is a count of the number of common neighbors two nodes have, mathematically

$$n_{ij} = \sum_k A_{ik} A_{kj} \quad (15)$$

Since low-degree nodes are penalized we introduce cosine similarity, defined as

$$\text{cs}_{ij} = \frac{n_{ij}}{\sqrt{d_i d_j}} \quad (16)$$

where n_{ij} is the number of common neighbors between node i and node j , and $\sqrt{d_i d_j}$ is the geometric mean of degrees.

Two alternatives to cosine similarity are Jaccard similarity defined as the number of common neighbors divided by the total number of distinct neighbors of both nodes, namely

$$J_{ij} = \frac{n_{ij}}{d_i + d_j - n_{ij}} \quad (17)$$

and Pearson Correlation

$$r_{ij} = \frac{\sum_k ((A_{ik} - \langle A_i \rangle)) \sum_k ((A_{jk} - \langle A_j \rangle))}{\sum_k (\sqrt{A_{ik} - \langle A_i \rangle^2}) \sum_k (\sqrt{A_{jk} - \langle A_j \rangle^2})} \quad (18)$$

where $\langle A_i \rangle$ is the average of the i -th row and of course $\langle A_j \rangle$ of the j -th.

Figure 16 and Figure 17 compare four structural equivalence metrics across two node groups: 25 random nodes and the top 25 by degree centrality. The charts show normalized common neighbors (top left), cosine similarity (top right), Jaccard (bottom left), and Pearson correlation (bottom right).

The results highlight how each metric captures different aspects of similarity. High shared neighbors don't always mean high cosine or Jaccard scores, due to normalization. Overall, high-degree nodes tend to show stronger structural similarity than random ones.

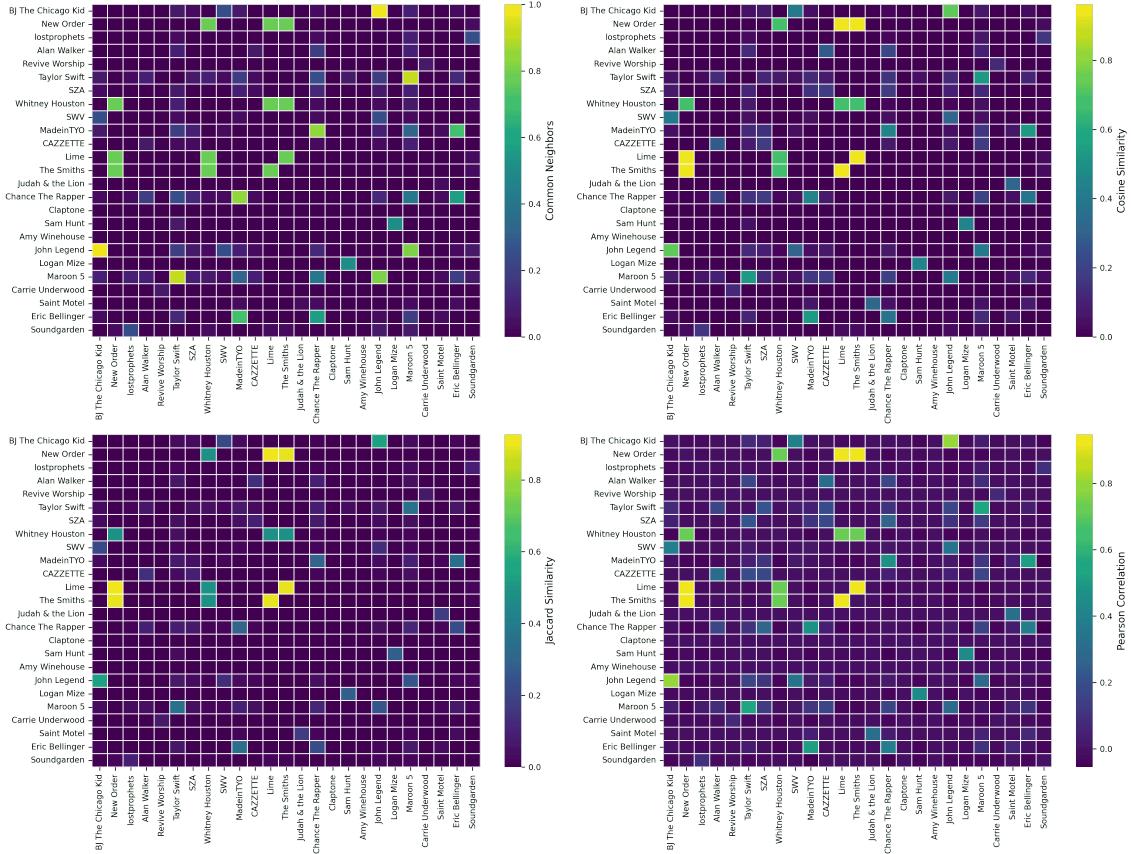


Figure 16: Structural equivalence metrics applied to 25 random nodes from the network

This pattern likely results from high-centrality nodes having more connections, which increases the chances of shared neighbors and stronger similarity scores. While using co-occurrence has its limitations, it still helps explain the observed trends.

Structural equivalence analysis offers insight into node roles and similarities, even without direct connections. Nodes with high structural equivalence often share network positions, making them suitable for predicting future links or recommendations. In music playlists, artists with similar connection patterns may appeal to the same listeners. Based on structural analysis only, we can safely say that some artists may be good candidates for mutual recommendation since their similar network positions suggest they may appeal to the same listeners (see Figure 17).

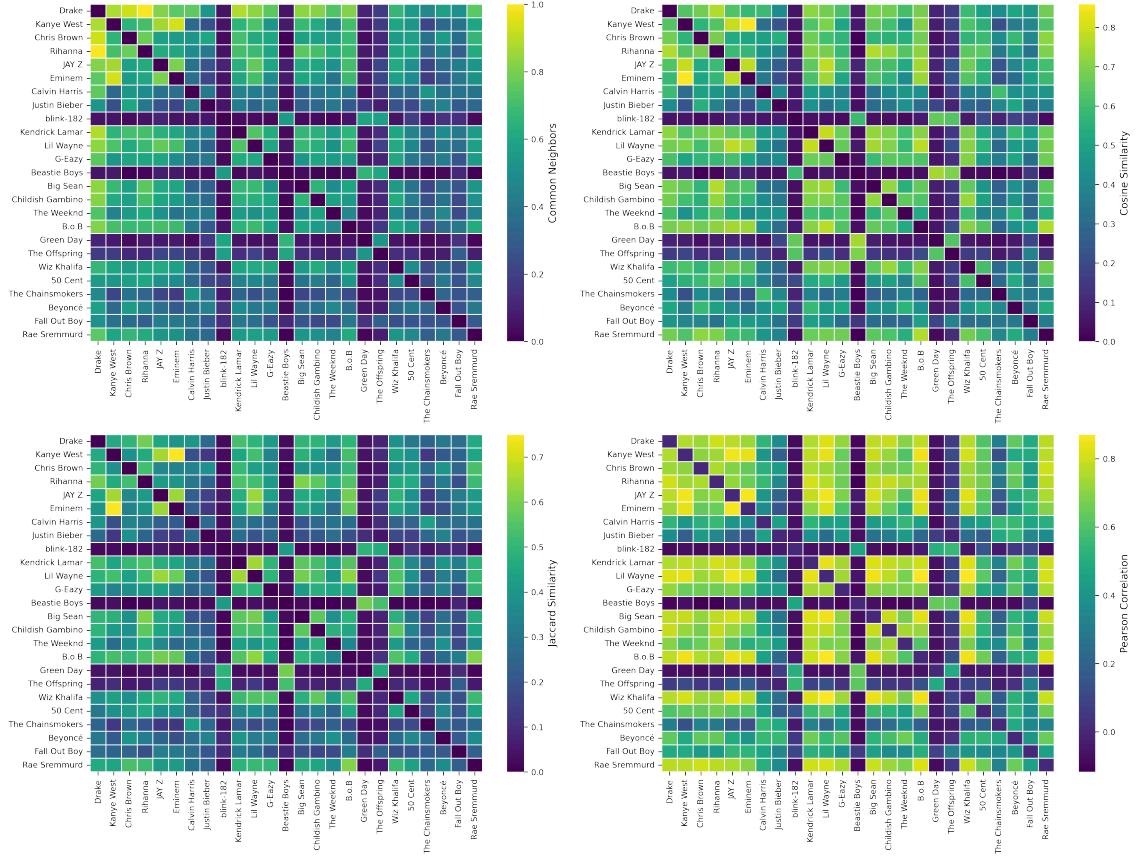


Figure 17: Structural equivalence metrics applied to top 25 artists by degree

5.8 Homophily and Assortative Mixing

A network is assortative if a significant fraction of the edges in the network run between nodes of the same type. We investigate a particular case of assortative mixing which is assortative mixing by degree. We want to understand if high-degree nodes tend to stick together, meaning that popular artists actually saturate playlists.

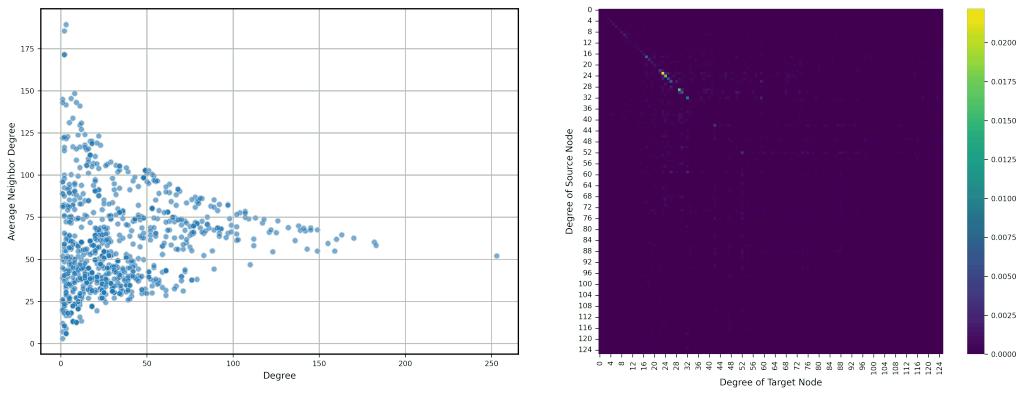


Figure 18: Homophily and assortative mixing by degree

Mixing by degree shows whether nodes tend to connect to others with similar degrees. In Figure 18, the strong diagonal indicates assortative mixing—especially among lower-degree nodes. While the network isn't fully assortative, this pattern stands out, likely due to the degree distribution peaking at lower values. Interestingly, this goes against the typical expectation of disassortative behavior in similar networks.

6 Conclusion

This study examined an artist co-occurrence network, built from Spotify playlist data [4], revealing important insights into user musical tastes and the network's structure. The network, containing 1,139 artists and 17,908 connections, showed features common in real-world social systems. These included a scale-free pattern, meaning some artists had many more connections than others, and a ‘small-world’ effect, where any two artists could be linked by a few steps. Looking at how central each artist was, we found that a small number had much more influence, often acting as key connectors between different parts of the network. K-core decomposition further revealed a layered structure, like an onion, with a tightly connected core group of artists essential for the network's strength. Finding tightly connected groups of artists (cliques) also gave us a direct way to suggest new artists based on what users already like. Conversely, artists at the network's edge were often part of specific, smaller music groups or styles.

Taken together, these findings provide a good starting point for building more effective music recommendation systems. The network's demonstrated small-world property and scale-free nature confirming its applicability for real-world recommendations. Specifically, the identification of cliques allows for direct recommendations within established taste groups, suggesting artists who are closely associated but not yet in a user's playlist. Also, K-core analysis helps determining coherent communities of artists, enabling recommendations that either deepen a user's core taste or expand it within a well-defined musical group of artists. In addition, the analysis of structural equivalence revealed that patterns based on the similarity of network positions may be used to develop heuristics for music recommendation. This suggests it may be possible to identify new recommendation strategies solely through examining the structure of the artist co-occurrence network. Finally, artists exhibiting high betweenness centrality emerge as prime candidates for recommendations aimed at bridging different musical tastes. These “bridge” artists connect distinct groups, making them ideal for introducing users to new but related sounds, thus expanding exploration beyond their current favorite artists.

7 Critique

This analysis is limited by the fact that we are working with a very small subset of the original dataset. The full dataset contains one million playlists, but due to computational constraints, we randomly sampled only 10 slices [4], resulting in 10,000 playlists. This reduced scale may not fully capture the diversity and complexity of user-generated content on the Spotify platform. However, the sampling was performed randomly to preserve general representativeness within practical limits. The difficulty of interpreting co-occurrence relationships has an impact on the entire study. Co-occurrence does not imply a significant or deliberate relationship between artists, even though it may imply similarities, such as similar genres or degrees of popularity. In addition, playlists created by users are arbitrary and might not always correspond with official music classification. As a result, even though the network structure shows intriguing patterns, inferences from it should be considered exploratory rather than final.

Bibliography

- [1] Wikipedia, “Spotify – Wikipedia.” [Online]. Available: <http://it.wikipedia.org/w/index.php?title=Spotify&oldid=144443141>
- [2] H. e. a. Schweiger, “Playlist Coherence and Characteristics,” *EPJ Data Sci.*, vol. 14, no. 1, p. 24, 2025, doi: [10.1140/epjds/s13688-025-00531-3](https://doi.org/10.1140/epjds/s13688-025-00531-3).
- [3] C.-W. Chen, P. Lamere, M. Schedl, and H. Zamani, “RecSys Challenge 2018,” in *Proc. 12th ACM Conf. on Recommender Systems*, 2018, pp. 527–528. doi: [10.1145/3240323.3240342](https://doi.org/10.1145/3240323.3240342).
- [4] Aicrowd, “RecSys Challenge 2018.” [Online]. Available: <https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge/#rules>
- [5] N. F. Y. H. Massimo Rondelli Michele Dinelli, “GitHub Repository.” [Online]. Available: <https://github.com/micheledinelli/sna>
- [6] N. F. Y. H. Massimo Rondelli Michele Dinelli, “GitHub Page.” [Online]. Available: <https://micheledinelli.github.io/sna>
- [7] P. S. Aric Hagberg Dan Schult, “NetworkX.” [Online]. Available: <https://networkx.org/>
- [8] J. D. Hunter, “Matplotlib Graphics Environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007, doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [9] M. Waskom, “Seaborn Visualization Library,” *J. Open Source Software*, vol. 6, no. 60, p. 3021, 2021, doi: [10.21105/joss.03021](https://doi.org/10.21105/joss.03021).
- [10] S. Giallorenzo, “Node Metrics.” [Online]. Available: <https://saveriogiallorenzo.com/teaching/na/slides/L05.pdf>
- [11] T. South, M. Roughan, and L. Mitchell, “Centrality in Spotify Networks,” *J. Complex Netw.*, vol. 8, no. 6, 2020, doi: [10.1093/comnet/cnaa050](https://doi.org/10.1093/comnet/cnaa050).
- [12] L. d. F. e. a. Costa, “Measures in Complex Networks,” *Adv. Phys.*, vol. 56, no. 1, pp. 167–242, 2007, doi: [10.1080/00018730601170527](https://doi.org/10.1080/00018730601170527).
- [13] Wikipedia contributors, “Eigenvector Centrality – Wikipedia.” [Online]. Available: https://en.wikipedia.org/w/index.php?title=Eigenvector_centrality&oldid=1216083063
- [14] L. e. a. Costantini, “Overlapping Centrality Measures,” *Phys. Rev. E*, vol. 105, no. 4, p. 44317, 2022, doi: [10.1103/PhysRevE.105.044317](https://doi.org/10.1103/PhysRevE.105.044317).
- [15] L. C. Freeman, “Centrality in Social Networks,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978, doi: [10.1016/0378-8733\(78\)90021-7](https://doi.org/10.1016/0378-8733(78)90021-7).
- [16] S. Giallorenzo, “Network Metrics.” [Online]. Available: <https://saveriogiallorenzo.com/teaching/na/slides/L06.pdf>
- [17] T. S. Evans and B. Chen, “Closeness and Degree Centrality,” *Commun. Phys.*, vol. 5, no. 1, 2022, doi: [10.1038/s42005-022-00949-5](https://doi.org/10.1038/s42005-022-00949-5).
- [18] Wikipedia contributors, “Small-World Network – Wikipedia.” [Online]. Available: https://en.wikipedia.org/w/index.php?title=Small-world_network&oldid=1284997287

A Full Network

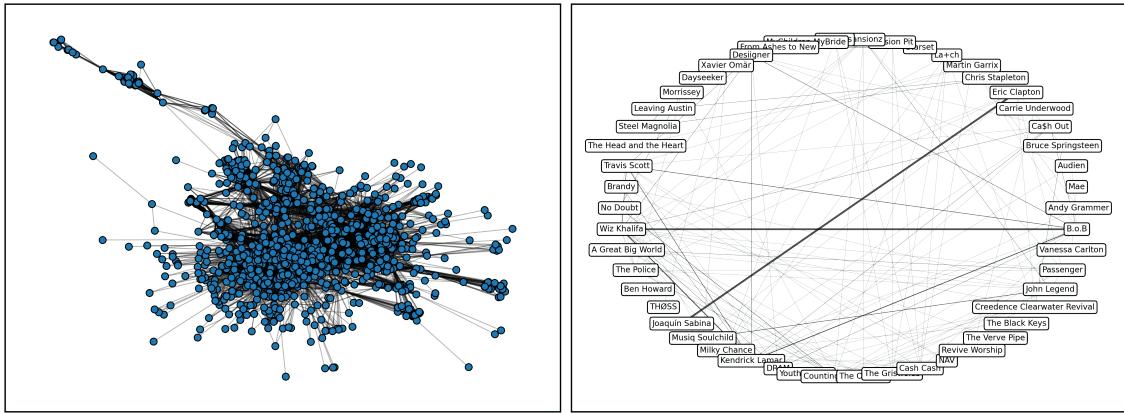


Figure 19: On the right the full network representation. On the left a random subset of nodes to demonstrate what the network looks like. The more darker and thicker the edges, the higher the co-occurrence between nodes.

B Centrality

Closeness centrality can be used to visualize the shortest path from a node u to other nodes $v \neq u$. In Figure 20 its shown the most central node by closeness centrality metric (Drake) and how it's connected with 3 random nodes. Intermediary nodes are highlighted in grey color.

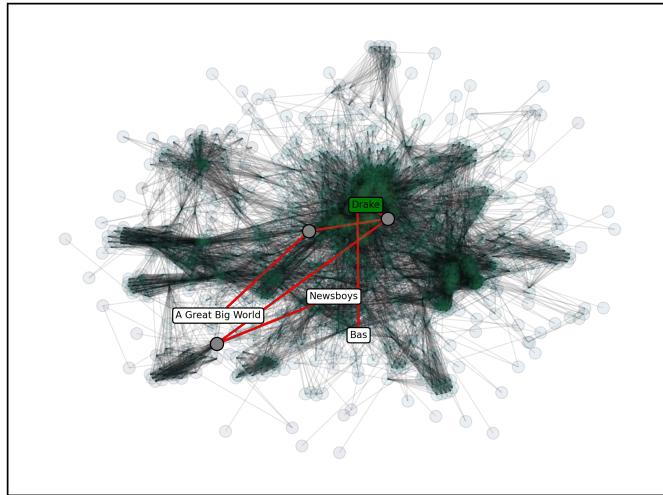


Figure 20: Shortest paths from the node with highest closeness centrality degree (Drake) to 3 random nodes

An example of what happens when removing nodes with high betweenness centrality values is shown in Figure 21 where the top 50 nodes are removed from the network. This results in a less connected network with 14 components.

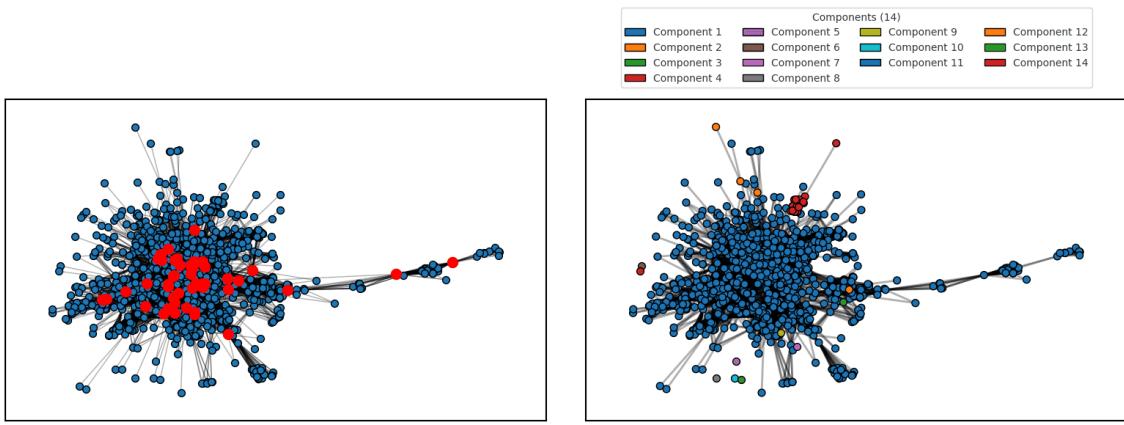


Figure 21: Removal of top nodes by betweenness centrality

C Small World Effect

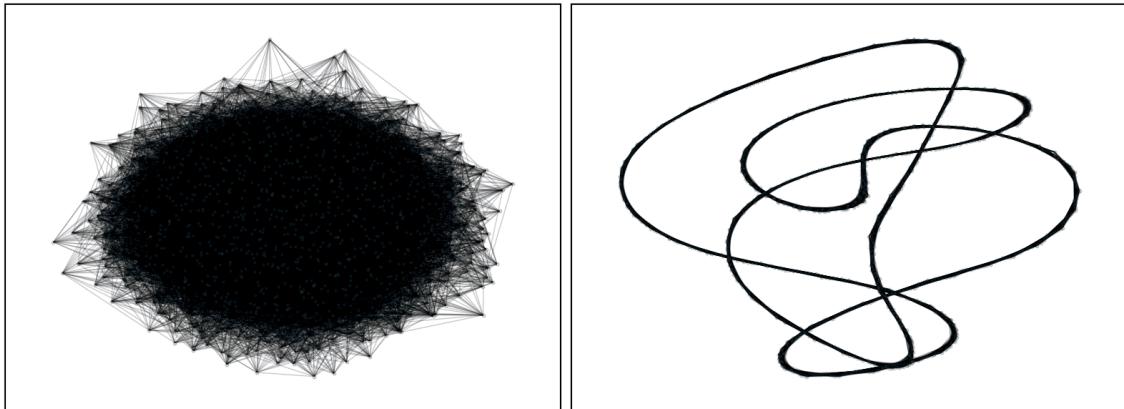


Figure 22: Erdős–Rényi random graph and a regular Graph

The Erdős–Rényi random graph and a regular graph used to calculate ω . Regular graph is generated using `nx.watts_strogatz_graph(n=n, k=k, p=0)`. When the rewiring probability p is set to 0, the Watts-Strogatz model generates a k -regular ring lattice. This means every node in the graph has the same degree, making it a regular graph.