Detecting artist communities: Network analysis and community detection on Spotify artist collaborations

Exploring the music landscape using collaboration data

University of Fribourg – Switzerland
23rd May 2025

Sheila Arnold
sheila.arnold@unifr.ch

Ahmad Hasan Ali Aldabbas
ahmad.aldabbas
@students.unibe.ch

Alina Cosmina Danila
alina-cosmina.danila
@students.unibe.ch

Repository:
https://github.com/SMA-project/SpotifyCommunities

Abstract

This study investigates the dynamics of artist collaborations by building and analysing a social graph based on a Spotify dataset of over 114,000 songs. By applying network science techniques such as centrality analysis and community detection, we identify influential artists, uncover collaboration patterns, and investigate the relationship between artistic attributes (such as genre, popularity, and explicit content) and networking. Our approach includes detailed data pre-processing, graph construction and the implementation of a community detection algorithm (Louvain), including a benchmarking comparison. Furthermore, we analyse the link between community structure and musical genres as well as the tendency of artists to collaborate with similar peers (assortativity and homophily). We also compare the real network structure to synthetic graph models, explore potential future links through link prediction and analyse word co-occurrences in song titles. Finally, we develop a recommendation system for artist collaborations. The results show strong genre specialisation in collaborations and a separation between artists with and without explicit content. Centrality analyses highlight important nodes, and community detection shows clusters of artists that often share a dominant genre. The recommendation system identifies compatible partners based on musical similarity and network position.

1 Introduction

The music industry is a complex ecosystem in which collaborations between artists play an important role. Understanding the underlying network structures can provide insights into the spread of trends, the identification of influencers and the prediction of future partnerships. This project applies network analysis methods to analyse the collaboration network of musicians based on a large Spotify dataset. The aim is to map the structure of this network, identify key players, uncover natural by Maharshi Pandya, contains detailed metadata on over 114,000 music tracks sourced collaboration communities and analyse factors that influence the formation of collaborations. In addition, a model will be developed to recommend potential future collaborations.

2 Data Loading and Preprocessing

The analysis is based on a CSV dataset ('dataset.csv'), which we first loaded and inspected. The dataset from Kaggle, titled "Spotify Tracks Dataset"

from Spotify. It includes information such as track names, artists, genres, popularity scores, and various audio features like danceability, energy, and tempo. This rich metadata enables a wide range of music analysis tasks, from genre-based clustering and popularity trends to network modelling of artist collaborations. It serves as a solid foundation for exploring both the structure of the music industry and listener preferences through data-driven methods.

The pre-processing of the data comprised several steps to ensure the quality for the subsequent network analysis. Firstly, we removed exact duplicates from the data set. Optional, duplicate tracks by the same artist could also be deleted by dropping duplicates based on the columns *'track_name'* and *'artists'*. To normalise the artist names, we developed a function called *parse_artists_final* which cleans up artist names and ensures that they are available as lists of lower-case, cleaned-up strings.

We also converted string representations of lists correctly and names that were separated by semicolons we split into separate artist entries. This splitting was explicitly applied using a function called *split_semicolon_artists*. In the next step, we standardized the genres: we filled missing genre entries with ‘unknown’, we converted all genres to lowercase, and we removed spaces. Finally, we removed irrelevant columns that were classified as less important for the planned network analysis (such as *duration_ms, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, time_signature*). We displayed the remaining columns for checking.

3 Synthetic Model Graphs

To understand how the real artist network structure compares to theoretical network models, we generated and analysed three types of synthetic graphs: Barabási–Albert (BA), Watts–Strogatz (WS), and Erdős–Rényi (ER). We used the Top 1000 nodes subgraph (G_top_1000) as the real graph for comparison (G_real). These synthetic graphs were generated to match the number of nodes and average degree of the real subgraph.

We compared the degree distributions of the real graph and the three model graphs by plotting them on a log-log scale. This visualisation helps identify patterns like power-law behaviour (characteristic of networks with hubs). We also printed key structural statistics for each graph, including the number of nodes and edges, average clustering coefficient (measuring local connectivity), and average shortest path length (measuring overall network efficiency).

This comparison allows us to see how well the synthetic models replicate the real artist collaboration network's connectivity patterns, clustering, and navigability. However, we found that the synthetic graphs failed to capture the complexity of real-world networks, highlighting the limitations of these models in approximating real collaboration structures.

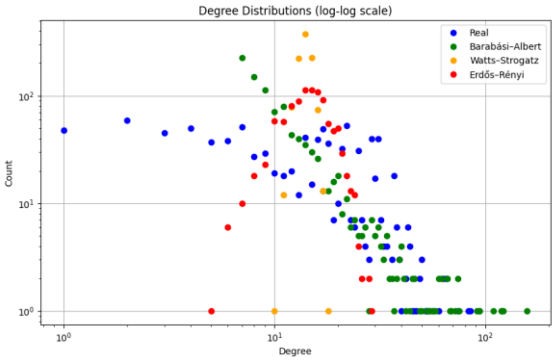

Figure 1: Degree Distributions

4 Graph Construction

We created an undirected graph (*nx.Graph*) for the network analysis, which represents the artists' collaboration network. Each node in the graph represents a single artist. An edge between two nodes indicates that these two artists have collaborated at least once, i.e. are listed together in the artist's field of a song. The weight of the edge indicates how often this collaboration has taken place: The first joint song creates an edge with a weight of 1. For each subsequent collaboration between the same artists, this weight is increased accordingly. We constructed the graph by traversing each line of the pre-processed DataFrame. For each song, we extracted the artists involved. If more than one artist was involved, we inserted edges between all possible artist pairs of that song or we strengthened existing edges accordingly. Artists who had no collaborations with others in the dataset we also included as individual nodes in the graph. Once we completed the construction, the graph comprised a certain number of artists (nodes) and collaborations (edges). To illustrate this, we displayed some edges with their respective collaboration weights as examples.

5 Network Exploration

The exploration of the network included several analyses of the structure and attributes of the collaboration graph. The aim was to identify key artists, visualise structural characteristics and gain additional insights at song level through clustering.

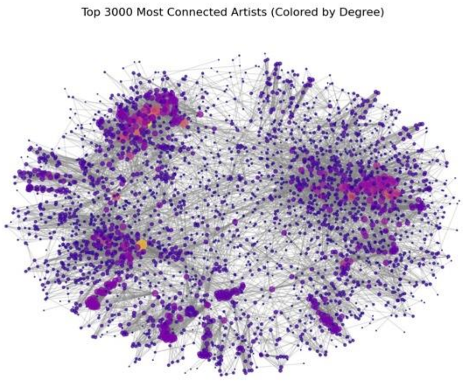

Figure 2: Top 3000 connected Artists

5.1 Subgraph and Degree Centrality

For better visualisation and analysis, we first created a sub-graph containing the 3000 most strongly networked artists. A key metric in this context was degree centrality, which measures how many direct collaborations an artist has. The calculation of this metric for all nodes revealed that most artists have very few connections, while a few are networked as hubs with numerous other artists. Such outliers with high degree centrality can be interpreted as key players or connection points in the network. The distribution of degree centrality indicates a sparsely connected graph with low density and a low average degree of connection, suggesting many loosely connected or isolated components.

For a more in-depth analysis, we also calculated the Betweenness Centrality and PageRank metrics on a subgraph comprising the 1000 nodes with the highest degree. Betweenness Centrality measures how often a node is on the shortest path between others, while PageRank assesses the ‘influence strength’ of an artist based on their connections to other important nodes. We summarised the results of these calculations together with the degree centrality in a DataFrame. We displayed the ten most influential artists according to PageRank. A subsequent visualisation showed the nodes in colour and size according to their PageRank value on this Top 1000 node subgraph. Artists with a high PageRank are particularly influential as they are networked with other important artists. Artists with high betweenness, on the other hand, act as bridges in the network, while the degree merely reflects the number of collaborations.

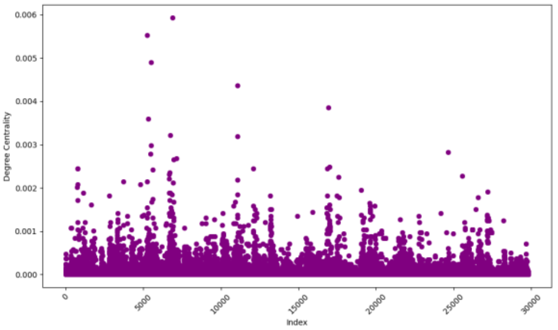

Figure 3: Graph Centrality

5.2 K-Means

The K-Means algorithm is a widely used method for cluster analysis that is used to divide data points into groups (so-called clusters) based on certain characteristics. The aim is to group data points that are as similar as possible within a cluster and as different as possible between clusters. The number of clusters (K) to be found is defined in advance.

In addition to analysing the network topology, we carried out a cluster analysis based on song characteristics. For this purpose, we divided the songs into three groups based on the characteristics of popularity and danceability. Before applying the K-Means algorithm, we scaled the characteristics accordingly. We labelled the resulting clusters as ‘Mainstream Dance Hits’, ‘Underrated Grooves’ and ‘Low-Key Tracks’. A subsequent scatterplot visualisation showed the three clearly distinguishable groups. This suggests that popularity and danceability are suitable characteristics for identifying different types of songs within the data set.

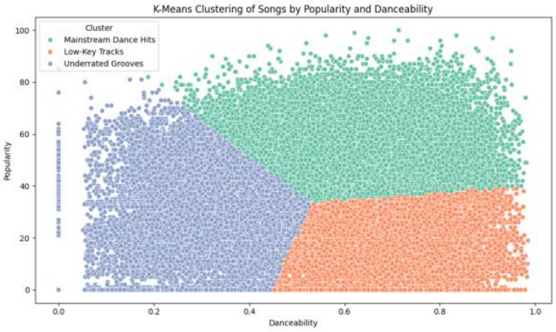

Figure 4: K-Means

6 Network Analytics

The core of the analysis focussed on community detection and the investigation of network-related characteristics in connection with attributes such as genre, popularity and explicit content. We aimed to find out how artists are grouped in the network, to what extent they have similarities and how these patterns can potentially be used for recommendations.

6.1 Louvain Implementation

The identification of communities within a network is a central goal of network analysis - especially for complex structures. The Louvain algorithm has established itself as a particularly efficient approach to community detection. It is based on the optimisation of modularity and makes it possible to identify groups of similar nodes. The algorithm works iteratively in two phases: First, nodes are moved to neighbouring communities in order to maximise modularity. These communities are then combined into super nodes, creating a new, simplified graph. This process is repeated until no further improvement is possible. The quality of the recognised structures can be evaluated using metrics such as modularity or Normalised Mutual Information (NMI).

To detect communities in the network, we implemented the Louvain algorithm, a proven method for maximising modularity that first moves nodes locally using an iterative process and then aggregates the graph. The implementation follows the classic two-phase approach. We executed a user-defined variant of the algorithm (*louvain_with_aggregation*) on a subgraph with the 1000 best connected artists (*G_top_1000*). The results of this computationally intensive step were saved and then loaded for subsequent analysis.

6.1.1 Benchmarking Louvain Implementation

We conducted a benchmark comparison between our implementation of the Louvain variant and the standard implementation from the python-louvain library across different graph sizes. The results, loaded from a CSV file previously obtained (*louvain_benchmark_results.csv*), showed that our implementation achieves comparable modularity scores to the standard library, indicating similar community quality. However, our implementation ran significantly slower, highlighting the efficiency advantages of optimized library implementations. Visualisation of the runtime and modularity comparisons clearly illustrates this difference. This exercise was valuable for understanding the internal workings of the Louvain algorithm and the reasons behind the performance differences compared to optimized library versions.

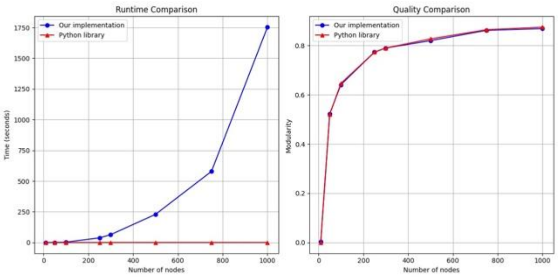

Figure 5: Benchmarking Louvain Implementation

6.1.2 Community Detection and Genre Analysis

To assess the quality of the recognised communities from the Louvain application on the Top 1000 node subgraph (*G_top_1000*), we assigned the community IDs to the nodes in the graph as an attribute. To analyse the extent to which these groups are characterised by musical genres, we created a mapping of artists to genres from the original dataset and grouped artists according to their community affiliations. The frequency of genres within each group was counted.

We identified the largest communities by number of artists and analysed their genre distributions. The analysis of the top 5 largest communities revealed varying levels of genre diversity. For example, Community 5 (Size: 63 artists) showed a strong presence of Spanish music (93.7%), alongside techno. Community 2 with the biggest size (Size: 163 artists) had a mix of genres of progressive-house: 28 (17.2%), trance: 20 (12.3%), reggae: 13 (8.0%), and others.

Furthermore, we printed the most genre-concentrated communities, filtering for those where a single genre accounts for over 90% of artists. Three communities stood out in particular due to their genre homogeneity: Community 5 (Size: 63 artists) with a 93.7% focus on spanish songs, Community 21 (Size: 32 artists) with 100.0% show-tunes concentration, and Community 24 (Size: 30 artists) with also 100.0% spanish music artists. These strongly focussed groups suggest specific musical niches or authentic genre purists. Other communities, like Community 14 (Size: 52 artists) with a strong Funk dominance (78.8%), also showed high concentration. These results highlight both the diversity of musical collaborations and the existence of genre-specific artist networks within the broader music ecosystem.

6.2 Assortativity and Homophily

Assortativity measures the tendency for nodes to connect with similar others, ranging from -1 (perfect disassortment) to 1 (perfect assortment), while homophily specifically quantifies the proportion of edges linking nodes with identical attributes.

Understanding these patterns helps identify natural collaboration communities, predict potential partnerships, and recognize invisible boundaries in the music collaboration network that might influence creative output and audience reach. The high homophily values particularly demonstrate how similarity breeds connection in this creative network, with artists predominantly teaming up with others who share their genre, popularity level, and content style characteristics.

We analysed whether artists with certain attributes - in particular genre, popularity and explicit content - prefer to collaborate with similar artists. This analysis was conducted on a subgraph containing the Top 10000 nodes based on degree centrality (*G_top_10000*). The explicit attribute was converted to a binary format (1/0) for this analysis. The genre assortativity coefficient of 0.675 showed a clear correlation, indicating that artists tend to collaborate within the same genre. This tendency was confirmed by a genre homophily ratio of 0.685, which means that around two thirds of all collaborations take place between artists of the same genre. The popularity assortativity of 0.614 indicated a moderately-high preference for equally well-known artists. In contrast, the analysis of explicit content showed a lower but still meaningful assortativity of 0.545, but an exceptionally high homophily of 0.921. This indicates that artists with and without explicit content act very separately from each other. An edge composition analysis showed that only 5.6% of edges connected explicit artists (Explicit-Explicit), while 86.5% involved connections between non-explicit artists (Clean-Clean). Only 7.9% of edges crossed the boundary between these two groups (Mixed edges). Overall, these metrics show that similarity on different levels - musically, in terms of content and in terms of reach - is a key driver for collaborations.

6.3 Social recommendation systems

Social recommendation systems have become essential tools in the music industry for identifying potential collaborations between artists. These systems analyze complex networks of relationships, musical attributes, and behavioral patterns to suggest partnerships that are both creatively compatible and strategically valuable. By examining the underlying social graph of artists and their connections, these systems can uncover hidden opportunities that might not be immediately apparent through traditional A&R (Artist & Repertoire) methods. The power of social recommendations lies in their ability to combine multiple data dimensions - from genre similarities to network centrality metrics - providing a more holistic view of potential collaborations than any single factor could offer.

Our implementation begins by processing the raw artist and track data to create comprehensive genre profiles for each artist. The system then examines the social graph structure, calculating both local metrics like shared neighbors and global network properties including degree centrality and betweenness centrality. These diverse factors are combined into a weighted score that balances musical similarity with network position, ensuring recommendations consider both creative compatibility and social connectivity. The implementation includes special handling for cross-community recommendations, which can surface innovative pairings that bridge different musical scenes or genres. The system was initialised using the Top 1000 node subgraph (*G_top_1000*).

Based on the network and attribute analyses, we developed a social recommendation system that suggests potential collaboration partners for artists. The system combines genre profiles, local network metrics (such as shared neighbours) and global centrality measures (e.g. Degree and Betweenness Centrality). We selected J Balvin, one of the top artists in our network, as an example. We carried out the implementation using a class called *ArtistRecommender*, which prepares the data, calculates genre vectors, quantifies similarities and generates recommendations based on this. The suggestion mechanism is based on a composite score that combines genre similarity, overlap of collaboration partners and network position.

The recommendations generated for J Balvin included artists such as Nicky Jam, Tainy, Natti Natasha, Daddy Yankee and Ozuna. These recommendations highlight strong connections between J Balvin and other prominent Latin artists, particularly in the reggaeton and Latin trap scenes. For example, Nicky Jam shares similar genres like Latin and electronic and has significant overlap in collaborators like Bad Bunny, Ozuna, and Daddy Yankee. The high number of common neighbors suggests frequent collaborations or shared industry circles. Natti Natasha, Daddy Yankee, and Ozuna all share strong genre alignment and extensive shared networks within the Latin urban music scene. The centrality and betweenness metrics further validate their influential positions. The score values of the suggested artists were close to each other, clustering between 0.37-0.39, indicating a group of highly compatible collaboration partners rather than one standout candidate.

7 Link Prediction

We explored the potential for predicting future collaborations by applying link prediction algorithms to the network. Using the Top 1000 nodes subgraph (*G_sub*), we sampled a subset of non-connected pairs (pairs of artists without an edge) to predict on. We computed link prediction scores for these non-edges using two common methods: the Jaccard Coefficient and the Adamic-Adar Index. The Jaccard Coefficient measures the similarity between the sets of neighbors of two nodes, while the Adamic-Adar Index sums the inverse logarithm of the degrees of common neighbors. We displayed the top 10 predicted links according to each method, showing potential future collaborations based on the current network structure.

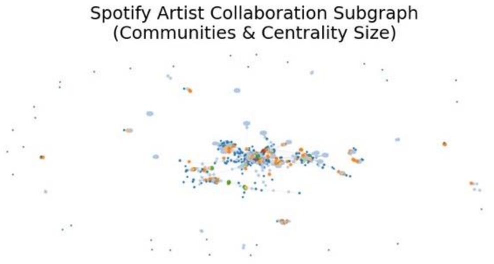
Spotify Artist Collaboration Subgraph
(Communities & Centrality Size)

Figure 6: Spotify Artist Collaborations Subgraph

8 Co-occurrences

To gain insight into recurring themes or styles within the music dataset beyond network structure, we analysed the co-occurrence of words in song titles. We extracted and counted all words from track names and identified the 50 most frequent words. We then counted how often each pair of these top 50 words appeared together in the same song title, creating a co-occurrence matrix. This co-occurrence data was used to build a graph where nodes represent the top words and edges indicate their frequency of co-occurrence in titles, with edge weights reflecting this frequency. Finally, we visualized this co-occurrence graph to illustrate the relationships between the most common words in the song titles.

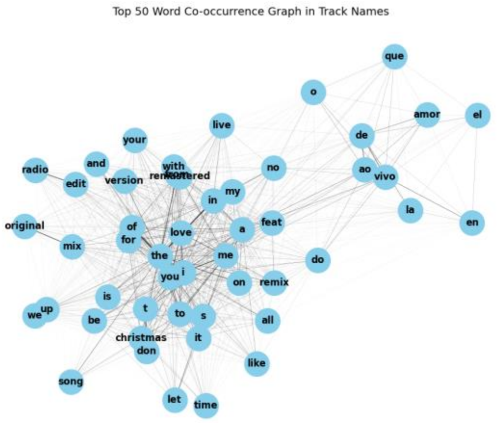

Figure 7: Top 50 Word Co-occurences Graph in Track Names

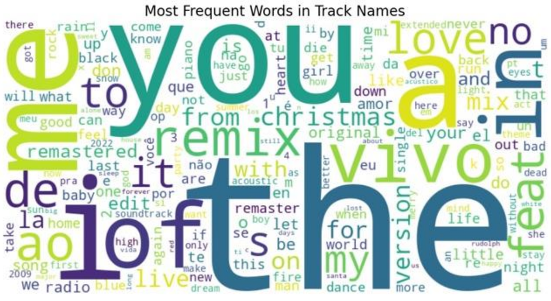

Figure 8: Most Frequent Words in Track Names

9 Conclusion and further steps

The analysis of the Spotify artist collaboration network using network science methods provided valuable insights into the structure and dynamics of musical collaboration. Based on the pre-processed data, we successfully constructed a graph to map the relationships between artists. We used various centrality analyses to identify key nodes that are particularly influential within the network. After benchmarking between our implementation of the Louvain method and the one from the python library up to 1000 nodes, because of the increasingly long time it took, we uncovered collaboration communities and analysed their genre composition using our Louvain algorithm on the top 1000 subgraph by degree. The analysis of assortativity and homophily, performed on a Top 10.000 subgraph, revealed a strong tendency to collaborate within the same genre, as well as a clear distinction between artists with and without explicit content. We also compared the real network to synthetic models, explored link prediction, and analysed word co-occurrences. Based on these findings, we developed a social referral system that suggests potential collaborators based on network position, genre similarity, and other relevant metrics.

There are several approaches for further analysis. For example, a model for predicting future collaborations could be developed that incorporates network features and artist attributes. Studying genre evolution over time would allow us to better understand the mixing and evolution of musical styles through collaborations. Similarly, it would be possible to analyse whether an artist’s network position, combined with audio features, can help predict a song’s popularity. An in-depth analysis of individual communities could also provide further insights into specific group structures, for example, by incorporating additional artist attributes or external data sources. Finally, the recommendation system we developed could be further optimised, for example, by a refined calculation of the composite score or by integrating additional factors for even more personalised recommendations.

Task distribution

In our project, we clearly divided the tasks in order to work together efficiently. All team members were jointly responsible for researching and selecting the data set. We collected ideas and discussed how we could best realise the project. Sheila took responsibility for writing the report and creating the presentation. Alina and Ahmad concentrated on the technical implementation and wrote the code following the split found in the table below.

Chapter implemented	Person responsible
Data loading + Preprocessing	Ahmad
Synthetic Model Graphs	Ahmad
Network exploration	Alina + Ahmad
Graph Centralities	Alina
Louvain Algorithm + Benchmarking Community detection Assortativity & Homophily Social Recommendation Systems	Alina
Link Prediction	Ahmad
Co-occurrences	Ahmad

This division of tasks allowed us to utilise our individual strengths and work on the project in a structured way.

Dataset: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset?resource=download