Unveiling Spotify's Collaboration Network

A Network Science Analysis of Artists' Collaborations

Antonio Castriotta
a.castriotta@studenti.unipi.it
Student ID: 685223

Abstract

In this project, we construct and analyze a collaboration network of Spotify artists, where nodes represent musicians and edges correspond to actual collaborations. The dataset was built by collecting information through the Spotify API, retrieving artists' attributes such as genres, popularity, and followers, and extracting collaboration links between them. We first characterize the network by studying its structural properties—degree distribution, connected components, clustering coefficient, assortativity, path lengths, and centrality measures—and compare them with synthetic benchmarks (Erdős–Rényi, Barabási–Albert, Watts–Strogatz, and Configuration Model). We then explore its modular organization by applying different community detection algorithms and evaluating their performance both quantitatively and semantically with respect to musical genres. Additionally, we tackle the link prediction task, testing unsupervised similarity-based methods as well as a supervised approach based on Node2Vec embeddings and machine learning classifiers. Finally, we address the open research question: *do collaborations predominantly occur within the same genre, or do they also act as bridges across different genres?* By combining topological analysis with genre attributes, we provide insights into how community structures and cross-genre connections shape the dynamics of musical collaborations on Spotify.

ACM Reference Format:
Antonio Castriotta. 2025. Unveiling Spotify's Collaboration Network: A Network Science Analysis of Artists' Collaborations. In *Social Network Analysis '25*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In recent years, Spotify has become the largest music streaming platform worldwide, hosting millions of songs and a vast number of artists across different genres. Beyond offering music consumption, Spotify provides access to valuable information through its API, enabling researchers to investigate the structure and dynamics of the music industry from a network science perspective. Among the many possible relations between artists, collaborations represent a particularly interesting dimension: they shape creative processes, influence audience reach, and often act as bridges between musical cultures and genres. By modeling artists as nodes and collaborations as edges, it is possible to unveil the structural properties of the collaboration landscape and to better understand how music production is organized at scale. The objective of this project is to construct and analyze a collaboration network of Spotify artists, collected directly through the Spotify API. After building the dataset, we first characterize its main structural properties and compare them with synthetic benchmark models. We then perform advanced analytical tasks, including community detection and link prediction, and finally address an open research question: whether collaborations predominantly occur within the same genre or whether they also act as bridges across different genres. This study aims to shed light on the interplay between network topology and genre-based organization in the context of contemporary music collaborations.

SNA '25, University of Pisa, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 Data Collection

The network analyzed in this project was constructed starting from the list of the top 3,000 most streamed artists published by `Kworb.net`, which was used as a seed set. From this initial set, the graph was iteratively expanded by retrieving collaboration information through the Spotify Web API, accessed via the Python library `Spotipy`. In each iteration, collaborators of the already discovered artists were added to the network until no new nodes satisfying the selection criteria were found. For every artist we collected a number of descriptive attributes, including:

- the main genre and the complete list of associated genres,
- the popularity score assigned by Spotify,
- the total number of followers,
- the unique Spotify identifier.

Edges represent actual collaborations: an undirected link is created between two artists whenever they appear together in at least one track. Each edge is weighted by the number of shared collaborations, so that stronger ties correspond to repeated joint appearances. To reduce noise, we applied a popularity threshold, discarding artists with a score below 20. We also removed self-loops and checked for duplicate entries. The resulting network is an undirected weighted graph with **11,083 nodes** and **14,497 edges**. The edge weights are highly skewed: the minimum and median are equal to 1 (indicating that most collaborations occur only once), the average is 2.41, while the maximum reaches 222, highlighting a few pairs of artists with very frequent joint work. No nodes are missing the main genre attribute, ensuring that semantic information is available for all vertices. This graph forms the basis for all subsequent analyses of network structure, community detection, link prediction, and the open research question.

3 Network Characterization

We first investigated the structural properties of the extracted collaboration network by computing a set of standard network statistics, including number of nodes and edges, density, degree/strength distribution, connected components, clustering coefficient, assortativity, and centrality measures. To assess how these properties compare to classical random graph models, we generated synthetic networks with approximately the same number of nodes and edges as our collaboration graph. Specifically, we considered:

- an Erdös–Rényi (ER) random graph, generated using $p = \frac{2E}{N(N-1)}$ as probability of link between two nodes,
- a Barabási–Albert (BA) preferential attachment model, with parameter m chosen to approximate the average degree of the real network,
- a Watts–Strogatz (WS) small-world model, with k set close to the average degree and rewiring probability $p = 0.2$,
- a Configuration Model (CM) explicitly built to reproduce the empirical degree distribution of the collaboration graph.

By comparing the results obtained on the real network with those of the synthetic models, we can highlight which structural features are specific to the music collaboration context (e.g., heavy-tailed distributions, high clustering, disassortativity) and which are instead reproduced by generic generative processes.

3.1 Degree and strength distribution analysis

Model	Number of nodes	Number of Links	Average Degree/Strength
AN (Unweighted)	11083	14497	2.616
AN (Weighted)	11083	14497	15.73
ER (Unweighted)	11083	11332	2.045
ER (Weighted)	11083	11332	2.045
BA (Unweighted)	11083	66462	11.994
BA (Weighted)	11083	66462	12.84
WS (Unweighted)	11083	66498	12.000
WS (Weighted)	11083	66498	12.91
CM (Unweighted)	11083	14441	2.606
CM (Weighted)	11083	14441	14.92

Table 1: Links and average degree/strength comparison between the collaboration network (AN) and synthetic networks, unweighted vs weighted

Model	Highest Degree/Strength	Lowest Degree/Strength	Self Loops
AN (Unweighted)	110	0	0
AN (Weighted)	972	0	0
ER (Unweighted)	9	0	0
ER (Weighted)	9	0	0
BA (Unweighted)	432	6	0
BA (Weighted)	489	6	0
WS (Unweighted)	18	7	0
WS (Weighted)	21	7	0
CM (Unweighted)	106	0	5
CM (Weighted)	927	0	5

Table 2: Highest/Lowest degree or strength comparison, unweighted vs weighted

From Table 1 we observe that the weighted version of the collaboration network exhibits a much higher average connectivity (average strength $\langle s \rangle = 15.7$), despite having the same number of nodes and links as the unweighted counterpart. This indicates that many collaborations occur repeatedly among the same pairs of artists. The Configuration Model remains the closest to the empirical network in terms of raw degree distribution, while the weighted analysis highlights the intensity of interactions, which none of the synthetic models are able to reproduce.

Focusing on extreme values (Table 2), the weighted network shows nodes with very large strength (up to 972), meaning that certain artists collaborate much more intensively than others. In contrast, the unweighted degree distribution only emphasizes the number of distinct collaborators (maximum 110). This distinction highlights the importance of considering weights: hubs by degree are not always the same as hubs by strength, and the weighted analysis reveals "super-collaborators" that were less evident in the unweighted graph.

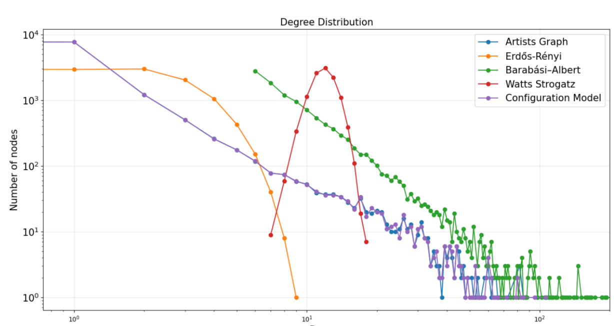


Figure 1: Degree distribution comparison between the real collaboration network (AN) and synthetic models (unweighted).

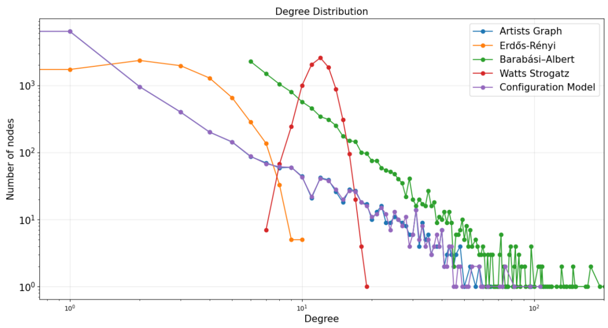


Figure 2: Strength distribution (weighted degree) of the collaboration network (AN).

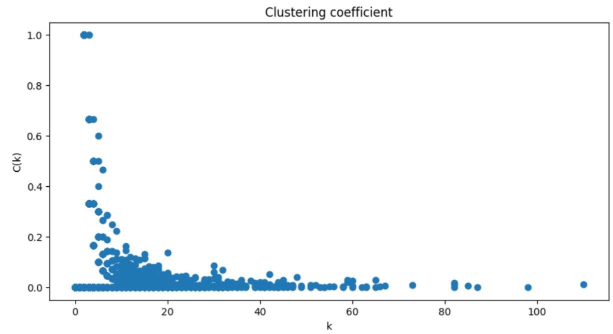


Figure 3: Relationship between degree k and clustering coefficient $C(k)$ in the unweighted collaboration network (AN).

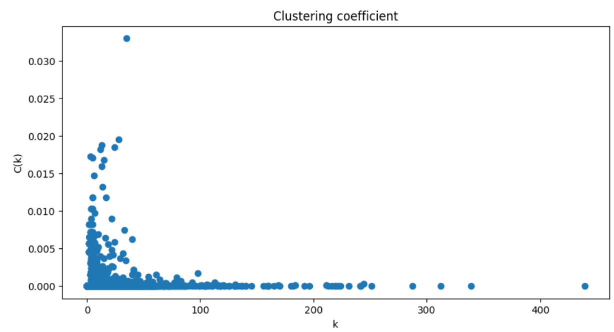


Figure 4: Relationship between strength s and weighted clustering coefficient $C(s)$ in the weighted collaboration network (AN).

Figure 1 shows that the degree distribution of the real collaboration network follows a heavy-tailed pattern, with most artists having very few distinct collaborators and a small fraction acting as hubs. When considering weights, Figure 2 highlights that the distribution of strengths is even more skewed: a very small set of artists accumulates the majority of collaborative intensity, while most nodes maintain only a limited number of repeated interactions. This confirms that degree alone underestimates the role of "super-collaborators," which emerge clearly when the frequency of collaborations is taken into account.

3.2 Connected components analysis

Both weighted and unweighted analyses confirm that the network does not collapse into a single giant component. In the weighted network, the giant component still covers the vast majority of nodes (around 87% of the total), while numerous smaller components remain. This fragmentation is consistent with the relatively low average degree compared to the theoretical connectivity threshold $\ln(N) \approx 9.31$. Weights do not substantially affect the size of components, since connectivity only depends on the existence of edges. However, within the giant component, weighted paths highlight stronger cores of artists connected through repeated collaborations, while peripheral artists remain loosely linked.

3.3 Clustering coefficient and assortativity

The local clustering coefficient was computed both in its unweighted and weighted definition (Barrat formulation). Results are summarized in Table 3.

Model	Density	Average CC	#Triangles	Assortativity
AN (Unweighted)	$2.36 \cdot 10^{-4}$	0.0176	2685	-0.3525
AN (Weighted)	$2.36 \cdot 10^{-4}$	0.0412	5129	-0.2870
ER (Unweighted)	$1.85 \cdot 10^{-4}$	0.00015	3	0.0029
ER (Weighted)	$1.85 \cdot 10^{-4}$	0.00015	3	0.0029
BA (Unweighted)	$1.08 \cdot 10^{-3}$	0.00766	11697	-0.0269
BA (Weighted)	$1.08 \cdot 10^{-3}$	0.00894	12841	-0.0215
WS (Unweighted)	$1.08 \cdot 10^{-3}$	0.3497	253059	-0.0094
WS (Weighted)	$1.08 \cdot 10^{-3}$	0.3621	260114	-0.0082
CM (Unweighted)	$2.35 \cdot 10^{-4}$	0.00191	1374	-0.0061
CM (Weighted)	$2.35 \cdot 10^{-4}$	0.00387	2915	-0.0048

Table 3: Density, clustering coefficient and assortativity comparison between models (unweighted vs weighted)

The weighted analysis yields higher clustering and a larger number of triangles compared to the unweighted case. This shows that repeated collaborations tend to concentrate within tightly connected groups, reinforcing local cohesion. In terms of assortativity, both versions remain disassortative, but the weighted network is slightly less so (-0.287 vs -0.3525), indicating that when considering the intensity of links, highly collaborative artists connect a bit more frequently with peers of similar strength than what is suggested by the purely unweighted topology.

3.4 Path analysis

Computing the exact diameter and average shortest path length of a large-scale network is computationally demanding. For this reason, we relied on approximation methods applied to the largest connected component of each network. Following [**?**], we estimated the diameter using the 2-sweep algorithm available in NetworkX for undirected graphs, which selects a random node, finds its farthest node, and returns the eccentricity of the latter as a lower bound for the diameter. To approximate the average shortest path length, we adopted a sampling strategy: 200 random pairs of nodes were selected and their shortest paths computed, and the procedure was repeated 100 times. The mean and standard deviation across the obtained samples provide the final estimate.

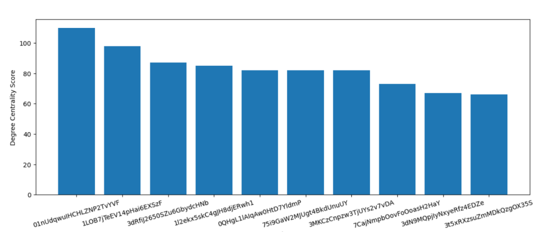
Model	2-sweep diameter	Appr. average shortest path
AN	19	6.9153 ± 0.133
ER	11	5.211 ± 0.052
BA	7	3.913 ± 0.027
WS	10	6.021 ± 0.045
CM	15	5.478 ± 0.061

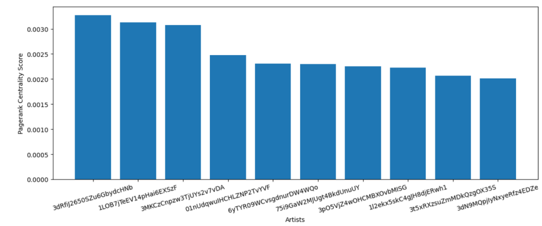
Table 4: Approximated diameter and average shortest path length for the collaboration network and synthetic models

The collaboration network (AN) displays the largest estimated diameter as well as the highest average shortest path length. This result reflects the sparse and fragmented nature of real-world collaboration structures: although artists are often linked through chains of collaborations, these paths tend to be relatively long compared to those observed in synthetic graphs. BA and ER produce significantly smaller values, with paths of only 3–5 steps on average, while the WS model comes closer to AN but still underestimates the diameter. The CM provides intermediate results, consistent with its ability to preserve the empirical degree distribution but not the global structure of the network.

3.5 Centrality analysis

We also investigated the most relevant artists according to different centrality measures. We considered degree centrality (number of direct collaborations of an artist), PageRank centrality (importance derived from being connected to other relevant nodes), harmonic and closeness centrality (identifying nodes that can reach the rest of the network through relatively short paths), and eigenvector centrality (importance derived from being connected to other highly connected artists).





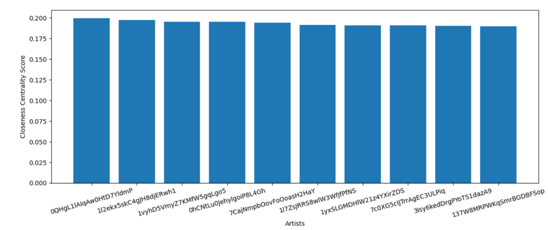


Figure 5: Top 10 nodes for degree, closeness and PageRank centrality

Looking at the top-ranked nodes for each centrality measure, we observe that closeness and harmonic centrality tend to overlap, consistently highlighting artists that are well embedded in the largest connected component and can quickly reach other musicians through collaboration chains. Degree and eigenvector centrality also show similar patterns, identifying the main hubs of the network, i.e., artists with the highest number of collaborations and strong ties to other well-connected musicians. PageRank centrality, on the other hand, produces partially different results: it tends to emphasize artists that, while not having the maximum number of collaborations, are strategically linked to central hubs and thus gain importance from the structure of their neighbors. This confirms that different centrality metrics provide complementary perspectives on influence in the collaboration network: while degree and eigenvector highlight the hubs, closeness and harmonic emphasize well-positioned artists, and PageRank captures structural importance beyond pure connectivity.

4 Task 1: Static Community Detection

We applied four community detection algorithms from the CDLIB library: Label Propagation, Louvain, Infomap and Greedy Modularity. Their performance was evaluated through internal metrics (modularity, cut ratio, average internal degree, conductance) and external validation with respect to artist genres. Louvain obtained the best results, with the highest modularity and the most coherent communities both structurally and semantically. Label Propagation also performed reasonably well, while Infomap and Greedy Modularity produced more fragmented or less consistent partitions.

For this reason, Louvain was chosen as the reference algorithm for the subsequent analysis of community composition.

4.1 Internal evaluation

We used the CDLIB implementations of Label Propagation, K-clique, Demon, and Louvain. For K-clique ($k \in [2, 10]$), Demon ($\epsilon \in [0.1, 0.9]$, $\texttt{min\_com\_size} \in \{3, 4, 5\}$), and Louvain (resolution

$\in [0.1, 0.9]$, with randomization), we performed a random search and selected the best configuration according to Girvan–Newman modularity Tables 5 and 6 summarize the results.

Algorithm	Parameters	#Communities	Modularity
Label Propagation	–	1317	0.7199
K-clique	$k = 3$	105	0.0268
Demon	min_com_size=3, $\epsilon = 0.3$	59	0.0250
Louvain	res=0.7, rand=True	713	0.8274

Table 5: Number of communities and modularity (best configurations).

Algorithm	Cut ratio	Avg. internal degree	Conductance
Label Propagation	4.27×10^{-5}	1.2131	0.1758
K-clique	1.2159×10^{-3}	2.6336	0.7952
Demon	9.8448×10^{-4}	3.5994	0.7271
Louvain	8.3933×10^{-6}	0.9199	0.0363

Table 6: Internal metrics: cut ratio, average internal degree, conductance.

Summary. Louvain attains the highest modularity (0.8274) and the best structural metrics (lowest cut ratio and conductance) with full node coverage, thus we select it for the subsequent community analysis. Label Propagation performs reasonably well but yields a more fragmented partition. Overlapping methods (K-clique, Demon) cover only small fractions of nodes and show poor internal separation.

4.2 External evaluation

We compared the partitions obtained by Label Propagation, Louvain, Infomap and Greedy Modularity. For the two algorithms covering the whole network (Label Propagation and Louvain) we computed the Normalized Mutual Information (NMI), obtaining a value of 0.694, which indicates a partial overlap. For comparisons involving methods with partial coverage we used the Normalized F1 score (NF-1). As shown in Table 7, all scores are rather low, confirming that the algorithms identify quite different partitions.

Algorithm	Label Propagation	Infomap	Greedy	Louvain
Label Propagation	-	0.102	0.013	0.003
Infomap	0.078	-	0.012	0.009
Greedy	0.019	0.008	-	0.067
Louvain	0.023	0.009	0.067	-

Table 7: Normalized F1 score between community detection algorithms

To complement the structural comparison, we exploited the semantic information on artists' genres. We first computed the average purity of Louvain communities with respect to the main genre, obtaining a low value ($24.56\% \pm 11.56\%$). This result, however, is influenced by the extreme granularity of the data (more than 4700 distinct genres in the network).

Switching to a coarser-grained approach provides more interpretable results: several Louvain communities appear dominated by consistent musical areas such as jazz, k-pop, j-pop, samba/bossa nova or pop/r&b. Some illustrative examples are reported in Table 8.

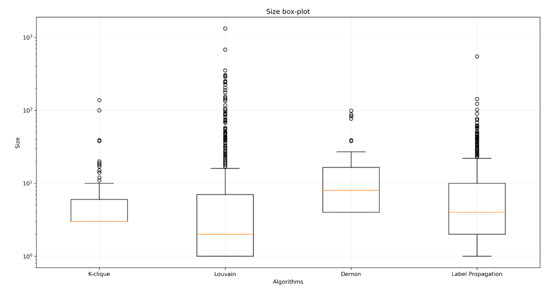


Figure 6: Box-plot of community sizes obtained with different community detection algorithms (K-clique, Louvain, Demon, Label Propagation).

Community label	Size	Top 5 genres in lists
59	143	death metal, brutal death metal, deathgrind, technical death metal, grindcore
38	665	italian adult pop, italian pop, classic italian pop, italian hip hop, genre not available
31	827	k-pop, korean pop, k-rap, genre not available, k-indie
54	261	bebop, jazz, hard bop, background jazz, jazz saxophone
3	2171	j-pop, j-rock, anime, j-rap, genre not available
0	3553	pop, r&b, alternative, hip hop, genre not available
29	900	mpb, samba, bossa nova, pagode, velha guarda
10	1827	chanson, french rock, nouvelle chanson francais, trip hop, genre not available

Table 8: Examples of most frequent genres in Louvain communities

5 Task 2: Link prediction

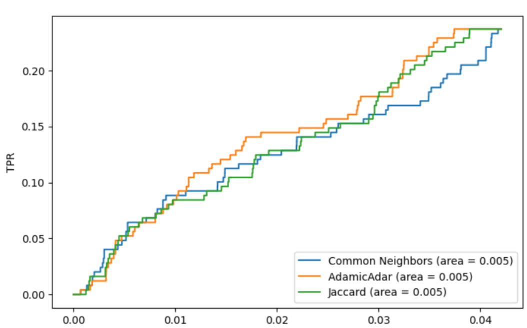
The aim of this task is to evaluate whether and how accurately we can predict potential collaborations between artists that are not directly connected in the current network. To this end, we approach the problem from two perspectives:

- **Unsupervised methods**, which rely on topological similarity measures such as Common Neighbors, Jaccard Coefficient, Adamic–Adar and Preferential Attachment, estimating the likelihood of a future collaboration without the need for any training data.
- **Supervised methods**, which reformulate the problem as a classification task. Here, we construct a labeled dataset of existing and non-existing edges, extract features from the graph (including Node2Vec embeddings), and train machine learning models to evaluate their predictive performance on unseen pairs of artists.

This dual approach allows us to contrast simple heuristic-based predictions with more advanced learning-based models, and to understand the extent to which structural patterns of the collaboration network can support accurate link prediction.

5.1 Unsupervised link prediction

We evaluated several unsupervised link prediction methods relying on network topology, namely Common Neighbors, Jaccard similarity, and Adamic Adar, together with a neighborhood-based similarity approach, SimRank. For the latter, we considered different decay factors $C \in \{0.3, 0.5, 0.7\}$, in order to assess the robustness of results against this parameter. A random baseline predictor was also included for comparison. Common Neighbors, Jaccard similarity, and Adamic Adar are local similarity measures: the first counts the number of neighbors shared by two nodes, the second normalizes such overlap over the union of their neighborhoods, while the third penalizes the contribution of high-degree nodes, giving more weight to shared neighbors with lower connectivity. SimRank instead follows a recursive rationale, assigning a similarity score proportional to the average similarity between the neighbors of two nodes. Due to the high computational cost of these measures — in particular for SimRank, whose complexity scales as $O(|V|^2)$ — the analysis was restricted to the subgraph considered in our experiments. Figure 7 reports ROC and Precision-Recall curves for local measures (Common Neighbors, Jaccard, Adamic Adar). Results show that all three local predictors perform very poorly, with ROC-AUC scores close to zero (CN: 0.0053, AA: 0.0058, Jaccard: 0.0055) and PR-AUC scores barely above the random baseline (around 0.0011 vs. 0.0006). This confirms the limited predictive power of purely local heuristics in this setting. A more significant improvement can be observed with SimRank (Figure 8). Across different values of the decay factor, results are stable, with ROC-AUC values around 0.76–0.77 ($C = 0.3$: 0.7658, $C = 0.5$: 0.7668, $C = 0.7$: 0.7705). In terms of PR-AUC, the three variants achieve scores between 0.0023 and 0.0024, which are still low in absolute terms but consistently better than both the baseline and local heuristics. Overall, $C = 0.3$ provided the best balance in terms of Precision-Recall trade-off, and is therefore selected for further comparisons. Finally, we compared the top-k predicted edges among methods. As expected, Common Neighbors and Adamic Adar shared most of their predictions (177 out of 250), while Jaccard showed more overlap with SimRank ($C = 0.3$) with 140 common predictions. In contrast, almost no overlap was observed between Common Neighbors/Adamic Adar and SimRank, highlighting how different unsupervised measures capture complementary structural signals.



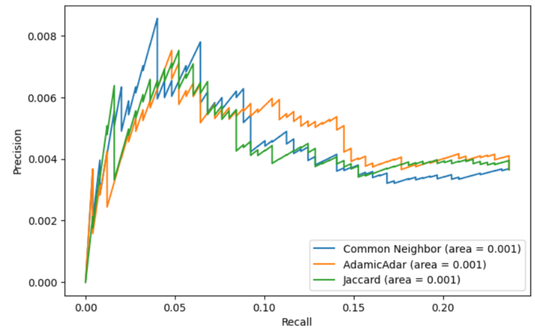


Figure 7: ROC and Precision-Recall curves for Common Neighbors, Jaccard, and Adamic Adar

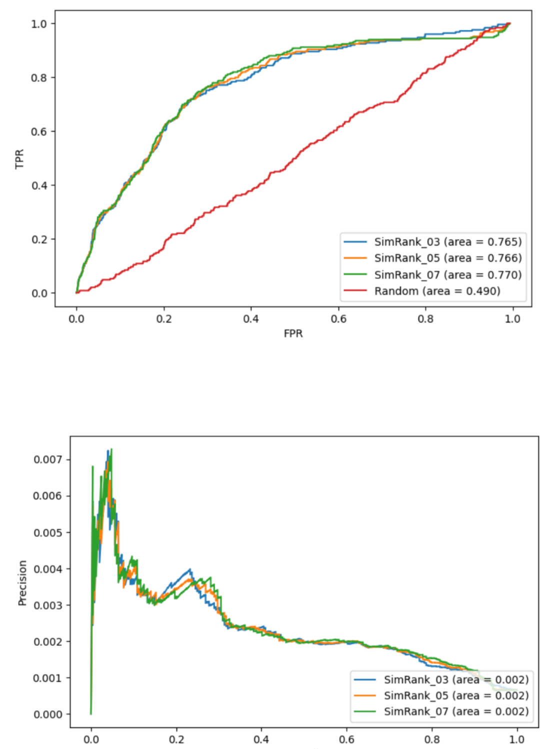


Figure 8: ROC and Precision-Recall curves for SimRank with different decay factors

Method	Common N.	AdamicAdar	Jaccard	SimRank (0.3)
Common N.	250	177	21	0
AdamicAdar	177	250	35	0
Jaccard	21	35	250	140
SimRank (0.3)	0	0	140	250

Table 9: Number of shared predictions among the top 250 scores across unsupervised methods

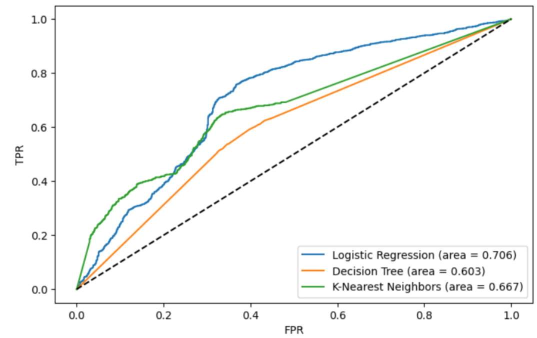
5.2 Supervised link prediction

We also investigated supervised approaches for link prediction, formulating the task as a binary classification problem. To this end, we manually generated a dataset of positive and negative node pairs: a positive label corresponds to an existing edge in the graph, while a negative label corresponds to the absence of such an edge. From the original network we sampled 10% of the links, splitting the resulting dataset into training (50,451 pairs) and testing (74,742 pairs). Node embeddings were generated with the Node2Vec algorithm , which uses biased second-order random walks to produce node sequences and then trains embedding vectors for each node. In order to construct feature vectors for edges, we combined the embeddings of their endpoint nodes using binary operators (Hadamard, Average, L1, L2). On these feature spaces we trained three standard classifiers implemented in scikit-learn: Logistic Regression (LR), Decision Tree (DT), and K-Nearest Neighbors (KNN). For each model, we performed an extensive grid search over hyperparameters using 5-fold stratified cross-validation, selecting the configuration with the highest validation accuracy. Table 10 reports the best validation accuracy (mean ± standard deviation across folds) together with the test accuracy for each classifier and feature space. Results clearly show that the Hadamard operator provides the most informative features, leading to superior performance across all classifiers. Logistic Regression with $C = 0.1$, l_2 penalty and `liblinear` solver achieved the best overall accuracy (73.98%), closely followed by KNN with $k = 7$ (inverse distance weighting). Decision Tree performed less competitively, but still benefited from the Hadamard operator.

Feature Space	Logistic Regression		Decision Tree		KNN	
	Mean VL	TS	Mean VL	TS	Mean VL	TS
Hadamard	73.9% ± 1.5%	73.98%	69.2% ± 1.3%	69.8%	72.5% ± 1.4%	72.9%
Average	68.7% ± 3.2%	68.7%	66.1% ± 2.9%	66.8%	70.3% ± 2.7%	70.5%
L1	66.8% ± 3.0%	66.8%	64.9% ± 2.5%	65.2%	69.1% ± 2.6%	69.5%
L2	67.7% ± 3.0%	67.8%	65.3% ± 2.8%	65.7%	68.4% ± 2.9%	68.6%

Table 10: Supervised link prediction: validation (VL) and test (TS) accuracy for each model and binary operator.

ROC and Precision-Recall curves are shown in Figure 9. Logistic Regression and KNN both achieve good discrimination ability, with ROC-AUC values of 0.790 and 0.774 respectively, while Decision Tree trails behind with 0.732. Although LR achieved the best accuracy, KNN exhibited a slightly more balanced trade-off between true and false positive rates, suggesting better robustness.



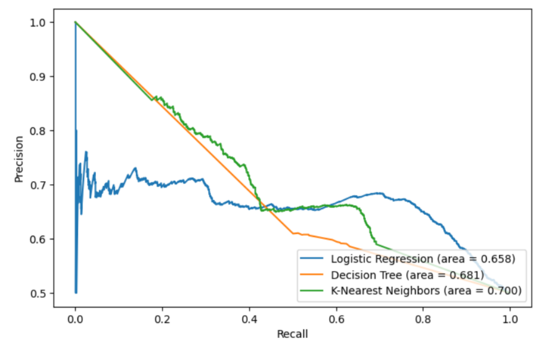


Figure 9: ROC and Precision-Recall curves for supervised models using Hadamard feature space

Table 11 reports Precision, Recall, and F1 scores for the Hadamard feature space on the test set. Logistic Regression achieved the highest F1 score, while KNN showed slightly higher recall, reinforcing the observation from ROC-AUC about its better trade-off.

Method	Precision	Recall	F1
Logistic Regression	0.695	0.854	0.766
Decision Tree	0.642	0.714	0.676
K-Nearest Neighbors	0.684	0.812	0.743

Table 11: Precision, Recall and F1 scores of supervised methods on the Hadamard feature space (test set).

6 Task 4: Open question

6.1 Research question

Do artists collaborate mostly within their own genre, or do cross-genre collaborations prevail? We address this question on the collaboration network collected for the project by quantifying (i) the tendency toward same-genre collaborations (homophily), (ii) the strength of community structure, (iii) the genre-to-genre interaction pattern (mixing matrix), and (iv) the existence of prominent cross-genre connectors (bridging artists).

6.2 Data and macro-genre mapping

We consider the undirected collaboration graph G with **11,083 nodes** and **14,497 edges**. Each node is an artist and carries a first-genre tag (possibly missing). We map first-genre strings to a finite set of *macro-genres* via a deterministic parser (e.g., grouping *"hip hop", "rap", "drill"* into `hip hop / rap`, separating `electronic / dance`, `latin / urbano`, etc.; missing/unavailable genres go to `unknown`). The distribution of macro-genres is summarized below:

Macro-genre	Count	Macro-genre	Count
unknown	3861	r&b / soul	182
other	1999	country / americana	177
hip hop / rap	1552	indie / alternative	163
electronic / dance	844	k-pop / j-pop	114
latin / urbano	650	metal	91
pop	634	reggae / dancehall	71
rock	345	jazz	66
classical	184	afro	53
		folk / cantautore	50
		soundtrack / ost	34
		blues	13

Table 12: Macro-genre distribution over nodes (from notebook outputs).

6.3 Methods

Genre mixing matrix. We build a symmetric *macro-genre* × *macro-genre* matrix counting edges between genres (diagonal = intra-genre edges). The matrix was exported (`matrix_macro_genre_edges.csv`) for inspection and plotting.

Assortativity by macro-genre. We compute the attribute assortativity coefficient r on `macro_genre`. Positive values indicate homophily (same-genre preference).

Null model significance. To assess significance, we generate degree-sequence–preserving random graphs (configuration model + conversion to simple graph and/or double-edge swap) and recompute r on each randomized graph, obtaining μ_{null}, σ_{null}, and a z-score for the observed r.

Communities and modularity. We detect communities with Louvain (`networkx.algorithms.community.louvain_communities`) and compute modularity Q (`quality.modularity`).

Cross-genre connectors. We quantify cross-community connectivity via the *participation coefficient* $P_i = 1 - \sum_c \left(\frac{k_{i,c}}{k_i}\right)^2$, where $k_{i,c}$ is the number of neighbors of node i in community c, and k_i is the degree of i. High P indicates that the node distributes its links across many different communities (potential cross-genre bridge).

6.4 Results

Mixing matrix (qualitative). The genre mixing matrix shows clear diagonal dominance (e.g., `classical-classical`: 97 edges; `country / americana-country / americana`: 124), confirming the prevalence of within-genre collaborations. Off-diagonal entries are non-negligible for closely related scenes (e.g., `hip hop / rap` with `r&b / soul` or `latin / urbano`), indicating structured cross-genre ties rather than random noise. The full matrix was saved for plotting and further analysis.

Assortativity (homophily). The observed assortativity by macro-genre is:

$$\boxed{r_{\text{obs}} = 0.3842}$$

Against degree-preserving nulls we obtain:

$$\mu_{\text{null}} = -0.0018, \quad \sigma_{\text{null}} = 0.0036, \quad \boxed{z = 106.53}$$

Hence, same-genre preference is *highly significant*: the observed r is ~ 100 standard deviations above the null mean.

Community structure. Louvain modularity yields:

$$\boxed{Q = 0.8436}$$

which indicates very strong community structure. Visual inspection (not shown) aligns communities with macro-genre blocs, with bridges at their boundaries.

Cross-genre connectors (participation coefficient). Top bridging artists by P (from the notebook) are reported in Table 13. These are artists whose collaborations span multiple communities/genres, acting as hubs for cross-genre links.

Artist	Macro-genre	P	Popularity
Ed Sheeran	pop	0.835556	92
DJ Premier	hip hop / rap	0.816327	68
Stefflon Don	electronic / dance	0.800000	68
Sofía Reyes	latin / urbano	0.800000	67
Redman	hip hop / rap	0.800000	67
Anitta	pop	0.785714	82
David Guetta	electronic / dance	0.785000	92
Kelis	unknown	0.777778	66
Kane Brown	country / americana	0.765432	76
Zara Larsson	unknown	0.764444	79

Table 13: Top-10 cross-genre connectors by participation coefficient P.

7 Discussion

7.1 Network collection, characterization and preliminary tasks

In this report, we analyzed a collaboration network of Spotify artists using network science tools. The network was built from the top streamed artists on Kworb.net and expanded through Spotify API queries to collect collaborations, resulting in an undirected weighted graph.

- When compared with random graph models, the Configuration Model (CM) best reproduces the degree distribution of the collaboration graph, while the Barabási–Albert (BA) model captures the presence of hubs but tends to overestimate their size. Erdős–Rényi (ER) and Watts–Strogatz (WS) models fail to reproduce the heavy-tailed nature of the real network. Overall, the real graph is sparse and fragmented, with an average degree far below $\ln(N)$, consistent with the existence of many small components alongside a giant component.

- The collaboration network has a higher clustering coefficient than ER and CM models, reflecting triadic closure typical of real collaboration processes. However, clustering is still much lower than in the WS model, which enforces high transitivity by construction. The real network also exhibits the largest estimated diameter and average shortest path, consistent with its sparsity and fragmentation.
- Regarding centrality, degree and eigenvector highlight the most prolific collaborators, while PageRank emphasizes artists connected to other central nodes. These measures identify different notions of influence: hubs with many links versus connectors that bridge important neighborhoods.
- For community detection, Louvain achieved the best results in terms of modularity ($Q = 0.84$) and internal evaluation metrics, and was therefore selected as the reference partition. At a coarser level of genre aggregation, Louvain communities align well with macro-genres (e.g., jazz, k-pop, latin/urbano), confirming strong homophily in the network.
- In link prediction, local unsupervised heuristics (Common Neighbors, Jaccard, Adamic–Adar) performed poorly, consistent with the network's disassortative mixing. SimRank achieved significantly better performance (ROC-AUC ≈ 0.77). In the supervised setting, Logistic Regression with Hadamard operator on Node2Vec embeddings reached the best results ($\approx 74\%$ accuracy), showing that structural information does provide predictive signals when adequately modeled.

7.2 Open question: within- vs. cross-genre collaborations

We investigated whether collaborations mainly occur within genres or across genres. Results show a clear prevalence of homophily: assortativity by macro-genre is $r = 0.38$, with a z-score above 100 compared to degree-preserving null models, and Louvain modularity confirms a strong genre-based community structure. Nonetheless, we also identified bridging artists with high participation coefficients (e.g., Ed Sheeran, David Guetta, Anitta) who connect otherwise separated communities and enable cross-genre collaborations. This suggests a dual pattern: strong within-genre clustering, complemented by a small set of cross-genre connectors.

7.3 Future Work

Future directions include: (i) extending the analysis to temporal dynamics, to study how collaborations evolve over time and whether bridges emerge early or late in artists' careers; (ii) exploiting edge weights more systematically in link prediction and community detection to capture the strength of collaborations; (iii) exploring diffusion processes (e.g., SIR, threshold models) to simulate how influence or trends propagate through the collaboration network. These extensions could further enrich the understanding of structural and dynamic aspects of music collaborations.