Link Prediction in a Music Artist Collaboration Network

Michael Xu, Aaryabrat Chhatkuli, Matthew Jing

Washington University in St. Louis, St. Louis, 63130

December 13, 2025

Abstract

This project explores different mechanisms to find out which best predicts future collaborations in a real-world music artist network. Using the Discogs database to construct an artist-artist collaboration graph labeled by release year and Spotify metadata to obtain genres, popularity, and group membership, our goal is to understand the underlying reasons why collaborations form between artists and which network driven processes explain them. We frame link prediction as a supervised temporal ranking problem: considering the collaborations before a cutoff year, we aim to predict which artist pairs collaborate afterward.

We compare three mechanism-based link prediction approaches, inspired by link prediction theory: (1) local neighborhood closure using Common Neighbors, Jaccard, Adamic–Adar, and Preferential Attachment; (2) community-based prediction using both genre-based communities and collaboration-driven communities to test homophily; and (3) a non-graph baseline that predicts collaborations from artist attributes such as genre overlap, popularity, and follower counts. This baseline provides a point of comparison to evaluate how much of collaboration behavior can be explained beyond simple metadata similarities.

We evaluate predictions and performance using AUC, Precision, Recall, and Hits@1000 across five historical eras of music, which allows us to study how predictability changes over time. To complement the quantitative evaluation, we also visualize the network using Gephi to examine community structure and motif patterns across eras. All together, these analyses help us study how collaboration mechanisms evolve historically and provide a deeper understanding of link formation in the music industry. Our goal is to determine whether future collaborations are best explained by triadic closure, community homophily, or attribute similarity.

Keywords: **Mechanism-Based Link Prediction**, **Music Graph Mining**, **Temporal Network Analysis**

1 Introduction

1.1 Problem Statement

In this project, we have a temporal collaboration graph $G = (V, E)$ at time t, where nodes represent artists and edges represent collaborations, our goal is to predict the new edges that form at time $t + 1$. For structural link prediction, we use only structural similarity scores that include Common Neighbors, Adamic–Adar, Jaccard, and Preferential Attachment. In essence, our overall problem of link prediction is a supervised temporal ranking problem.

1.2 Motivation

Link prediction in networks focuses on identifying which new edges will form in a network based on its past structure. This is a problem that appears across various fields such as biology, social networks, and recommendation systems. In this project, we look at link prediction in the large-scale collaboration network of musical artists using Discogs [1]. As technology develops and cultures shift over time, the patterns in collaborations between artists also changes, which makes this temporal view necessary. This brings us to our question: *Can future collaborations be predicted solely from structural history?*

1.3 Related Work

Our project and our ideas build up on the work of Liben-Nowell and Kleinberg [2], who in their paper introduce the process of using the methods mentioned earlier (Common Neighbors, Adamic–Adar, Jaccard, and Preferential Attachment) for link prediction. Yang et al. [3] further identified the major problems when it comes to link predictions in temporal networks, identifying and mentioning issues such as temporal leakage, class imbalance, and flawed negative sampling. This is why our work takes into account their recommendations of using chronological splits, balanced negatives, and also ensuring careful evaluation, all practices followed in our methodology.

2 Graph-based Link Prediction Methodology

2.1 Data Processing and Parquet Construction

The data used comes from Discogs [1], a comprehensive open-source music metadata database, consisting of XML dumps containing millions of music release records. The first step was ensuring memory-efficient iterative parsing using ElementTree.iterparse on the Discogs XML dump, much so because of its size of over 60GB. Through this process we extracted the artist IDs and release years. In constructing our graph, each release with at least two credited artists was used to form an undirected collaboration edge, and each edge was attributed with the release year. In cleaning up, we removed all releases with invalid or missing dates. In the end, we had a final dataset consisting of over 1 million artists and 5.6 million artist-artist collaboration pairs, stored in a Parquet format. This meant the final network we processed and finalized had over 1 million nodes, 5.6 million edges, with an average node degree (or average collaboration per artists on the grand scheme) of roughly 5.

For the genre-based community detection, the data extraction step was different because Discogs stores genres and styles in the original releases XML file, rather then at the artist level. Due to this, we ran a second iterparse pass through the XML dump, which helped in collecting all genre and style tags associated with each release, and then attached those tags to every artist per release. This left us with a mapping from artist IDs to sets of genres/styles, which we then converted into a DataFrame and merged with full Discogs artist list before saving them as a dedicated parquet file. Artists with missing tags were assigned the label "Unknown", and they were later treated as isolated nodes in the community-based graph.

2.2 Temporal Splits and Link Prediction Pipeline

Our overall goal with this step was to create a single temporal pipeline that all the structural and community-based link prediction methods can follow. The pipeline makes sure that every approach is trained and evaluated on the exact same graph splits, and uses identical rules for defining training edges, future edges, and negative samples. In doing so, we eliminate any inconsistencies across all methods and make the comparison between our different methods meaningful and fair.

In our data engineering step, we had to keep in mind of musical, technological, and cultural differences in the music industry over time. In this process, we came to the conclusion that to evaluate the collaboration dynamics based on their different historical contexts, we had to divide the network into the following five eras: 1897–1945 (Early Recording); 1946–1965 (Post-WWII / Early Rock); 1966–1982 (Classic Rock / Studio Revolution); 1983–1999 (MTV / Digital Sampling); 2000–Present (Internet / Streaming). After this process, each era included only the edges and artists active during its respective time span.

Our final step involved a temporal train/test split. For each era, we calculated a cutoff year at the 80th percentile. All edges before the cutoff year were part of the training graph, and the edges after are our future collaborations, meaning they were part of our test set. To avoid any leakage, we ensured the test edges only included the pairs of nodes who appear in the training graph, as it would not make sense logically to test for a future collaboration on an artist who appears after the cutoff year, because we cannot possibly predict future collaboration for them.

Then we identified all real future collaborations after the cutoff year and used them as our positive examples. To ensure meaningful and unbiased evaluation due to class imbalance, we sampled an equal number of negative examples which included pairs of artists part of the training graph that never actually collaborated. This approach follows the recommendations of Yang et al. (2015) where they highlighted the importance of balancing negative sampling in link prediction problems.

For the genre-based community prediction, artists whose genre was "Unknown" were treated as belonging to an isolated node or a standalone category, so they would not artificially inflate the community density.

2.3 Structural Link Prediction

With the temporal pipeline ready, we then move to the structural link prediction methods by computing four structural similarity scores—Common Neighbors, Jaccard, Adamic-Adar, and Preferential Attachment—for both the positive and negative pairs. All the methods used were originally proposed by Liben-Nowell and Kleinberg (2007), and this give us different perspectives of structurally understanding how links form in networks, like are nodes with a lot of shared neighbors more likely to connect or are high degree nodes more likely to attract more links. By giving scores to both the positive and negative pairs, we were able to assess how well these structural methods can differentiate between actual future collaborations vs. non-collaborations. Lastly, we stored all the scores for each pair of nodes and their corresponding true label, i.e. 1 for a real future collaboration and 0 for no future collaboration. This way, we were able to ensure an analysis independent of any external metadata or artist-specific information, solely focusing on the predictive strength of pure network structure.

2.4 Community-based Prediction

2.4.1 Predictions using Collaboration-based Communities

For collaboration-based communities, we use the same pipeline we used for the structural link prediction by first detecting **collaboration-driven** communities using the Louvain algorithm on each era's training graph. We used the Louvain algorithm on the unweighted training graph of each era with a resolution of 1.0, applied to the largest connected component. We then compute two community based similarity scores-Same Community Flag and Within-Community CN. The same community flag is an indicator of whether or not two artists are in the same detected community. And then Within-Community Common Neighbors is a count of the shared collaborators that are within the same community. The features that are computed test **homophily**: are artists within the same community more likely to collaborate frequently? We then score the actual collaborations (positive pairs) and the non-collaborations (negative pairs). By doing this we can assess how well community structure alone can differentiate between future links and non-links. We then analyze homophily across all five of our eras and then compare the community-based features with the structural features(Common Neighbors, Jaccard, Adamic-Adar, and Preferential Attachment).

2.4.2 Predictions using Genre-based Communities

For genre-based communities, we used almost the same pipeline as the structural pipeline. But instead of solely relying on structural features we used artist genre metadata from Discogs to compute three genre-based similarity features for both positive and negative pairs. We computed Genre Jaccard, Genre Overlap, and Same Genre Flag. Genre Jaccard is the normalized overlap of genre sets, Genre Overlap is shared genres, and Same Genre Flag represents whether or not artists share a genre. These features also test homophily but for genre: do artists with similar musical styles collaborate more frequently? By evaluating both actual future collaborations and non-collaborations, we can assess how well genre similarity alone can differentiate between future links and non-links. We then analyze the genre based homophily across all five of our eras and then compare the genre-based features to the structural based features and then collaboration based features using ROC-AUC, Precision, Recall, and Hits@K as our metrics across all the eras.

3 Non-Graphical Link Prediction Methodology

3.1 Data Preprocessing and Preparation

The non-graphical link prediction data came mostly from the same pipeline outlined in 2.1 Data Preprocessing and Preparation. The general intuition was to perform a prediction using metadata for every pair of artists in a given era. However, due to computational restraints, a sample of 1000 nodes (artists) was selected from era 5. Because not all artists from the Discogs dataset were on Spotify and/or the API call didn't go through, the set of 1000 was selected from era 5 on the condition that Spotify metadata was successfully retrieved for each artist. This provided genre, popularity, and followers for each artists which would later be used in the prediction algorithm.

3.2 Prediction Algorithm

Given metadatas genre, popularity, and followers for each artists, the prediction algorithm sought to calculate a "score" for each metric between any two artists, then assign a weight for which the weighted sums of all three metrics would represent the probability that an edge existed between the two given artists. For genre, the score was calculated as the length of the set of intersecting genre's divided by the length of the set of unionized genres, giving a final value between 0 and 1. For popularity, the algorithm took the absolute value of difference between the two popularity scores, divided by 100, then subtracted from 1 (for normalization). Finally, for followers, the algorithm took the log of each artists' follower count (to normalize for large follower counts), then calculated the difference, dividing by 20 (as a benchmark), finally subtracting from 1 (for normalization). Weights were dynamically assigned to each of these three scores via input parameters in the function.

3.3 Backtesting

It's not immediately clear what parameters make the most sense for this prediction algorithm, so backtesting was used to computationally find the most optimal parameters that would yield the most accurate predictions. Given that the final value of the prediction algorithm needed to be a value between 0 and 1 since it represented the probability that the edge should be present, the backtesting used inputs between [0, 1] for each metric weight. The output set of edges was then compared with a true set of edges, and the precision of each combination in the back test was used to evaluate efficiency of the parameters. Surprisingly, backtesting found that genre weight = 1 with popularity weight = follower weight = 0 yielded the highest precision. Using this parameter, the prediction algorithm was ran again to get the empirically most optimal set of edges (represented as a graph) which would be used for later analysis.

4 Evaluation and Results

To evaluate all our link prediction models, we used the following four metrics, that helped ensure a comprehensive assessment that was a mix of ranking quality, prediction accuracy, and effectiveness of retrievals under class imbalance.

- **AUC (Area Under the ROC Curve):** To measure the model's ability to rank true future collaborations higher than non-collaborations across all possible thresholds.
- **Precision@0.5:** This gives the proportion of predicted collaborations with scores above 0.5 that which were true future collaborations.
- **Recall@0.5:** This gives the fraction of true future collaborations that are correctly identified using the same 0.5 threshold.
- **Hits@1000:** To capture the proportion of true future collaborations out of the top 1,000 highest-ranked predicted artist pairs.

4.1 Structural Link Prediction

For each pair of artists, each of the 4 methods used gives a score, which we evaluated using AUC (the ability to rank true links above false ones), Precision@0.5 (the proportion of predictions with a score of 0.5 that are true), Recall@0.5 (the proportion of true links that are correctly predicted using the same 0.5 threshold), and Hits@1000 (the proportion of the top 1000 scored predictions that were actual true future collaborations). We performed this evaluation on all 5 eras and got the following results:

Era	Method	AUC	Precision	Recall	Hits@1000
Era 1	AA	0.8619	1.0000	0.00022	0.0564
	CN	0.8628	1.0000	0.00022	0.0561
	JACCARD	0.8625	1.0000	0.08287	0.0592
	PA	0.9266	1.0000	0.00022	0.0561
Era 2	AA	0.9244	1.0000	0.00003	0.0056
	CN	0.9246	1.0000	0.00045	0.0041
	JACCARD	0.9241	0.9991	0.03797	0.0041
	PA	0.9570	1.0000	0.00222	0.0041
Era 3	AA	0.9253	1.0000	0.00008	0.0030
	CN	0.9257	1.0000	0.00009	0.0032
	JACCARD	0.9253	0.9997	0.03482	0.0030
	PA	0.9609	1.0000	0.00135	0.0030
Era 4	AA	0.8869	1.0000	0.00004	0.0032
	CN	0.8873	1.0000	0.00004	0.0032
	JACCARD	0.8869	1.0000	0.03598	0.0032
	PA	0.9362	1.0000	0.00241	0.0032
Era 5	AA	0.7525	1.0000	0.00005	0.0026
	CN	0.7528	1.0000	0.00005	0.0026
	JACCARD	0.7528	0.9996	0.02524	0.0026
	PA	0.8874	1.0000	0.00085	0.0026

Table 1: Evaluation Metrics for Link Prediction across Musical Eras

When looking at the results across all eras, Table 3 shows that Preferential Attachment (PA) has the highest AUC, which shows a strong influence of degree-based methods on predicting future collaborations. Jaccard, on the other hand, has the highest recall across the eras, which shows that normalized neighborhood overlap is the most effective when it comes to identifying true future links, especially in the earlier and denser collaboration periods. On the flip side, Common Neighbors (CN) and Adamic-Adar (AA) show similar performances, giving a reasonable AUC values but extremely low recall, which shows their behavior when it comes to sparse graph with very few node pairs sharing direct neighbors.

Overall, precision remains near 1.0 for all methods on all eras, which was due to the balanced evaluation setup and the high threshold, but at the same time comes with extremely low recall for CN, AA, and PA, which shows that most true future collaborations are not predicted by these methods. Jaccard does slightly better but still does not fully overcome this struggle and tradeoff. There is also a clear downward shift in Hits@1000, as it drops from around 5-6% in Era 1 to almost zero in Era5, which shows a decline in structural predictability since over time collaborations have become more diverse and less tied to local graph structure in the era of the internet/streaming. Overall, structural heuristics capture meaningful but limited predictive signals, and their performance going down over time suggests the need for more expressive temporal models or those that focus on underlying factors when it comes to predicting future collaborations in network such as the modern artist-artist collaboration network.

4.2 Community-based Prediction

4.2.1 Predictions using Collaboration-based Communities

Era	Method	AUC	Precision	Recall	Hits@1000
Era 1	same_community	0.9156	0.9696	0.8581	0.631
	within_community_cn	0.8429	1.0000	0.0022	0.203
Era 2	same_community	0.8845	0.9681	0.7953	0.889
	within_community_cn	0.8573	1.0000	0.0004	0.090
Era 3	same_community	0.8592	0.9797	0.7336	0.932
	within_community_cn	0.8241	1.0000	0.0001	0.105
Era 4	same_community	0.8378	0.9767	0.6921	0.929
	within_community_cn	0.7860	1.0000	0.0006	0.085
Era 5	same_community	0.8607	0.9700	0.7445	0.908
	within_community_cn	0.7266	1.0000	0.0001	0.223

Table 2: Evaluation Metrics for Collaboration-Based Prediction across Musical Eras

After doing some analysis on the data produced we can see that the same_community method shows a strong performance across all the eras with AUC values being relatively high along with a recall being consistently above 0.69. Compared with the baseline structural model we can see that recall was substantially better since it was basically zero for all the methods used there. However, we can also see that the within_community_cn performs similarly to the original CN heuristic where precision remains basically 1 but the recall is extremely low. This indicates that this method only identifies a handful of true future links even though the predicted links are usually correct. Across all the eras we can see that our same_community method shows a decline in performance from Era 1 to Era 4 which reflects the same trend observed in the original structural results: predictability in collaboration decreases as the music network becomes more diverse. Overall, these results show that community-based heuristics allow us to capture a more detailed and richer predictive structure than purely using the local structural features. The same_community approach allows the model to leverage a more broader network organization than focusing on direct neighbor overlap or degree which makes it a lot more effective for modeling how artists collaborate in changing ecosystems.

4.2.2 Predictions using Genre-based Communities

Era	Method	AUC	Precision	Recall	Hits@1000
Era 1	genre_jaccard	0.8094	0.7401	0.6391	0.981
	genre_overlap	0.9332	0.9995	0.2365	1.000
	same_genre	0.6897	0.6171	1.0000	1.000
Era 2	genre_jaccard	0.8418	0.8095	0.5358	1.000
	genre_overlap	0.9048	0.9957	0.1573	1.000
	same_genre	0.7279	0.6476	1.0000	1.000
Era 3	genre_jaccard	0.8503	0.8092	0.5395	1.000
	genre_overlap	0.9159	0.9972	0.1670	1.000
	same_genre	0.7559	0.6719	1.0000	1.000
Era 4	genre_jaccard	0.8756	0.8342	0.5281	1.000
	genre_overlap	0.9168	0.9977	0.1172	1.000
	same_genre	0.8074	0.7219	1.0000	1.000
Era 5	genre_jaccard	0.8481	0.7805	0.6108	1.000
	genre_overlap	0.8917	0.9989	0.0435	1.000
	same_genre	0.8094	0.7240	1.0000	1.000

Table 3: Evaluation Metrics for Genre-Based Prediction across Musical Eras

The genre_jaccard method performs consistently well across the five eras with a balanced precision and recall. This is stable and thus shows that the normalized genre_overlap is a reliable signal for collaboration. Artists that share more genres with each other are more likely to collaborate. The Jaccard normalization applied also prevents genre-rich artists from dominating the predictions. Moving on to the genre_overlap method. This method indicates the highest AUC of all three of the genre methods which indicates that it has great ranking ability. However the recall is extremely low, meaning that it only identifies a small set of true links. But the high precision implies that when overlap exists, genre_overlap is a strong indicator of collaboration. This result is most similar to PA in the structural results. Lastly, the same_genre feature indicates the highest recall and perfect hits@1000 across all the eras which shows that every true collaboration occurred between artists sharing at least one genre label. However, the lower AUC and lower precision indicate that there is some generalization going on. This method alone captures that collaborations usually happen within genres but it might not be exactly accurate. Across all the eras, the genre-based prediction performance remains relatively stable only with some drop offs by Era 5. This is different compared to the structural features which showed a steady decline over time. The lack of a sharp performance drop indicates that genre similarity remains a strong indicator in music collaboration even though network structures become more diverse and unpredictable.

4.3 Non-Graphical Link Prediction

Using the optimal parameter of genre weight = 1 and popularity/followers weight = 0, analysis was performed on the resulting set of edges (represented as a graph) to give the following metrics:

Method	AUC	Precision	Recall	Hits@1000
Non-Graphical	0.5352	0.2095	0.6403	0.0014

Table 4: Evaluation Metrics for Non-Graphical Link Prediction

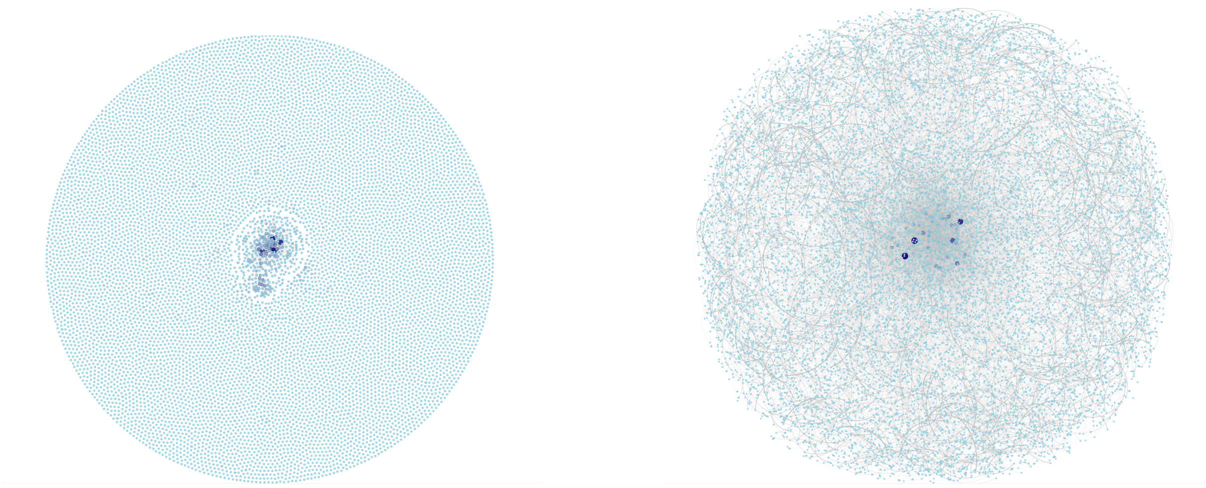
When looking at these results, the most telltale sign that this model is not viable is the AUC value. Being that it is only slightly higher than 0.5, this means that the non-graphical link prediction algorithm performed only marginally better than random guessing. This underpinned by the recall value, which only demonstrates moderate ability of the model to correctly identify true edges. Precision is also rather low, which suggests that there may be a degree of precision recall tradeoff, where the model is predicting a large amount of edges which results in recall increasing artificially (due to random chance). Hits@1000 is very low at 0.0014. While this may be caused in part to the network's large size and sparsity, it is still far to low to contribute to the efficacy of this model.

4.4 Structural vs. Community-based vs. Non-graphical Link Prediction

Given that the metrics for the Non-graphical Link Prediction (NGLP) were derived from a dataset with artists from era 5, the primary analysis to follow will use era 5 metrics from the Structural Link Prediction (SLP) as well. However, given similarities across the eras, conclusions made can be generalized.

Beginning with AUC, it's clear to see that for SLP has much higher AUC, suggesting that the predicted edges are far more accurate than just random predicting which is what the NGLP barely surpasses. This is the convincing evidence already that SLP is more effective than NGLP. As for precision and recall, it's important to mention earlier mentions that precision-recall tradeoff is at play here. Specifically, the tradeoff is much greater in the SLP than the NGLP. This means that the predictions (namely AA, CN, and PA) are fairly conservative, each producing few links with a high probability that they are correct. The Jaccard model is the exception to this with a significantly higher recall compared to the other metrics. This demonstrates the idea that Jaccard is a more forgiving and generous probability giving function. Finally, it's observed that era 5's Hits@1000 are almost double that of NGLP directly signaling higher efficiency in its predictions.

Following this comparison, it's clear to see that SLP tend to give a better guessed result for link prediction compared to NGLP which tends to improve just barely the baseline of random guessing.



(a) Era 1, Jaccard

(b) Era 5, Jaccard

Figure 1: Comparison in predicted graphs between Era 1 and Era 5 using Jaccard

Figure 1 illustrates how using only structural link prediction fails in more recent eras. Graph (b) in Figure 1 shows far more connections (likely due to advances in technology) but doesn't have many distinct communities, highlighting a key limitation of structural link prediction: it ignores semantic components such as genre, style, audience, and culture. Consequently, structural link prediction does not capture communities effectively. To build a more accurate hybrid model, it is therefore necessary to incorporate community detection that accounts for these semantic factors.

5 Hybrid Model

5.1 Methodology

From the conclusions from analysis on structural link prediction and community detection, we find that Jaccard and Common Neighbors are most effective to be utilized in a hybrid model. In our hybrid model we aimed to use Jaccard and Common Neighbors as a baseline for link prediction, and community detection for a more robust prediction. In doing so, we modeled the following prediction algorithm given starting and ending nodes u and v:

$$\begin{aligned}\text{Prob}(u, v) = {} & w_{cn}\,\text{CommonNeighbors} + w_{jaccard}\,\text{Jaccard} \\ & + w_{\text{genreJaccard}}\,\text{GenreJaccard} + w_{\text{genreOverlap}}\,\text{GenreOverlap} \\ & + w_{\text{sameGenre}}\,\text{sameGenre} + w_{\text{communityWeight}}\,\text{communityWeight}\end{aligned} \tag{1}$$

In consideration of time and efficiency, the method we employed to calculate semi-optimal values for each weight was to fix all other weights and run our prediction iterating in increments of 0.1 between 0 and 1 for the current weight. Our optimal results weights were then calculated as follows:

$$\begin{aligned} w_{cn} &= 1.0 \\ w_{jaccard} &= 1.0 \\ w_{\text{genreJaccard}} &= 0.0 \\ w_{\text{genreOverlap}} &= 0.0 \\ w_{\text{sameGenre}} &= 0.2 \\ w_{\text{communityWeight}} &= 0.8 \end{aligned}$$

We then ran the prediction model again to populate a column of prediction scores for every edge. The edge was then denoted as existing in our predicted graph if the prediction value was above 0.5.

5.2 Evaluation and Results

We extracted a subgraph from Era 5 by filtering it such that it consisted of 1000 artists with all necessary structural, genre, and community information, such that it gave us 500,000 artists-artists pairs. This final subgraph was used to evaluate our final hybrid model. Using the aforementioned weights, the final model showed strong performance across all the evaluation metrics. Particularly, it gave an AUC-ROC of approximately 0.99, which helped affirm its strengths when it comes to globally ranking the true vs false future collaborations. It gave a Precision-Recall AUC of above 0.50, which considering the extreme class imbalance, is inherent when it comes to link prediction tasks. When we used a prediction score threshold at 0.5, the model gave the a much higher recall than any sole structural heuristic used earlier, while maintaining a high precision as well. Finally, in a top-K hits evaluation, when looking at the top-10 vs top-100 hits, the model was perfect with exactly 10 matches and 100 matches on both, and even when looking at the top-1000 hits the model was correct on 798 hits, meaning it identified almost 80% of true future collaborations within the top 1000 ranked predictions. This was a significant improvement over the earlier structural standalone predictor models.

Here, it is important to note that this hybrid model is evaluated over the full set of artists pairs, without using any negative sampling, while earlier structural, community-based, and genre-based evaluations used balanced positive-negative sets to isolate individual mechanisms. Because of this, the hybrid model here reflects a much more challenging retrieval task, and due to that results like this show the strength of combining these different models into one hybrid models as this is more representative of real-world prediction tasks.

So, the results that we have show that when we combine local structural similarity with collaboration-based community information then we get a much more effective predictor when it comes to predicting future collaborations than any individual mechanism alone. While Common Neighbors and Jaccard similarity help in differentiating likely links within dense regions of network, community based information gives an idea on the higher-level structure that exists even as the network grows and becomes large and sparse. Genre features also help maintain slight stylistic compatibility, but all in all the model's performance is driven mainly by the network-based information. Overall, with this hybrid approach, it helps successfully balance the ranking accuracy and precision/recall of retrievals. With this we can confidently say that when it comes to predicting future collaborations in a modern music network, the best results are achieved by integrating multiple structural and community-level mechanisms rather than relying on popularity, metadata, or local structure in isolation. To ensure and validate generalization, further work can be done on a fully held-out temporal test set.

6 Conclusion and Recommendations

This project shows that when it comes to predicting future collaborations in large-scale music networks, it cannot be explained by a single mechanism. While structural methods can help capture local effects such as triadic closure and degree-based attachment, their predictive power goes down in the modern eras, since collaboration diversity increases. Community-based signals give a stronger and more stable explanation as they capture relevant community structural while genre similarity acts as a limitation on collaboration. Metadata-only approach like with the non-graphical link prediction, however, fails, as it cannot model the relational dynamics that drive why and how collaborations form in the future. With the mix of local structure with collaboration-based community information, the hybrid model shows best how collaborations form in practice, while also showcasing a strong performance when looking under a more realistic, and large-scale lens. Thus, our recommendation is a community-aware, network-driven approach, that is enhanced by lightweight context-based features (like with genres), as this proves to be the most effective strategy when it comes to link prediction in an evolving artist-artist collaboration network.

7 Statement of Contributions & Work Division

- **Aaryabrat Chhatkuli** Data Processing, Link Prediction Pipeline, Structural Link Prediction
- **Michael Xu** Community-based Prediction—both Collaboration-based and Genre-based
- **Matthew Jing** Non-graphical Link Prediction, Gephi Analysis, Hybrid Model

References

[1] Discogs, "Discogs: Music database and marketplace," 2025. Accessed: 2025-11-29.

[2] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 556–559, ACM, 2003.

[3] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, "Evaluating link prediction methods," *Knowledge and Information Systems*, vol. 45, no. 3, pp. 751–782, 2015.

[4] "Modeling artist influence for music selection and recommendation: A purely network-based approach," *Harvard Data Science Review*, 2020.

[5] K. Jacobson, M. B. Sandler, and B. Fields, "Using audio analysis and network structure to identify communities in on-line social networks of artists," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2008.

[6] Y. Yoo *et al.*, "Quantitative analysis of a half-century of k-pop songs: Association rule analysis of lyrics and social network analysis of singers and composers," *Journal of Popular Music Studies*, vol. 29, no. 3, pp. e12225–n/a, 2017.

[7] T. South, "Network analysis of the spotify artist collaboration graph," tech. rep., Australian Mathematical Sciences Institute, 2018.

[8] S. Donker, "Networking data. a network analysis of spotify's socio-technical related artist network," *International Journal of Music Business Research*, vol. 8, no. 1, pp. 67–101, 2019.

GitHub Repo: `https://github.com/cse4106-fl25/finalproject-matthew-michael-aarya`