Group 28: Comparing Interpretable Music Recommendation Systems through Graph and Feature-Based Visualizations

Austin McGregor, Bryan Shook, Kai Hackney, Langchao Shen, Myeongsun Choi

1 Introduction

Modern music streaming platforms contain millions of songs, making it difficult for listeners to discover music that matches their preferences. Recommendation systems help address this problem by suggesting songs based on listening history or similarities between songs. However, many existing systems rely on complex models that produce recommendations without clearly explaining why they were generated.

Lack of transparency limits users' ability to explore music and understand the factors influencing recommendations. In particular, users often cannot distinguish whether recommendations arise from listening behavior, musical similarity, or genre relationships.

In this project, we design an interpretable music recommendation system that combines two complementary approaches: collaborative filtering based on listening patterns and a graph-based recommendation method based on semantic relationships between songs, artists, and tags. The system also provides visual exploration tools that allow users to examine how songs relate to one another through musical features such as tempo, energy, and popularity.

By integrating recommendation algorithms with interactive visualizations and semantic relationships derived from music metadata, the system aims to make music recommendations both effective and understandable, allowing users to better explore and interpret their musical preferences.

2 Problem Definition

We aim to build a system that recommends songs to a user and clearly shows why those songs are recommended.

Most music recommendation systems produce suggestions without explaining the reasoning behind them. As a result, users receive recommendations but cannot easily understand how their listening behavior, musical attributes, or genre relationships influenced the result.

Our goal is to develop a system that both generates recommendations and helps users understand the relationships between songs. The system will use listening history and music metadata from the Million Song dataset [2] to suggest songs and provide visual tools that allow users to explore how songs are related through musical features, artists, and tags.

3 Literature Survey

Music recommendation has been studied using collaborative, graph-based, and hybrid approaches. A common challenge across recommendation systems is the tradeoff between recommendation quality and interpretability. Volkovs et al. [18] proposed a two-stage hybrid recommender for playlist continuation that combines candidate generation with re-ranking using co-occurrence and audio features, demonstrating strong performance at scale. However, such systems primarily emphasize accuracy and ranking effectiveness rather than transparent explanations for users.

Collaborative filtering remains one of the most widely used recommendation paradigms. Hu et al. [6] developed collaborative filtering methods for implicit feedback data, which are especially relevant for music listening logs where user preferences are inferred from behavior rather than explicit ratings. Ludewig et al. [11] later showed that relatively simple nearest-neighbor methods based on co-occurrence can remain competitive with more complex recommender systems while being easier to interpret. These works motivate the collaborative filtering component of our system, but they do not directly address semantic explanations or user-facing visualization.

Graph-based recommendation offers a complementary direction. Gori et al. [4] introduced ItemRank, a random-walk-based scoring method that applies PageRank-style propagation to recommendation problems. More recently, Musto et al. [15] extended graph-based recommendation with context-aware personalized PageRank, showing how random walks over structured relationships can incorporate richer signals. These approaches are attractive because graph structure can expose interpretable paths between users, items, and metadata. Our work builds on this idea by constructing a music graph over users, songs, artists, and tags, then extending it with ontology-inspired semantic structure.

Ontology-based recommendation has also been explored for music datasets. Ly et al. [12] used ontology with Million Song Dataset entities to generate recommendations from artist matches, tag matches, and song relations through heuristic score aggregation. This work is especially relevant to our project because it demonstrates how semantic relationships among music entities can support explainable recommendation. However, its ranking pipeline is largely heuristic. Our approach instead uses ontology concepts to enrich the graph structure and then applies personalized PageRank to rank songs through multi-hop relationships in a more unified framework.

Several prior studies have focused more directly on explainability and user interaction. Zhao et al. [20] proposed an interpretable song recommender that generates reasons based on user mood, while Andjelkovic et al. [1] 's MoodPlay system showed how interactive interfaces can support mood-based music discovery. These works suggest that recommendation systems become more useful when users can understand the rationale behind suggestions. Our project shares this emphasis on interpretability, but instead of focusing only on mood, we compare explanations derived from listening behavior, graph relationships, and audio-feature visualization.

Visualization research further motivates the interface component of our system. Khulusi et al. [9] surveyed music visualization techniques and highlighted both the breadth of existing methods and the difficulty of making them intuitive for non-expert users. Petridis et al. [16] proposed TastePaths, which uses artist similarity graphs to help users explore preferences through network navigation rather than static ranked lists. Jin et al. [8] showed that visualization design affects how users perceive recommendation diversity and discover hidden interests. Together, these studies support the idea that visualization is not merely a presentation layer, but a tool for helping users interpret recommendation structure.

Finally, graph representation and clustering methods provide useful technical foundations for visual exploration. Grover and Leskovec [5]'s node2vec demonstrated how graph nodes can be embedded into geometric spaces, while Traag et al. [17]'s Leiden algorithm improved community detection by guaranteeing better-connected clusters. Although these methods are not themselves recommenders, they inform how relational music data can be organized and visualized. In our project, we similarly use low-dimensional projection of song features and graph relationships to help users inspect recommendation patterns and compare collaborative filtering with ontology-enhanced personalized PageRank.

Overall, prior work shows that collaborative filtering, graph-based recommendation, ontology modeling, and visualization each offer useful strengths, but these elements are often studied separately. Our project combines them into one interpretable music recommendation system that allows users to compare recommendation signals and explore why songs are suggested.

4 Proposed Methods

4.1 Intuition

We will have two recommendation engines: a collaborative filtering engine and an ontology-based engine. Given a song to generate recommendations for, the collaborative filtering engine will look at what listeners of that song also listened to while the ontology-based engine will find songs with similar genres, moods, styles, etc. Including multiple recommendation engines will give users a better understanding of why certain songs are being recommended.

4.2 Innovations

Our work introduces key innovations focused on improving interpretability in music recommendation systems:

- We integrate collaborative filtering, graph-based methods, and feature-based similarity to allow direct comparison of recommendations.
- We normalize noisy tag data into structured concepts (genre, mood) and introduce concept-level relationships to enable more meaningful and interpretable recommendation paths.
- We apply UMAP to project audio features into 2D to enable interactive visualization of song similarity and cluster structure.

4.3 Algorithms

4.3.1 Collaborative Filtering: The collaborative filtering recommendation system uses an interpretable item-item approach built for implicit feedback listening data. Instead of predicting ratings, the model treats user play histories as signals of confidence that a listener is interested in a song. It then recommends songs that are most similar to multiple input seed songs based on shared listening behavior. We represent the data as a sparse user-song matrix derived from train triplets.txt, where rows correspond to users and columns correspond to songs. There are also weighted versions of entries for observed play counts. Since data consists of implicit feedback based on users' actions, repeated listening would be treated as stronger evidence compared to ratings.

To improve recommendation quality, the raw interaction counts are transformed before similarity is computed. Play counts are log-scaled to reduce the effect of outliers, and an optional TF-IDF-style weighting is used so that songs that are listened to by many users across the dataset are downweighted. In addition, user-activity normalization is applied. Doing so makes highly active listeners have less impact on the similarity structure as they interact with many songs. After weighting, each song is represented by its listener vector, and song-song similarity is computed using cosine similarity over the listener profiles.

By providing one or more songs, the recommendation scores are produced by summing the similarity contributions from all selected seed songs, as well as ranking the highest scoring candidates that are unseen. These explanations are used directly in the interface and are also used to build a recommendation graph showing the similarity of the edges for the selected and recommended songs. For an interactive frontend, we built a fast retrieval index using truncated SVD over the item-user matrix. This allows the system to approximate candidate neighborhoods efficiently while verifying final similarities.

In the interface, the CF Recommender tab lets users search for seed songs and view the top-5 recommendations, each showing its collaborative score, shared listener count, and strongest supporting seed. A D3 force-directed graph below renders the co-listen relationships, with edge thickness proportional to shared listeners.

4.3.2 Ontology-Based: Our ontology-based recommendation system applies Personalized PageRank (PPR) to a heterogeneous graph containing songs, artists, and semantic tag nodes. Recommendations are generated by performing a biased random walk starting from a user's visible listening history or chosen songs over song–artist, song–tag, artist–tag, and tag hierarchy relations. Multiple edge-weight configurations are evaluated offline, and the best validation configuration is used for recommendation.

To build the ontology-aware graph, Last.fm tags and MSD artist terms are first normalized by lowercasing, trimming, merging lexical variants, and removing noisy or weak personal labels. The remaining tags are grouped into semantic categories: genre, style, mood, and era. Tags are then mapped to canonical concept nodes so that equivalent or near-equivalent labels share the same representation.

The ontology layer introduces parent–child taxonomy relations, such as subgenre and broader semantic category links. These relations allow recommendation signals to move beyond exact tag matching. For example, two songs tagged with different subgenres can still be connected through a shared parent genre. Relation types are distinguished explicitly, such as song–genre, song–style, artist–genre, and tag hierarchy edges, so each semantic link type can contribute with a different validated weight.

For evaluation, the system uses visible user listening histories as PPR seeds and hidden listening histories as ground truth. The recommender ranks songs from the local graph neighborhood and computes top-K metrics such as Precision@K, Recall@K, NDCG@K, diversity, novelty, and coverage. This setup lets us compare whether ontology expansion improves not only hit-rate accuracy, but also semantic interpretability and recommendation breadth.

In the interface, the Ontology-based Recommender tab renders a lane-structured SVG graph with nodes color-coded by kind (user, song, artist, tag) and role (visible, hidden, recommended). Users can inspect sampled users' recommendation results or select songs for on-demand recommendation.

4.3.3 Feature-Based Embedding and Visualization (UMAP): In addition to graph-based recommendation, we incorporate a feature-based visualization method to allow users to explore relationships between songs using their audio attributes. Each song is represented as a vector of numerical features derived from MSD metadata including: Tempo (BPM), Danceability*, Energy*, Loudness (dB), Key (0-11), Mode (0=minor, 1=major), Duration (s), Time Signature, Song Hotttness*. Features with a "*" were generated from The Echo Nest and are not objective properties of the songs.

To visualize similarities between songs, we apply Uniform Manifold Approximation and Projection (UMAP) [14] which projects high-dimensional feature vectors into a two-dimensional space. UMAP preserves local neighborhood structure, allowing songs with similar audio characteristics to appear close together in the visualization.

Before applying UMAP, feature vectors are normalized to ensure that attributes with larger scales, such as tempo, do not dominate the embedding.

In the interface, the UMAP View tab renders this projection as an interactive D3 scatter plot where users can toggle which audio features are included, hover to inspect song metadata, and select two songs for a side-by-side feature comparison. Songs added to the shared playlist from any tab are highlighted in the plot.

5 Experiments and Evaluation

There are two types of experiments that we will conduct to validate our recommendation systems: user studies and quantitative offline evaluation.

5.1 User Studies

We will recruit our peers to interact with our visualization and listen to songs recommended to them by our recommendation systems. Then, we will ask them for feedback and prompt them with questions like "was there anything that was difficult to do with the visualization?" and "was there anything missing from the songs recommended to you?" We will use this feedback to improve the final product.

5.2 Quantitative Offline Evaluation

To quantitatively evaluate recommendation quality, we will use the Million Song Dataset Challenge benchmark. For each target user, the visible listening history will be used as input to the recommender, and the hidden listening history will be treated as ground truth. We will tune models on the validation split and reserve the test split for final comparison.

We plan to evaluate top-K recommendation quality using Precision@K, NDCG@K, and Recall@K. Precision@K measures how many of the top-K recommended songs are actually relevant, NDCG@K measures whether relevant songs are ranked near the top of the list, and Recall@K measures how much of the held-out listening history is recovered. We will compare the Precision@K and NDCG@K metrics with the Precision@10 and NDCG of the best recommendation algorithm in the Million Song Dataset Challenge leaderboard[13].

In addition to ranking accuracy, we will also evaluate several beyond-accuracy properties of the recommender. Diversity [10] measures whether the recommended songs are meaningfully varied rather than overly similar. Novelty [19] evaluates whether the recommended songs are relatively less popular or less frequently listened to items. Serendipity [7] measures whether the system recommends relevant songs that are not obvious based on the user's past listening behavior. Catalog coverage [3] measures the proportion of the song catalog that appears in recommendation lists across users, while distributional coverage [7] evaluates how evenly recommendation frequency is distributed across the catalog. These metrics allow us to compare the different recommendation approaches in our system, including collaborative filtering and graph-based methods, not only in terms of ranking accuracy but also in terms of their ability to promote discovery and broader exploration of the music catalog.

5.3 User Study Result

Overall, the qualitative evaluation suggests that the system was successful in providing an engaging and interpretable music recommendation experience. Users were generally able to understand how to interact with the visualization, especially once they reached the Graph PPR view, where songs can be searched, added, and used to generate recommendations. The interface was strongest in its visual clarity: legends, node types, and recommendation lists made the system feel transparent and interactive. The UMAP view was also well received because it allowed users to compare selected songs directly, with feature differences highlighted in a clear way. Recommendation quality was generally positive. For example, when users selected "My Heart Is Yearning" by NOFX, it produced recommendations such as "My Girlfriend's Dead" suggesting that the system could surface songs with related musical characteristics rather than simply repeating identical artists or tracks. The graph-based explanation also helped users understand why recommendations appeared, since the visualization exposed relationships among songs, artists, and tags.

At the same time, the evaluation identified several minor areas for improvement. The starting workflow could provide more guidance, since the application opens with the UMAP view even though users must first add songs from the Graph PPR or CF tab before the UMAP comparison becomes useful. Some graph interactions were less intuitive and would benefit from clearer descriptions, especially because dense tag regions in the 1-hop and 2-hop lanes can be dragged horizontally, but users may not immediately realize that they are draggable. Users also noticed that these dense tag regions could become laggy when too many tags were displayed, although the main graph interaction, node inspection, and recommendation display were generally responsive. In addition, even though recommendation scores were visible, additional explanation of what those scores mean would make the system more interpretable for non-technical users. Finally, it was observed during evaluation that popular artists could dominate the recommendation results. For example, when songs such as "Nothin' On You" by B.o.B or "Crawling" by Linkin Park were included as seeds, many recommendations came from the same artists, and adding less popular songs did not always sufficiently reduce this dominance. This suggests that the current graph-based ranking may overemphasize highly connected artists or popular seed neighborhoods, causing them to dominate the recommendation list even when more diverse personalized choices may be preferable. These issues were relatively small compared with the system's strengths, and the overall feedback indicates that the visualization achieved its main goal: helping users explore recommendations in a way that is more transparent, interactive, and understandable than a conventional black-box recommender.

5.4 Quantitative Evaluation Result

Table 1 shows the result of the quantitative evaluation after running the experiments on the original Million Song Dataset Challenge.

The quantitative results show that Onto-PPR substantially improves top-K precision over the collaborative filtering baseline. It also produces much more novel recommendations. Onto-PPR achieves a Precision@K of 0.0833, compared with 0.0097 for CF, which is an approximately 8.6x improvement. This suggests that

Table 1: Evaluation metrics for three recommenders: Collaborative Filtering (CF), ontology-based Personalized PageRank (Onto-PPR), and the best recommender in Million Song Dataset Challenge Recommender (MSDC). Abbreviations: Precision (Prec), Recall (Rec), Diversity (Div.), Novelty (Nov.), Serendipity (Ser.), Catalog Coverage (Cat. Cov.), and Distributional Coverage (Dist. Cov.).

Metric	CF	Onto-PPR	MSDC
Prec@K	0.0097	0.0833	0.136
NDCG@K	0.0538	0.0542	0.297
Rec@K	0.0970	0.0523	Unknown
Div.	0.9340	0.8763	Unknown
Nov.	0.4390	0.8586	Unknown
Ser.	0.3547	0.0518	Unknown
Cat. Cov.	0.1046	0.0026	Unknown
Dist. Cov.	0.6986	0.5151	Unknown

the ontology-based graph structure is much better at placing relevant songs into the recommendation list. However, Onto-PPR is still below the best Million Song Dataset Challenge recommender, which reports a Precision@10 of 0.136. Onto-PPR reaches about 61% of that benchmark precision, which is a strong result given that our system prioritizes interpretability and visualization in addition to ranking accuracy.

NDCG@K is nearly identical between CF and Onto-PPR, with CF scoring 0.0538 and Onto-PPR scoring 0.0542. This suggests that even though Onto-PPR recommends more relevant songs overall, it doesn't consistently rank those relevant songs much higher in the list. In comparison, the MSDC benchmark achieves an NDCG of 0.297, meaning that both project recommenders still have significant room for improvement in ranking order. An improvement we could make in the future would be to add a reranking step that places stronger matches higher on the rank.

Recall@K shows a different trend from previous metrics. CF achieves 0.0970, while Onto-PPR achieves 0.0523. This shows that CF recovers a larger portion of the hidden listening history, suggesting CF may be better at broadly recovering many held-out songs, while Onto-PPR is more selective and precise. The tradeoff is acceptable for our project goal because the system emphasizes interpretable and high-confidence recommendations rather than exhaustive recovery of all hidden songs.

For beyond-accuracy metrics, Onto-PPR performs especially well on novelty. Its novelty score is 0.8586, while CF earned 0.4390, nearly a 96% improvement. This indicates that Onto-PPR is much more effective at recommending less obvious or less globally popular songs. Diversity remains high for both models, though CF performs slightly better with 0.9340 compared with 0.8763 for Onto-PPR. This means Onto-PPR recommendations are still meaningfully varied, but the graph structure may concentrate recommendations around certain connected regions of the network.

The main weaknesses of Onto-PPR appear in serendipity and coverage. CF has a serendipity score of 0.3547, while Onto-PPR scores 0.0518, suggesting that CF produces more unexpected relevant recommendations. Onto-PPR also has much lower catalog coverage, 0.0026 compared with 0.1046 for CF, and lower distributional coverage, 0.5151 compared with 0.6986. This means Onto-PPR recommends from a much smaller portion of the catalog and distributes recommendations less evenly. This matches our qualitative observation that certain songs, such as "Nothin' On You" by B.o.B and "Crawling" by Linkin Park, appeared repeatedly across different inputs.

Apart from the contents in Table 1, we also tested multiple graph variants, including raw tags, cleaned tags, canonical tags, and ontology-expanded tags. On the final test split of 200 users, the cleaned-tag model achieved the strongest Precision@10 among graph variants at 0.0805, while the ontology-expanded model was close at 0.0775 and achieved the highest novelty at 0.8660. This result shows that cleaning and structuring music metadata has a meaningful effect on recommendations. Rather than one graph variant being best on every metric, each representation emphasized different strengths. This supports the project's broader goal of comparing interpretable recommendation approaches.

Overall, the results show that the two models have different strengths and weakness. Onto-PPR performs better on Precision@K and novelty, suggesting that the graph-based ontology approach is effective at producing relevant and less obvious recommendations. CF performs better on Recall@K, diversity, serendipity, and coverage, meaning it captures a broader portion of user history and explores the catalog more widely. Therefore, the evaluation highlights a tradeoff where Onto-PPR is better for precise and interpretable recommendations, while CF is stronger for broader discovery and coverage. Future work could combine these strengths through reranking, hybrid scoring, or popularity and coverage normalization.

6 Conclusion and Discussion

This project developed an interpretable music recommendation system that uses collaborative filtering and ontology-based Personalized PageRank. The main goal was to not only recommend songs, but also to help users understand why those songs were recommended through factors such as listening history, artist links, tag relationships etc. Overall, the system achieved this goal by connecting recommendation results with visible explanations. The graph view helped users inspect relationships among songs, artists, users, and tags, while the UMAP view gave a complementary feature-based perspective for comparing songs. The user study showed that participants found the interface engaging, transparent, and useful for exploring recommendations beyond a standard ranked list.

The quantitative evaluation also showed that the models captured different strengths. Onto-PPR performed well on precision and novelty, suggesting that the ontology-based graph structure was effective at producing relevant and less obvious recommendations. CF performed better on recall, diversity, serendipity, and coverage, showing that collaborative listening patterns were useful for broader discovery across the catalog. These results support our approach of comparing multiple recommendations rather than relying on a single model. Future work can improve ranking quality, reduce repeated recommendations from highly connected songs, and combine the strengths of CF and Onto-PPR through hybrid scoring or reranking. Overall, the project demonstrates that music recommendation systems can be made more interpretable and exploratory while still producing meaningful recommendation results.

Distribution of Effort: All team members have contributed a similar amount of effort.

References

[1] Ivana Andjelkovic, Denis Parra, and John O'Donovan. 2016. Moodplay: Interactive Mood-based Music Discovery and Recommendation. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization* (Halifax, Nova Scotia, Canada) *(UMAP '16)*. Association for Computing Machinery, New York, NY, USA, 275–279.

https://doi.org/10.1145/2930238.2930280
[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.
[3] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*. 257–260.
[4] Marco Gori, Augusto Pucci, Via Roma, and I Siena. 2007. Itemrank: A random-walk based scoring algorithm for recommender engines.. In *IJCAI*, Vol. 7. 2766–2771.
[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 855–864. https://doi.org/10.1145/2939672.2939754
[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*. 263–272. https://doi.org/10.1109/ICDM.2008.22
[7] Aryan Jadon and Avinash Patil. 2024. A comprehensive survey of evaluation techniques for recommendation systems. In *International Conference on Computation of Artificial Intelligence & Machine Learning*. Springer, 281–304.
[8] Yucheng Jin, Nava Tintarev, and Katrien Verbert. 2018. Effects of Individual Traits on Diversity-Aware Music Recommender User Interfaces. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization* (Singapore, Singapore) *(UMAP '18)*. Association for Computing Machinery, New York, NY, USA, 291–299. https://doi.org/10.1145/3209219.3209225
[9] Richard Khulusi, Johannes Kusnick, Christian Meinecke, Christian Gillmann, Josef Focht, and Stefan Jänicke. 2016. A Survey on Visualizations for Musical Data. *Computer Graphics Forum* 35, 6 (2016), 365–389. https://doi.org/10.1111/cgf.12905
[10] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-based systems* 123 (2017), 154–162.
[11] Malte Ludewig, Iman Kamehkhosh, Noveen Landia, and Dietmar Jannach. 2018. Effective Nearest-Neighbor Music Recommendations. In *Proceedings of the ACM Recommender Systems Challenge 2018*. 1–6. https://doi.org/10.1145/3267471.3267474
[12] Thanh Huy Ly, Song Toan Do, and Thi Thanh Sang Nguyen. 2018. Ontology-based recommender system for the million song dataset challenge. In *2018 10th international conference on knowledge and systems engineering (KSE)*. IEEE, 236–241.
[13] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, and Gert R.G. Lanckriet. 2012. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web* (Lyon, France) *(WWW '12 Companion)*. Association for Computing Machinery, New York, NY, USA, 909–916. https://doi.org/10.1145/2187980.2188222
[14] Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint arXiv:1802.03426* (2018).
[15] Cataldo Musto, Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2021. Context-aware graph-based recommendations exploiting personalized pagerank. *Knowledge-Based Systems* 216 (2021), 106806.
[16] Savvas Petridis, Nediyana Daskalova, Sarah Mennicken, Samuel F Way, Paul Lamere, and Jennifer Thom. 2022. TastePaths: Enabling Deeper Exploration and Understanding of Personal Preferences in Recommender Systems. In *Proceedings of the 27th International Conference on Intelligent User Interfaces* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 120–133. https://doi.org/10.1145/3490099.3511156
[17] Vincent A. Traag, Ludo Waltman, and Nees Jan van Eck. 2019. From Louvain to Leiden: Guaranteeing Well-Connected Communities. *Scientific Reports* 9, 1 (2019), 5233. https://doi.org/10.1038/s41598-019-41695-z
[18] Maksims Volkovs, Hivash Rai, Zhaoyue Cheng, Guangda Wu, Yichao Lu, and Scott Sanner. 2018. Two-stage Model for Automatic Playlist Continuation at Scale. In *Proceedings of the ACM Recommender Systems Challenge 2018*. 1–6. https://doi.org/10.1145/3267471.3267480
[19] Liang Zhang. 2013. The Definition of Novelty in Recommendation System. *Journal of Engineering Science & Technology Review* 6, 3 (2013).
[20] Guoshuai Zhao, Hao Fu, Ruihua Song, Tetsuya Sakai, Zhongxia Chen, Xing Xie, and Xueming Qian. 2019. Personalized Reason Generation for Explainable Song Recommendation. *ACM Trans. Intell. Syst. Technol.* 10, 4, Article 41 (July 2019), 21 pages. https://doi.org/10.1145/3337967