Why didn't more people see it? Recommendation transparency for providers

Meysam Varasteh[1,*], Robin Burke[2]

[1]Department of Computer Science, University of Colorado Boulder, Boulder, CO 80309, USA
[2]Department of Information Science, University of Colorado Boulder, Boulder, CO 80309, USA

Abstract

Transparency in recommender systems has been widely studied from the perspective of those receiving recommendations, yet the needs of item providers, the creators whose content is distributed through these platforms, remain largely unexplored. Providers often lack insight into how their items do or do not receive exposure in users' recommendation lists. In this work, we address this gap by proposing a surrogate modeling approach to explain item exposure at a system level. Rather than explaining individual user–item pairs, we train a proxy model to approximate the exposure distribution produced by a recommender. By quantifying the contribution of each feature, we seek to explain the factors driving the recommendation model's decisions across the entire user base. We evaluate our approach on two datasets and three recommendation models. Results show that the surrogate model captures the global behavior of all three recommenders with high fidelity and that the most influential factors vary meaningfully across models and domains.

Keywords

recommender systems, transparency, explanation, multistakeholder recommendation

1. Introduction

Transparency and interpretability in recommender systems form a significant research area in both academia and industry. This area has focused heavily on the explanation of individual recommender system outputs. Such explanations in recommender systems serve multiple goals. They promote transparency by revealing how the system works, and scrutability by allowing users to correct it when wrong [1]. They build trust and support effectiveness by increasing user confidence and helping them make better decisions. Additionally, explanations enhance persuasiveness by encouraging users to act on recommendations, improve efficiency by speeding up decision-making, and boost satisfaction by making the overall experience more enjoyable and easy to use [2, 1].

We can describe this area of emphasis as one of *local explanation*, explaining the output of an individual user–item pair: why item i is recommended to user u. There is less emphasis on describing aggregated, system-wide behavior of the model, a *global explanation*. While local explanations aim to illuminate the model's behavior for a specific user–item interaction, global analysis requires a broader view of the model's overall recommendation patterns [3].

Many of the forms of local explanation that have been proposed, especially those focused on persuasiveness, tend to drift away from explanations that are grounded in the details of a recommender's decision-making process, which are complex and difficult to communicate, and towards content-oriented features, whether or not such features play a part in recommendation. This may be appropriate for supporting item-level decision-making; however, global explanations of the type that we envision require *veracity*, fidelity to the recommendation model, a property emphasized in work on counterfactual explanation [4, 5, 6, 7, 8].

While there are a variety of global explanations that might be considered, for our purposes, we are focusing on a generalization of the standard "why item i for user u?" explanation ($i \rightarrow u$) to summarize recommendation behavior for an item across all users ($i \rightarrow U$). Some evidence from human-centered

IntRS'26: Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, September 28, 2026, Minneapolis.
*Corresponding author.
meysam.varasteh@colorado.edu (M. Varasteh); robin.burke@colorado.edu (R. Burke)

studies suggests that explanations of this type will be helpful in supporting providers in their interactions with recommenders [9, 10, 11, 12].

To ensure veracity, global explanations will need to be grounded in the structural characteristics of the recommendation data and an item's positioning within the recommendation model. In the purely collaborative models, which we study here, we know that an item's recommendation pattern is frequently influenced by such factors as item popularity [13, 14], user profile density [15], influential users [16, 17], and neighborhood interactions [16]. These structural properties form the foundation of the internal logic of CF models and represent an important yet largely overlooked source of explanations.

Building on these observations, we train a model as a surrogate to predict properties of the $i \rightarrow U$ relation: in particular, to predict the exposure of a target item in recommendation lists across the user base. These models are necessarily specific to the application domain / dataset and the recommendation model. From such models, we can derive correlations between structural features of datasets and recommendation models relative to item characteristics and offer explanations for the variations in system-level exposure across different items.

Our contributions in this work can be summarized as follows:

- We show that item exposure is globally predictable from a small set of structural features, achieving $R^2 = 0.83$–0.93 across three recommendation models and two datasets.
- We demonstrate that dominant exposure drivers differ meaningfully across models and datasets, providing model-specific diagnostic transparency to providers.
- We show that in each algorithm a single feature (but not always the same one) explains $60 - 70\%$ of exposure variance, revealing that RS exposure is driven by a small number of critical signals.

2. Problem Formulation

Given a trained recommender, our goal is to explain the variance of *exposure* across items: why certain items receive higher exposure than others across the entire user base in their top-k recommendations. Let $\mathcal{U} = \{u_m\}_{m=1}^{M}$ and $\mathcal{I} = \{i_n\}_{n=1}^{N}$ represent the sets of users and items, respectively. The historical interactions are captured in the binary matrix $R \in \{0, 1\}^{M \times N}$, where $r_{ui} = 1$ if an interaction exists and 0 otherwise. Additionally, $\hat{r}_{u,i} = f(u, i|\theta)$ is the prediction value where θ represents the learned latent parameters of the model.

For each user u, the model generates a ranked list of size k by sorting items in descending order of their predicted scores. The top-k recommendation set for user u is defined as:

$$L_u^k = \{i \in \mathcal{I} \mid \text{rank}(i; \hat{r}_u) \leq k\} \tag{1}$$

where $k = 10$ in all experiments.

We also define $\mathcal{U}_i^*$ as all users who received item i in their top-k recommendation list:

$$\mathcal{U}_i^* = \{u \in \mathcal{U} \mid i \in L_u^k\} \tag{2}$$

Based on this, we can define the exposure of item i as the proportion of users whose top-k recommendation lists contain the item across all users' recommendation lists: $E_i = |\mathcal{U}_i^*|/|U|$.

Let X_i be the vector of features (described in Section 3.2) derived from the dataset for item i. We aim to train a surrogate model g such that: $E_i \approx g(X_i)$.

The quality of this approximation can be evaluated using the Coefficient of Determination (R^2), which calculates the proportion of the variance accounted for by the model.

3. Approach

3.1. Datasets and Experimental Setup

In this paper, we evaluate three CF models with fundamentally different learning objectives: NCF [18, 19] (nonlinear embedding-based), VAE [20] (generative probabilistic), and BPR [21] (matrix factorization

Table 1
Dataset Statistics

Dataset	Users	Items	Sparsity
Last.fm	24,631	6,584	99%
ML-1M	6,040	3,953	96%

with a pairwise ranking objective)[1], evaluated on two diverse datasets: MovieLens-1M [22] and Last.fm [23]. These differences in learning paradigms naturally produce divergent exposure distributions across models and the datasets differ in content type (movies vs. music), enabling us to assess generalization across domains.

The interaction data is split 80/20 to train and evaluate the recommender. From the trained recommendation model, recommendation lists are generated for all users. The resulting item exposures across all users after training the model are split separately 80/20 at the item level to train and evaluate the surrogate model.

Dataset statistics are shown in Table 1. The code for all stages of the experiments, including data generation, model training, and evaluation, is available in the associated GitHub repository [2].

As our regression model, we employed the XGBoost Regressor [24], an ensemble learning algorithm based on gradient-boosted decision trees. XGBoost was selected for its ability to capture non-linear interactions between diverse feature classes. Furthermore, its built-in feature importance mechanisms let us quantify the contribution of each category to the overall recommendation behavior. Feature importance scores correspond to the gain-based importance metric provided by XGBoost, which measures the average reduction in the loss attributable to each feature across all tree splits. We used a maximum depth of 7, number of estimators ranging from 200 to 500 (varying per model and dataset), and a learning rate of 0.01. Finally, in this study, we use "provider" as the stakeholder associated with an item; for example, a creator or rights holder whose content is distributed by the recommender.

3.2. Features

We incorporate features derived from established literature [15, 16, 17, 13], designed to capture systemic factors that influence recommendation outcomes. The features are in three categories: frequency-based, neighborhood-based, and item-centric.

3.2.1. Frequency-based features

There are two features in this category: item popularity (**POP**) and item-specific profile size (**SZ**). For popularity, we use a standard definition, counting the proportion of users in the dataset who have interacted with item i: $POP(i) = \frac{|U_i|}{|U|}$. For the SZ feature, we are interested in the profile size of users who received the item i in their top-K recommendations, averaging across all such users. If the item has no exposure, SZ is set to zero:

$$SZ(i) = \begin{cases} \frac{1}{|\mathcal{U}_i^*|} \sum_{u \in \mathcal{U}_i^*} |P_u| & \text{if } E_i > 0 \\ 0 & \text{if } E_i = 0 \end{cases} \tag{3}$$

3.2.2. Neighborhood-based features

For each user u, we identify a neighborhood of similar users, denoted as $\mathcal{N}_u$, based on the Jaccard similarity (JS) of their profiles:

$$\mathcal{N}_u = \underset{v \in \mathcal{U} \setminus \{u\}}{\arg \text{ top-}k} \ (\text{JS}(P_u, P_v)) \tag{4}$$

[1]We use the implementations provided in RecBole https://recbole.io
[2]https://github.com/that-recsys-lab/ProviderSideTransparency

Table 2
Feature importance, R^2 for surrogate model. Dashed lines indicate unavailable features for the given dataset.

Dataset	Rec	POP	SZ	YR	GD	ND	RI	AE	INS	R^2
ML-1M	NCF	0.1	0.1	0	0	0.8	0	-	-	0.85
	VAE	0.2	0.1	0	0	0.7	0	-	-	0.93
	BPR	0.5	0.2	0	0.1	0.2	0	-	-	0.83
Last.fm	NCF	0	0.2	0	0	0.4	0.4	0	0	0.87
	VAE	0.5	0.1	0	0	0.3	0.1	0	0	0.87
	BPR	0.7	0.1	0	0	0.1	0.1	0	0	0.88

where P_u and P_v denote the profiles of users u and v, respectively. A profile is defined as the set of items the user has previously interacted with. In the experiments described below, we set $k = 10$.[3]

This calculation enables us to define Neighborhood Density (**ND**) relative to an item: the average frequency with which the target item appears in the histories of a user's nearest neighbors.

$$ND(i) = \begin{cases} \frac{1}{|\mathcal{U}_i^*|} \sum_{u \in \mathcal{U}_i^*} \frac{1}{|\mathcal{N}_u|} \sum_{v \in \mathcal{N}_u} \mathbb{I}(i \in P_v) & \text{if } E_i > 0 \\ 0 & \text{if } E_i = 0 \end{cases} \tag{5}$$

We define a different kind of neighborhood by looking at similarities between items in terms of user overlap. If two items i and j are interacted with the same group of people, they are considered similar:

$$\mathcal{S}_i = \underset{j \in \mathcal{I} \setminus \{i\}}{\arg \text{top-}k} \ (\text{JS}(U_i, U_j)) \tag{6}$$

where U_i and U_j are the set of users who interacted with items i and j respectively. As with ND, we set $k = 10$.[4]

With this similarity metric, we are also able to define the Related Items feature (**RI**), which measures the fraction of items related to the target item within similar users' profiles.

$$RI(i) = \begin{cases} \frac{1}{|\mathcal{U}_i^*|} \sum_{u \in \mathcal{U}_i^*} \frac{1}{|P_u|} \sum_{e \in P_u} \mathbb{I}(e \in \mathcal{S}_i) & \text{if } E_i > 0 \\ 0 & \text{if } E_i = 0 \end{cases} \tag{7}$$

3.2.3. Item-specific features

In both datasets, items are associated with genres. If a user shows a propensity to a certain genre or category, we might expect that items of that genre would be more likely to be recommended to them. We therefore define Genre Density (**GD**) as the average number of items within the target item's genre present in the profiles of users in $\mathcal{U}_i^*$.

$$GD(i) = \begin{cases} \frac{1}{|\mathcal{U}_i^*|} \sum_{u \in \mathcal{U}_i^*} \sum_{j \in P_u} \mathbb{I}(G(j) = G(i)) & \text{if } E_i > 0 \\ 0 & \text{if } E_i = 0 \end{cases} \tag{8}$$

where $G(i)$ is the genre (or set of genres) associated with item i and $\mathbb{I}(\cdot)$ is an indicator function.

We have three other features, drawn from the item metadata:

- **YR**: The release year of the target item, utilized as a temporal indicator of the item's age.
- **AE** (Last.fm only): Audio energy is a high-level acoustic feature representing the perceived intensity and activity of the track, ranging from 0 to 1.
- **INS** (Last.fm only): Instrumentalness represents the likelihood that a track contains no vocal content, ranging from 0 to 1.

[3] We experimented with varying neighborhood sizes (k=10,20,30,40) and found that the results are not sensitive to this choice.
[4] The results in our experiments were found not to be sensitive to choice of k.

4. Results

Table 2 reports the regression model's performance across all model-dataset combinations. The surrogate achieves high fidelity in all settings, with R^2 scores ranging from 0.83 to 0.93, demonstrating that structural features effectively capture the global exposure behavior of all three recommenders across both domains. Despite this overall consistency, the contribution of individual features varies substantially across models and domains, reflecting their different learning objectives and optimization strategies.

ML-1M dataset: On the ML-1M dataset, Neighborhood Density (ND) is the dominant feature for both NCF and VAE ($ND = 0.7 - 0.8$), indicating that items appearing frequently in the histories of a user's nearest neighbors are strongly favored by both models. This suggests that both NCF and VAE heavily rely on neighborhood-based features when distributing exposure on the movie domain. In contrast, BPR exhibits a notably different pattern. The popularity of the item ($POP = 0.5$) becomes the dominant factor, followed by the density of neighborhoods and profile size features ($ND = 0.2, SZ = 0.2$). This reflects BPR's pairwise ranking objective, which is more sensitive to globally popular items than to localized neighborhood structure.[5]

Last.fm Results: Compared to ML-1M, the Related Items feature, which was negligible on ML-1M, appears to play a somewhat more notable role on Last.fm, highlighting the impact of item similarity within user profiles on the global behavior of the model in the music domain. For NCF, the contribution is distributed across multiple features: Related Items ($RI = 0.4$) and Neighborhood Density ($ND = 0.4$) are the dominant factors, with average profile size contributing to a lesser extent and popularity contributing negligibly. This indicates that in the music domain, NCF relies more heavily on item similarity and collaborative neighborhood structure than on global popularity signals, favoring songs that are similar to those already present in the user's listening history and that appear frequently in the profiles of similar users. VAE, on the other hand, exhibits a strong popularity bias ($POP = 0.5$), making item popularity the dominant driver of exposure with a smaller but non-negligible contribution from neighborhood density ($ND = 0.3$). Finally, BPR exhibits a similar pattern to its behavior on ML-1M. Popularity remains the dominant factor ($POP = 0.7$), followed by ND and RI (ND=0.1, RI=0.1).

Comparing across datasets, several patterns emerge. First, Neighborhood Density plays a strong role across nearly all model-dataset combinations, confirming the importance of neighborhood-based signals regardless of domain. Second, the Related Items feature shows a clear domain dependency: negligible on ML-1M, it emerges as a significant factor on Last.fm, indicating that item similarity within user profiles carries more weight in the music domain than in the movie domain. This may be due to movie viewers potentially having broader genre preferences, whereas music fans tend to focus more on particular genres. This would lead the RI feature to be more predictive for music. Third, BPR is strongly influenced by item popularity across both datasets, suggesting its global behavior is largely driven by popularity bias. Finally, the acoustic features AE and INS, available only in Last.fm, contribute minimally across all models. It is possible that the information in these features is subsumed by the GD feature.

Figure 1 presents the results of a backward feature elimination study on the Last.fm dataset across three recommender system models, evaluated using R^2. At rank 0, all models begin with their full feature set, achieving $R^2 \approx 0.87$-0.88. Features are then removed iteratively in order of least importance, and the resulting R^2 is recorded at each step. The curve remains largely flat from ranks 0 to 4, indicating that the early-removed features, INS, AE, GD, and YR, contribute minimally to the model's predictive power across all three models. A gradual decline begins at ranks 5 to 7, where the removal of POP, SZ, and RI introduces moderate performance degradation. The most striking observation occurs at rank 8, where removing the final remaining feature causes a sharp collapse to near zero for all models. This reveals that a single feature alone, RI for NCF and POP for both VAE and BPR, is sufficient to explain 0.6-0.7 of the variance in item exposure.

A key motivation of this work is to provide transparency to item providers. However, it is important to acknowledge that not all the factors identified by our approach are directly actionable by the providers;

[5] Popularity bias is a well-known characteristic of BPR; see [25].

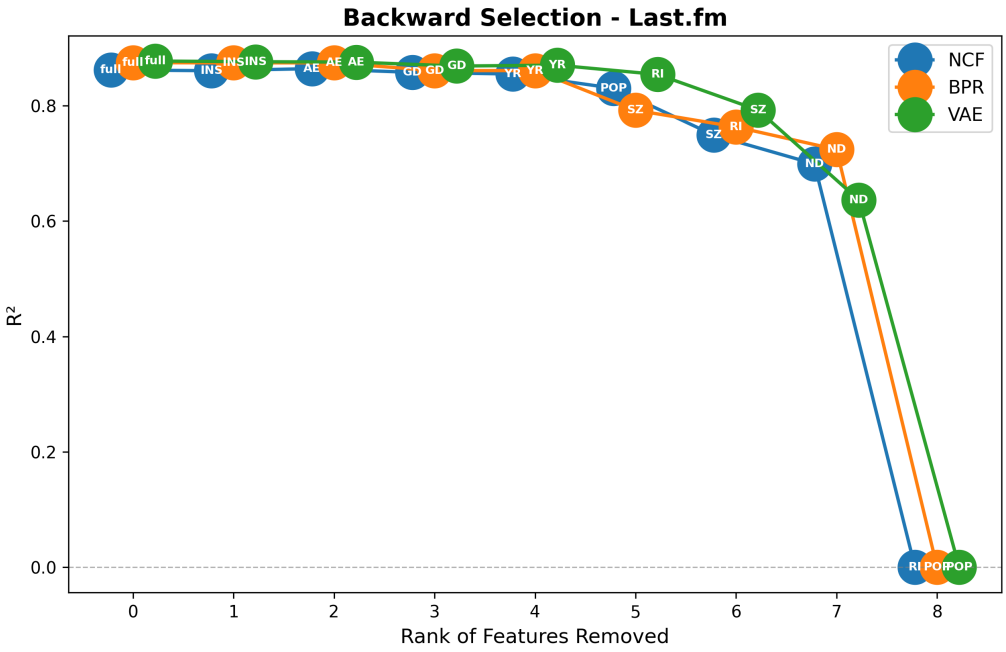

Figure 1: Backward feature elimination results on the Last.fm dataset for NCF, BPR, and VAE models, measured by R^2. Each node label indicates the feature removed at that step; rank 0 represents the full feature set.

in this sense, the interpretations we provide are diagnostic rather than prescriptive. Table 3 provides an analysis of the actionability of our feature types from the provider and platform perspectives. However, diagnostic or informative transparency is itself a meaningful and valuable form of explanation [26]. From the provider's perspective, we think it is useful to understand that low exposure may stem from structural dataset properties, such as low neighborhood density or low popularity, rather than item quality alone. This perspective helps providers set realistic expectations, reduces their uncertainty, and replaces speculative folk theories [9] with evidence-based insight [26]. From the platform designer's perspective, these findings may be directly actionable: engineers can use this analysis to identify and mitigate structural biases in their recommendation algorithms, for example, by designing re-ranking strategies or post-hoc models to reduce the dominance of neighborhood density or popularity [27, 28].

A potential methodological concern is whether the conditioning of features SZ, GD, ND, and RI on $\mathcal{U}_i^*$ introduces circularity. We argue it does not. This conditioning is not incidental but reflects the core explanatory goal of our framework: we seek to characterize the structural properties of recommended items as experienced by the users who actually received them as recommendations. The question we ask is descriptive rather than counterfactual (what properties characterize an item's observed exposure pattern), making conditioning on $\mathcal{U}_i^*$ a deliberate design choice aligned with this post-hoc goal. In addition, we note that Table 2 argues against circularity: GD contributes negligibly across all models and datasets, and RI contributes zero on ML-1M, despite sharing the same conditioning as ND. Meanwhile, POP, entirely independent of $\mathcal{U}_i^*$, dominates for BPR across both datasets.

5. Conclusion

We proposed a surrogate modeling framework to explain the black-box recommender systems from the perspective of item providers. Rather than explaining individual user-item pairs, we trained an interpretable proxy model on a set of structural features, including neighborhood density, popularity, profile size, and item similarity, to approximate the global exposure distribution produced by three recommendation models across two domains. Our results demonstrate consistently high surrogate

Table 3
Actionability of features by provider and platform.

Feature	Actionable by provider?	Actionable by platform?
YR, AE, INS	✓ Yes (item properties)	✗ No
POP	✗ No	✓ Yes (re-ranking)
ND, RI, SZ, GD	✗ No	✓ Yes (algorithm design)

fidelity ($R^2 = 0.83 - 0.93$), confirming that structural dataset properties are reliable predictors of recommendation exposure. Feature importance analysis reveals that the dominant drivers vary meaningfully across models and domains: neighborhood density leads exposure for NCF and VAE on ML-1M, popularity dominates BPR across both datasets, and item similarity plays a notably larger role in the music domain. Acoustic content features contribute minimally.

While not all identified factors are directly actionable by providers, we argue that this distinction does not diminish their value in contributing to explanations. Additional human-centered research is required to understand how the insights available through a global model of this type may be presented as explanations to providers, and to understand how providers would make use of such information.

6. Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (OpenAI) for grammar and spelling checks, paraphrasing and rewording, improving writing style, and programming assistance. After using this tool, the authors reviewed and edited the generated content and code as needed, validated the resulting implementation, and take full responsibility for the publication's content and results.

References

[1] N. Tintarev, J. Masthoff, Beyond explaining single item recommendations, in: Recommender Systems Handbook, Springer, 2022, pp. 711–756.

[2] N. Tintarev, J. Masthoff, Explaining recommendations: Design and evaluation, in: Recommender systems handbook, Springer, 2015, pp. 353–382.

[3] T. Chowdhury, R. Rahimi, J. Allan, Equi-explanation maps: concise and informative global summary explanations, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 464–472.

[4] M. Varasteh, E. McKinnie, A. Aird, D. Acuña, R. Burke, Comparative explanations for recommendation: Research directions, in: Proceedings of the 11th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS 2024), 2024, pp. 1–14. URL: https://ceur-ws.org/Vol-3815/paper1.pdf.

[5] O. Barkan, Y. Schein, Y. Elisha, V. Bogina, M. Baklanov, N. Koenigstein, Fidelity-aware recommendation explanations via stochastic path integration, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, 2026, pp. 14484–14492.

[6] S. Verma, A. Singh, V. Boonsanong, J. P. Dickerson, C. Shah, Recrec: Algorithmic recourse for recommender systems, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 4325–4329.

[7] A. Ghazimatin, O. Balalau, R. Saha Roy, G. Weikum, Prince: Provider-side interpretability with counterfactual explanations in recommender systems, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 196–204.

[8] M. Baklanov, V. Bogina, Y. Elisha, Y. Schein, L. Allerhand, O. Barkan, N. Koenigstein, Refining fidelity metrics for explainable recommendations, in: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2025, pp. 2967–2971.

[9] M. A. DeVito, D. Gergle, J. Birnholtz, " algorithms ruin everything" # riptwitter, folk theories, and resistance to algorithmic change in social media, in: Proceedings of the 2017 CHI conference on human factors in computing systems, 2017, pp. 3163–3174.
[10] K. Dinnissen, C. Bauer, Amplifying artists' voices: Item provider perspectives on influence and fairness of music streaming platforms, in: Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization, 2023, pp. 238–249.
[11] Z. Michalewicz, K. Dinnissen, E. Herder, H. Hauptmann, (de) composing the algorithm: Explaining music recommender systems to artists for understanding, transparency, and empowerment, in: CEUR Workshop Proceedings, volume 4027, CEUR WS, 2025.
[12] K. Dinnissen, I. Saccardi, M. Vredenborg, C. Bauer, Looking at the faccts: Exploring music industry professionals' perspectives on music streaming services and recommendations, in: Proceedings of the 2nd International Conference of the ACM Greek SIGCHI Chapter, 2023, pp. 1–5.
[13] H. Abdollahpouri, R. Burke, B. Mobasher, Managing popularity bias in recommender systems with personalized re-ranking, arXiv preprint arXiv:1901.07555 (2019).
[14] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Multistakeholder recommendation: Survey and research directions., User Modeling and User-Adapted Interaction 30 (2020) 127–158.
[15] S. Sahebi, P. Brusilovsky, Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation, in: International Conference on User Modeling, Adaptation, and Personalization, Springer, 2013, pp. 289–295.
[16] F. Eskandanian, N. Sonboli, B. Mobasher, Power of the few: Analyzing the impact of influential users in collaborative recommender systems, in: Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, 2019, pp. 225–233.
[17] A. M. Rashid, G. Karypis, J. Riedl, Influence in ratings-based recommender systems: An algorithm-independent approach, in: Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 556–560.
[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 173–182.
[19] M. Varasteh, M. S. Nejad, H. Moradi, M. A. Sadeghi, A. Kalhor, An improved hybrid recommender system: Integrating document context-based and behavior-based methods, in: 2023 31st International Conference on Electrical Engineering (ICEE), IEEE, 2023, pp. 881–887.
[20] D. Liang, R. G. Krishnan, M. D. Hoffman, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the 2018 world wide web conference, 2018, pp. 689–698.
[21] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, arXiv preprint arXiv:1205.2618 (2012).
[22] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, Acm transactions on interactive intelligent systems (tiis) 5 (2015) 1–19.
[23] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, P. Lamere, The million song dataset, In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011) (2011).
[24] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[25] D. Jannach, L. Lerche, I. Kamehkhosh, M. Jugovac, What recommenders recommend: an analysis of recommendation biases and possible countermeasures, User Modeling and User-Adapted Interaction 25 (2015) 427–491.
[26] N. Sonboli, J. J. Smith, F. Cabral Berenfus, R. Burke, C. Fiesler, Fairness and transparency in recommendation: The users' perspective, in: Proceedings of the 29th ACM conference on user modeling, adaptation and personalization, 2021, pp. 274–279.
[27] A. Aird, E. Štefancová, A. Buhayh, C. All, M. Homola, N. Mattei, R. Burke, Integrating individual and group fairness for recommender systems through social choice, in: Proceedings of the Nineteenth ACM Conference on Recommender Systems, 2025, pp. 177–186.
[28] C. Xu, S. Chen, J. Xu, W. Shen, X. Zhang, G. Wang, Z. Dong, P-mmf: Provider max-min fairness re-ranking in recommender system, in: Proceedings of the ACM Web Conference 2023, 2023, pp.

3701–3711.