The Forgotten Users of Recommender Systems

Sam Pieters[1,*], Basil Rommens[1], Noah Daniëls[1] and Bart Goethals[1,2]

[1]University of Antwerp, Antwerp, Belgium
[2]Froomle, Antwerp, Belgium

Abstract

Offline evaluation of recommender systems is dominated by aggregate ranking metrics such as NDCG@K, Recall@K, and HitRate@K. While these metrics provide a convenient basis for comparing algorithms, they reduce the performance of an entire user population to a single average and therefore reveal little about how recommendation quality is distributed across individual users. This limitation is particularly relevant given the well-known discrepancy between offline evaluation and online performance, suggesting that aggregate metrics fail to capture important aspects of user experience. In this paper, we argue that recommender evaluation should move beyond aggregate metrics by explicitly analyzing recommendation quality at the user level. As a first step towards such a methodology, we introduce the notion of **forgotten users**: users for whom a personalized recommender consistently fails to provide good recommendations and even performs worse than a simple popularity-based recommender across repeated evaluation splits.

Through experiments on four benchmark datasets and multiple representative recommendation algorithms, we show that a substantial fraction of users are repeatedly forgotten, including many with long interaction histories. Moreover, these users persist across different algorithms, datasets and evaluation splits, indicating that they are not merely a consequence of random evaluation artifacts. Our findings demonstrate that strong average performance can coexist with systematic failures for individual users, motivating a more fine-grained evaluation methodology that complements traditional ranking metrics with analyses of how recommendation quality is distributed across the user population.

Keywords

recommender systems, ranking metrics, forgotten users

1. Introduction

Recommender systems are ultimately designed to help individual users discover relevant content. Yet, despite this inherently user-centric objective, recommender systems are almost exclusively evaluated using aggregate ranking metrics such as NDCG@K, Recall@K, and HitRate@K. These metrics summarize the performance of an algorithm as a single number by averaging over all users, making it straightforward to compare competing approaches [1, 2]. While this methodology has served the community well, it provides only a limited view of recommendation quality. Two algorithms with nearly identical aggregate metrics may behave very differently for individual users. Likewise, the well-known discrepancy between offline evaluation and online performance suggests that current evaluation protocols overlook important aspects of recommendation quality. Understanding why this discrepancy exists requires looking beyond averages and examining how recommendations are distributed across users rather than considering only their mean performance. One possibility is that aggregate metrics conceal systematic failures affecting subsets of users. An algorithm may achieve an excellent average NDCG@K while repeatedly failing to retrieve relevant items for the same users. From the perspective of these users, the recommender effectively does not work, yet these failures remain almost invisible in conventional evaluation. We refer to these users as **forgotten users**. The goal of this paper is not merely to introduce a new user category, but to advocate a more fine-grained perspective on recommender evaluation.

FRAME'26: Methodology First - Rethinking Research Assessment in RecSys Workshop, September 28, 2026, Minneapolis, Minnesota, USA

*Corresponding author.
sam.pieters@uantwerpen.be (S. Pieters); noah.daniels@uantwerpen.be (N. Daniëls); bart.goethals@uantwerpen.be (B. Goethals)
https://www.bartgoethals.com/ (B. Goethals)
0009-0001-7008-3847 (S. Pieters); 0009-0001-7778-4769 (N. Daniëls); 0000-0001-9327-9554 (B. Goethals)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-26

CEUR-WS.org/Vol-4341/paper4.pdf

Rather than asking only *which algorithm achieves the highest average score, we argue that we should also ask which users actually receive useful recommendations, which users do not, and whether these failures are systematic.* We believe that answering these questions is an important step towards understanding the gap between offline evaluation and real-world recommender performance. As a first step towards this broader evaluation methodology, we study forgotten users across multiple benchmark datasets and recommendation algorithms. We define forgotten users as users who consistently receive no useful recommendations across repeated test splits, thereby reducing the influence of random fold-in/held-out partitions. Our experiments reveal that forgotten users are not limited to cold-start scenarios [3]: many have substantial interaction histories and remain forgotten across multiple recommendation algorithms. Furthermore, these users are not concentrated in a specific region of the user space, suggesting that they represent a more fundamental limitation of current recommendation approaches. The contributions of this paper are threefold:

We argue that aggregate offline evaluation provides only a limited view of recommendation quality and advocate complementing average ranking metrics with user-level analyses. We introduce the notion of forgotten users and propose a robust methodology for identifying them across repeated splits. We demonstrate that forgotten users occur consistently across multiple benchmark datasets and recommendation algorithms, revealing systematic failures that remain hidden behind aggregate evaluation metrics. The remainder of this paper addresses the following research questions:

- **RQ1:** How prevalent are forgotten users across different recommendation datasets?
- **RQ2:** To what extent do different recommendation algorithms forget the same users?
- **RQ3:** Can combining multiple recommendation algorithms eliminate forgotten users?

2. Related Work

2.1. Cold-start and data sparsity

A well-studied challenge in recommender systems is the *cold-start problem*, where users with few or no interactions are difficult to serve [4, 3]. Numerous methods have been proposed to mitigate this issue, ranging from hybrid recommenders to meta-learning approaches [5, 6]. While cold-start users naturally fall into the category of forgotten users, our focus lies beyond this already well-explored area. We specifically study users with sufficient interaction history who nonetheless receive no relevant recommendations, highlighting a different source of disparity.

2.2. Fairness and bias in recommendation

More recent research has examined issues of fairness, bias, and underrepresentation in recommender systems. Work in this area has considered fairness across users, items, or groups defined by sensitive attributes [7, 8]. These studies demonstrate that recommendation quality is not always evenly distributed. Our work is aligned with this line of research, but we approach fairness from a user-centric perspective: rather than focusing on predefined groups, we identify forgotten users directly based on their experienced utility (or lack thereof) from the recommender.

2.3. Beyond aggregate metrics

Several researchers have argued that evaluation in recommender systems should go beyond aggregate metrics. Coverage-based metrics [9, 10] and long-tail evaluation [11] have been proposed to capture diversity in what is recommended. More recently, work on robustness and worst-case evaluation emphasizes that good average performance can coexist with very poor performance for specific users or items [12, 13]. Our notion of forgotten users extends this perspective by explicitly identifying the set of users for whom the recommender system fails entirely, providing a direct measure of worst-case user experience.

3. Identifying forgotten users

3.1. Identifying forgotten users with hitrate

We first define the notion of a *hit* for a user. Given a user u, a *hit* occurs when a recommended item appears among u's relevant items in the test set. The top-k recommendations produced by a model for user u are denoted $\hat{R}_u^k$, and the set of relevant items for u is R_u. We define a hit as any item in $\hat{R}_u^k$ that is also in R_u. Formally, the number of hits for user u can be written as:

$$s_u^{\text{model}} = |R_u \cap \hat{R}_u^k|.$$

HitRate@k is the user-level metric that measures the proportion of users who receive at least one hit among their top-k recommendations:

$$\text{HitRate@k}(u) = \begin{cases} 1 & \text{if } s_u^{\text{model}} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

3.2. Robust identification of forgotten users

Evaluating a user on a single test split may incorrectly classify them as forgotten users. Random variation in the split can cause a user to appear forgotten in one instance, even if they are generally well-covered. To mitigate coincidental misclassification, we evaluate each user across multiple independently sampled holdout splits while keeping the training set fixed. This repeated holdout protocol ensures that the recommender model remains unchanged across evaluations, allowing differences in user performance to be attributed solely to variation in the sampled test interactions rather than changes in the trained model. Rather than holding out a fixed percentage of each user's interactions, we hold out a fixed number of interactions per user. This ensures that all users are evaluated on an equal number of relevant test items, making the resulting hit counts easier to interpret and compare across different history lengths. This repeated splitting strategy reduces the impact of random variation in the data and allows us to assess the stability of forgotten users across different evaluation scenarios, following common practice in recommender system evaluation [14]. A visual aid for our splitting strategy is given in Figure 1.

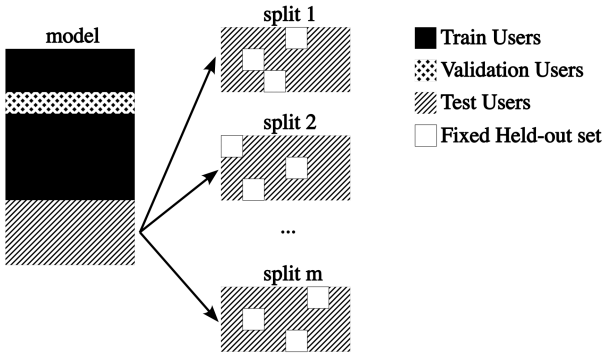


Figure 1: Splitting strategy used in the experiments

A user is classified as forgotten users only if they are consistently forgotten across these splits, ensuring robustness and stability in their classification.

3.3. Popularity baseline for comparison

To distinguish regular users from forgotten users, we compare each recommendation algorithm against a non-personalized popularity-based recommender. A popularity recommender reflects a realistic

baseline that is commonly used in recommender system evaluation and provides a stronger point of comparison for personalized methods.

We execute the popularity algorithm using the same training data and evaluation protocol as the personalized recommender. This ensures that both algorithms are evaluated under identical conditions. Since the popularity ranking depends on the item popularity distribution of the training data, deriving a closed-form theoretical expectation for every user would require modeling the popularity of all candidate items after excluding the user's observed interactions. Instead, we directly obtain the baseline by evaluating the popularity recommender on the same splits.

For each user, we record the number of successful recommendations (hits) across all evaluation splits for both the popularity recommender and the personalized algorithm. The difference between these values is defined as

$$\Delta_u = \text{Hits algorithm}_u - \text{Hits popularity}_u.$$

A negative value ($\Delta_u < 0$) indicates that the popularity recommender retrieves relevant items more frequently than the personalized algorithm for user u. We therefore classify these users as forgotten users, as they benefit less from personalization than from a simple non-personalized strategy. Conversely, users with $\Delta_u \geq 0$ are not generally considered forgotten, as the personalized recommender performs at least as well as the popularity baseline. A $\Delta_u = 0$ does not necessarily indicate a forgotten user, as both methods may achieve the same number of successful recommendations. Only when $\Delta_u = 0$ and both recommenders obtain zero hits do we classify the user as forgotten users, since neither strategy is able to retrieve any relevant items for all splits.

Using a popularity baseline offers two advantages. First, it provides a practical and reproducible reference that reflects a commonly adopted recommendation strategy. Second, it identifies users for whom sophisticated personalized models fail to outperform a simple non-personalized popularity-based recommender. If personalization cannot consistently exceed this non-personalized baseline, it suggests that the model is not effectively exploiting the user's individual preferences, indicating users who are systematically forgotten despite the additional complexity of personalized recommendation.

While cold-start users are expected to perform poorly under personalized recommendation, our analysis extends beyond this well-studied setting. We specifically investigate users with sufficient interaction history whose recommendation quality remains below that of the popularity baseline. This allows us to identify users who are forgotten despite having enough historical information for personalization, revealing disparities that are not captured by traditional cold-start analyses.

3.4. Greedy aggregation approach

Besides evaluating individual recommenders, we also consider a greedy aggregation strategy that combines their recommendations. The greedy approach iteratively selects recommendations from the available models to maximize the number of relevant items. Although this represents an optimistic ensemble rather than a practical recommender, it allows us to assess whether combining multiple algorithms can eliminate forgotten users. If users remain forgotten even in this setting, it suggests that the problem is not specific to any single recommendation algorithm.

4. Experimental Setup

4.1. Datasets & Algorithms

To evaluate the generality and robustness of our findings, we conduct experiments on four widely used recommender system datasets: *MovieLens [15]*, *Amazon [16]*, *CiteULike [17]*, and *Netflix [18]*. These datasets cover diverse domains, activity levels, and sparsity patterns. Furthermore, they allow comparison with established cold-start definitions, as prior work provides dataset-specific thresholds. We adopt the highest reported cold-start threshold for each dataset to ensure a conservative and consistent evaluation.

Cold-start users are identified using dataset-specific interaction thresholds chosen based on commonly used definitions in previous work and the sparsity characteristics of each dataset. The thresholds used in our experiments are summarized in Table 1.

Dataset	Cold-start threshold	Reference
MovieLens-1M	< 20 interactions	[19, 20, 21]
Amazon PrimePantry	< 5 interactions[1]	–
CiteULike	< 10 interactions	[22, 23]
Netflix	< 20 interactions	[20, 21]

Table 1
Cold-start user definitions used for each dataset.

These thresholds are used solely to distinguish cold-start users in the visualizations and analyses. Users with interaction histories exceeding the corresponding threshold are considered non-cold-start users.

We evaluate forgotten users using representative algorithms from commonly studied recommendation paradigms: *ItemKNN*, *EASE*, and *SVD*. The experiments are implemented using the *RecPack* recommendation framework [24], which we extended with additional dataset loaders, evaluation procedures, and analysis utilities required for this work. To facilitate reproducibility, the complete source code and experimental pipeline are publicly available on GitHub:[2].

4.2. Data splitting stategy

4.2.1. Preprocessing

For the explicit feedback datasets *Amazon*, *MovieLens*, and *Netflix*, we treat a rating higher than three as an interaction according to common practice [25, 26, 27]. For all datasets, we remove duplicates and incomplete entries, as well as all features other than the user ID and item ID. Following previous works, we perform 5-core filtering, which means that we prune the data to ensure that all users and items contain at least five interactions [28, 29, 30].

4.2.2. Data Splitting

For each dataset, we generate a train–test split where 80% of the users are used for training and the remaining 20% are reserved for testing. The training portion is further divided into training and validation sets for model selection and hyperparameter tuning.

Each test set is further partitioned into 20 independent fold-in/held-out configurations. The corresponding training and validation sets remain fixed across all configurations, ensuring that the recommendation model and its selected hyperparameters are identical for every fold-in/held-out split. Consequently, any differences in recommendation performance can be attributed to the sampled fold-in/held-out interactions rather than to variations in model training. For each of these configurations, a fixed number of five interacted items is selected as the held-out set, while all remaining test interactions are used as fold-in interactions (history items). We chose a fixed held-out size of five interactions to ensure that every user is evaluated against an equal amount of relevant items. Using fewer than five interactions could result in users having only a small number of test items, making them more likely to appear as forgotten users simply due to having fewer opportunities to receive hits rather than due to poor recommendation performance.

At the same time, a held-out size of five interactions allows the evaluation protocol to naturally include users with no available interaction history after the fold-in split. These cases represent a

[1]No dataset-specific cold-start threshold was identified in the literature for Amazon Prime Pantry. Therefore, we follow commonly used definitions that classify approximately 10% of users as cold-start users and select the corresponding interaction threshold. For this dataset, this corresponds to users with fewer than 5 interactions.

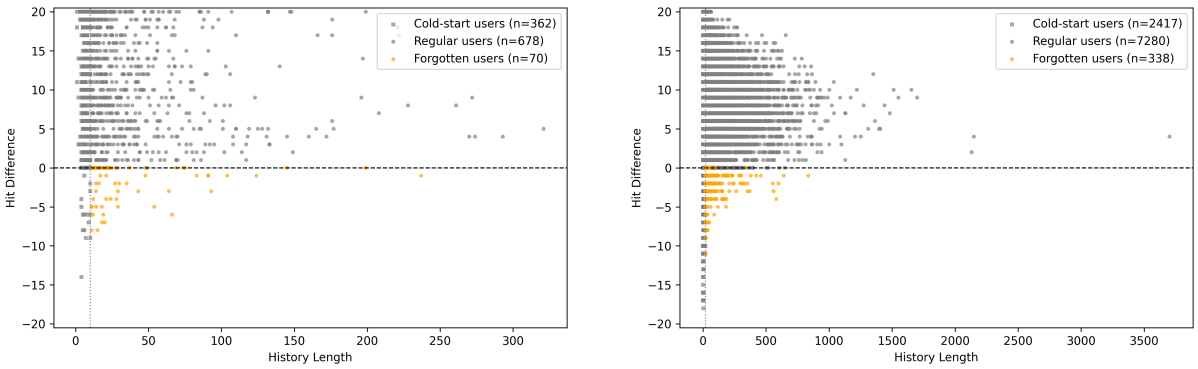
[2]https://github.com/sampieters/forgotten-users

pure cold-start scenario, where the recommendation model must generate predictions without any user-specific historical information. Therefore, this setup allows us to distinguish performance issues caused by limited user history from failures occurring for users with sufficient historical information.

4.3. Visualization of forgotten users

To visualize the distribution of forgotten users, we first compute, for each user, the total number of hits obtained by both the recommendation model and the popularity baseline across all repeated holdout splits. We then compute the difference between these two quantities and plot it against the user's interaction history length. Values below zero indicate that the personalized model achieves fewer successful recommendations than the popularity baseline, classifying the user as forgotten. A value equal to zero indicates an equal number of successful recommendations for both models and is considered forgotten only when both models achieve at least one successful recommendation. Values above zero indicate that the personalized model outperforms the popularity baseline.

To distinguish cold-start users from users with longer interaction histories, the plots additionally include a vertical line indicating the cold-start threshold, where the cutoff is defined as described in Section 4.2.1. An example is shown in Figure 2.



(a) ItemKNN for CiteULike

(b) EASE for Netflix

Figure 2: Forgotten users across different datasets and recommendation algorithms.

4.4. Analyzing forgotten users histories

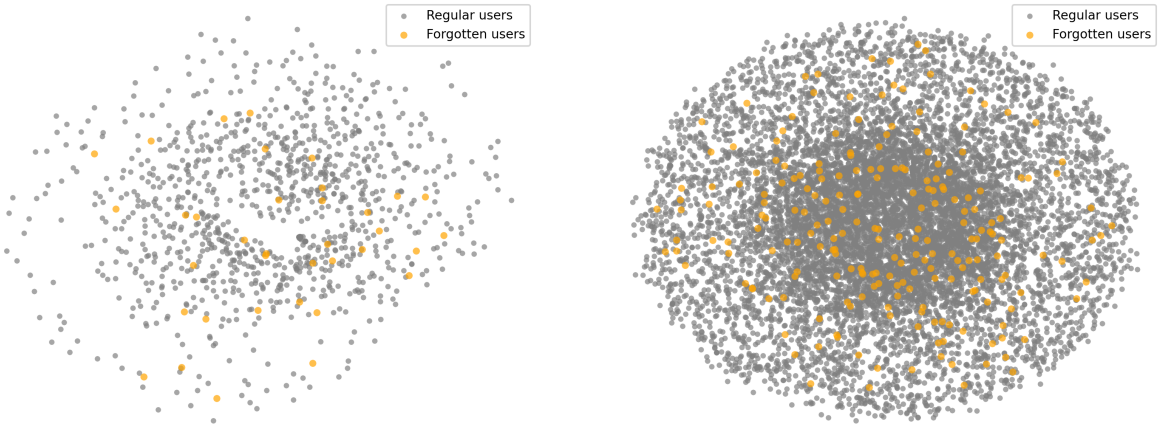
To better understand the characteristics of forgotten users, we analyze their interaction histories using Uniform Manifold Approximation and Projection (UMAP) [31]. Each user is represented by their interaction vector, and UMAP is applied to project these high-dimensional user profiles into a two-dimensional space.

By plotting all users and highlighting the subset of forgotten users, we can visually inspect whether these users share similar interaction patterns or occupy distinct regions of the user space. The resulting visualization, shown in Figure 3, reveals that the forgotten users are widely scattered across the projection.

This dispersion indicates that forgotten users do not form a single coherent cluster or share a common preference structure. Instead, they are distributed throughout the entire user space, suggesting that their histories are highly diverse. Consequently, the occurrence of forgotten users cannot be attributed to a specific user type or behavioral pattern.

5. Results and Discussion

To answer our research questions, we measure how often users are classified as forgotten relative to the popularity baseline. We further compare these patterns across multiple datasets and recommendation



(a) ItemKNN for CiteULike (b) EASE for Netflix

Figure 3: UMAP projections of user histories, highlighting forgotten users identified by different recommendation algorithms on different datasets.

algorithms.

Dataset	ItemKNN	EASE	SVD	Average	Greedy
ML-1M	13.66%	3.56%	4.64%	7.28%	0.58%
Amazon PrimePantry	53.58%	38.91%	41%	44.50%	17.32%
CiteULike	6.3%	3.87%	6.57%	5.58%	1.35%
Netflix	19.72%	3.37%	5.6%	9.56%	1.37%

Table 2
Percentage of forgotten users for different datasets and recommender algorithms.

Table 2 provides a summary of these experiments. For each dataset-algorithm pair, we report the ratio of forgotten users. Our results reveal that a non-negligible portion of users are consistently forgotten users.

Dataset	ItemKNN	EASE	SVD	Average	Greedy
ML-1M	0.41%	0.00%	0.00%	0.14%	0.00%
Amazon PrimePantry	38.54%	36.76%	37.29%	37.53%	16.78%
CiteULike	3.06%	1.80%	4.68%	3.18%	0.81%
Netflix	4.68%	1.22%	1.89%	2.60%	1.00%

Table 3
Percentage of users for whom both the recommender algorithm and the popularity baseline fail to retrieve a relevant item in the top-K recommendations.

Table 3 provides a further breakdown of the forgotten users reported in Table 2. Specifically, it identifies the subset of forgotten users that receive no relevant recommendations from either the evaluated recommender or the popularity baseline across the validation splits. This subset represents a more severe failure case where both recommendation strategies consistently fail to retrieve any relevant items, as they are not served by either the personalized recommender or the non-personalized baseline.

5.1. Beyond cold-start users

To distinguish cold-start users from forgotten users, we analyze users based on their interaction history length. While sparse histories increase difficulty, many forgotten users have moderate or extensive

histories and remain forgotten across models and splits. This shows that data sparsity alone cannot explain the issue, suggesting deeper limitations in how recommender systems represent diverse user preferences.

5.2. Diversity among forgotten users

To investigate why users remain forgotten, we apply UMAP to user interaction profiles. The projection shows that forgotten users are dispersed throughout the embedding space rather than forming a distinct cluster. This indicates that they do not share a single behavioral pattern, suggesting that forgetting a user is a broader limitation in model generalization rather than a problem specific to one user group.

5.3. Limitations of offline evaluation

As with offline recommender system evaluation in general, the identification of forgotten users should be interpreted with caution. Offline evaluation typically assumes that unobserved interactions represent negative feedback, while in practice, they may simply correspond to items that were never exposed to the user. Therefore, some observed forgotten users may be influenced by missing exposure information rather than solely reflecting a mismatch between user preferences and model representations.

6. Conclusion

Offline evaluation of recommender systems has traditionally focused on aggregate ranking metrics that summarize recommendation quality as a single average across users. While these metrics have become the standard for comparing recommendation algorithms, they provide only a partial view of recommender performance. By reducing the behavior of an entire user population to a single number, they can conceal systematic failures experienced by individual users. In this paper, we argued that recommender evaluation should complement aggregate metrics with analyses at the level of individual users. As a first step towards such a methodology, we introduced the notion of *forgotten users*: users for whom personalization consistently fails and performs worse than a simple popularity-based recommender across repeated evaluation splits. By comparing against a popularity baseline, our approach identifies users who derive little or no benefit from personalization, despite the additional complexity of modern recommendation algorithms. Our experiments demonstrate that forgotten users are present across multiple benchmark datasets and recommendation algorithms. Importantly, they are not confined to cold-start scenarios, nor do they disappear when combining multiple recommendation techniques. These findings suggest that forgotten users are not simply artifacts of data sparsity or a particular algorithm, but expose blind spots that remain largely invisible when evaluation relies solely on average ranking metrics. More broadly, our findings illustrate that understanding recommender performance requires looking beyond aggregate ranking metrics. Aggregate scores reveal how well an algorithm performs on average, but they provide little insight into who actually benefits from personalization and who is consistently left behind. Examining the distribution of recommendation quality across users provides a complementary perspective that can reveal weaknesses that remain invisible in traditional offline evaluation. The forgotten users identified in this work are not merely a characteristic of recommender systems. They expose a limitation of how recommender systems are currently evaluated. As the community continues to seek a better understanding of the gap between offline evaluation and online performance, we believe that future evaluation methodologies should move beyond reporting a single average number and instead characterize how recommendation quality is distributed across individual users. We hope this work encourages the development of richer, more user-centric evaluation methodologies that provide deeper insight into recommender behavior and ultimately lead to more effective personalization.

7. Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT solely to assist with writing the paper, rephrasing, and improving the language of the text. Generative AI was not used for any other purpose, including research design, data collection, implementation, experiments, analysis, or the generation of results and figures. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] Y.-M. Tamm, R. Damdinov, A. Vasilev, Quality metrics in recommender systems: Do we calculate metrics consistently?, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 708–713. URL: https://doi.org/10.1145/3460231.3478848. doi:10.1145/3460231.3478848.

[2] D. Valcarce, A. Bellogín, J. Parapar, P. Castells, Assessing ranking metrics in top-N recommendation, Information Retrieval 23 (2020) 411–448. URL: https://doi.org/10.1007/s10791-020-09377-x. doi:10.1007/s10791-020-09377-x.

[3] H. Yuan, A. A. Hernandez, User cold start problem in recommendation systems: A systematic review, IEEE Access 11 (2023) 136958–136977. doi:10.1109/ACCESS.2023.3338705.

[4] D. K. Panda, S. Ray, Approaches and algorithms to mitigate cold start problems in recommender systems: a systematic literature review, Journal of Intelligent Information Systems 59 (2022) 341–366. URL: https://doi.org/10.1007/s10844-022-00698-5. doi:10.1007/s10844-022-00698-5.

[5] N. Khaledian, A. Nazari, M. Barkhan, CFCAI: improving collaborative filtering for solving cold start issues with clustering technique in the recommender systems, Multimedia Tools and Applications 84 (2025) 34207–34228. URL: https://doi.org/10.1007/s11042-024-20579-z. doi:10.1007/s11042-024-20579-z.

[6] D.-K. Chae, J. Kim, D. H. Chau, S.-W. Kim, Ar-cf: Augmenting virtual users and items in collaborative filtering for addressing cold-start problems, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1251–1260. URL: https://doi.org/10.1145/3397271.3401038. doi:10.1145/3397271.3401038.

[7] Y. Wang, W. Ma, M. Zhang, Y. Liu, S. Ma, A survey on the fairness of recommender systems, ACM Trans. Inf. Syst. 41 (2023). URL: https://doi.org/10.1145/3547333. doi:10.1145/3547333.

[8] Y. Zhao, Y. Wang, Y. Liu, X. Cheng, C. C. Aggarwal, T. Derr, Fairness and diversity in recommender systems: A survey, ACM Trans. Intell. Syst. Technol. 16 (2025). URL: https://doi.org/10.1145/3664928. doi:10.1145/3664928.

[9] M. Kaminskas, D. Bridge, Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems, ACM Trans. Interact. Intell. Syst. 7 (2016). URL: https://doi.org/10.1145/2926720. doi:10.1145/2926720.

[10] M. Ge, C. Delgado-Battenfeld, D. Jannach, Beyond accuracy: evaluating recommender systems by coverage and serendipity, in: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, Association for Computing Machinery, New York, NY, USA, 2010, p. 257–260. URL: https://doi.org/10.1145/1864708.1864761. doi:10.1145/1864708.1864761.

[11] S. Wang, M. Gong, H. Li, J. Yang, Multi-objective optimization for long tail recommendation, Knowledge-Based Systems 104 (2016) 145–155. URL: https://www.sciencedirect.com/science/article/pii/S0950705116300600. doi:https://doi.org/10.1016/j.knosys.2016.04.018.

[12] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, M. S. Pera, All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness, in: S. A. Friedler, C. Wilson (Eds.), Proceedings of the 1st Conference on Fairness, Accountability and Transparency, volume 81 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 172–186. URL: https://proceedings.mlr.press/v81/ekstrand18b.html.

[13] Z. Ovaisi, S. Heinecke, J. Li, Y. Zhang, E. Zheleva, C. Xiong, Rgrecsys: A toolkit for robustness evaluation of recommender systems, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, WSDM '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 1597–1600. URL: https://doi.org/10.1145/3488560.3502192. doi:10.1145/3488560.3502192.

[14] G. Shani, A. Gunawardana, Evaluating Recommendation Systems, volume 12, 2011, pp. 257–297. doi:10.1007/978-0-387-85820-3_8.

[15] F. M. Harper, J. A. Konstan, The MovieLens datasets, ACM Transactions on Interactive Intelligent Systems 5 (2015) 1–19. URL: https://doi.org/10.1145/2827872. doi:10.1145/2827872.

[16] J. Ni, J. Li, J. McAuley, Justifying recommendations using distantly-labeled reviews and fine-grained aspects, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 188–197. URL: https://aclanthology.org/D19-1018/. doi:10.18653/v1/D19-1018.

[17] K. Emamy, R. Cameron, CiteUlike: a researcher's social bookmarking service, Ariadne 249 (2007) 5–110. URL: http://ariadnestaging.lboro.ac.uk/issue/51/emamy-cameron/. doi:10.1016/j.ejogrb.2020.04.008.

[18] J. Bennett, S. Lanning, The netflix prize (2007).

[19] P. Li, L. Yao, X. Li, Dual debiased recommendation model based on stacked autoencoder and reward mechanism for exposure bias and cold start, IEEE Access 14 (2026) 51170–51185. URL: https://doi.org/10.1109/access.2026.3672233. doi:10.1109/access.2026.3672233.

[20] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Knowledge-Based Systems 26 (2012) 225–238. URL: https://www.sciencedirect.com/science/article/pii/S0950705111001882. doi:https://doi.org/10.1016/j.knosys.2011.07.021.

[21] H. J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Sciences 178 (2008) 37–51. URL: https://www.sciencedirect.com/science/article/pii/S0020025507003751. doi:https://doi.org/10.1016/j.ins.2007.07.024.

[22] D. Lee, Personalizing information using users' online social networks: A case study of citeulike, Journal of Information Processing Systems 11 (2015) 1–21. doi:10.3745/JIPS.04.0014.

[23] D. H. Lee, P. Brusilovsky, Using self-defined group activities for improvingrecommendations in collaborative tagging systems, in: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, Association for Computing Machinery, New York, NY, USA, 2010, p. 221–224. URL: https://doi.org/10.1145/1864708.1864752. doi:10.1145/1864708.1864752.

[24] L. Michiels, R. Verachtert, B. Goethals, Recpack: An(other) experimentation toolkit for top-n recommendation using implicit feedback data, in: Proceedings of the 16th ACM Conference on Recommender Systems, RecSys '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 648–651. URL: https://doi.org/10.1145/3523227.3551472. doi:10.1145/3523227.3551472.

[25] O. Barkan, R. Hirsch, O. Katz, A. Caciularu, N. Koenigstein, Anchor-based collaborative filtering, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2877–2881. URL: https://doi.org/10.1145/3459637.3482056. doi:10.1145/3459637.3482056.

[26] D. Liu, P. Cheng, Z. Lin, X. Zhang, Z. Dong, R. Zhang, X. He, W. Pan, Z. Ming, Bounding system-induced biases in recommender systems with a randomized dataset, ACM Trans. Inf. Syst. 41 (2023). URL: https://doi.org/10.1145/3582002. doi:10.1145/3582002.

[27] S. Zhang, Y. Zhang, J. Chen, H. Sui, Addressing correlated latent exogenous variables in debiased recommender systems, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 3808–3818. URL: https://doi.org/10.1145/3711896.3736832. doi:10.1145/3711896.3736832.

[28] N. Marin, E. Makhneva, M. Lysyuk, V. Chernyy, I. Oseledets, E. Frolov, Tensor-based collaborative filtering with smooth ratings scale, ArXiv abs/2205.05070 (2022). URL: https://api.semanticscholar.org/CorpusID:248665600.

[29] Z. Yue, Z. He, H. Zeng, J. McAuley, Black-box attacks on sequential recommenders via data-free model extraction, in: Proceedings of the 15th ACM Conference on Recommender Systems, RecSys '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 44–54. URL: https://doi.org/10.1145/3460231.3474275. doi:10.1145/3460231.3474275.

[30] Z. Yue, H. Zeng, Z. Kou, L. Shang, D. Wang, Defending substitution-based profile pollution attacks on sequential recommenders, in: Proceedings of the 16th ACM Conference on Recommender Systems, RecSys '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 59–70. URL: https://doi.org/10.1145/3523227.3546770. doi:10.1145/3523227.3546770.

[31] L. McInnes, J. Healy, N. Saul, L. Großberger, UMAP: Uniform manifold Approximation and Projection, The Journal of Open Source Software 3 (2018) 861. URL: https://doi.org/10.21105/joss.00861. doi:10.21105/joss.00861.