A Replication & Reproduction of "Time-dependent Evaluation of Recommender Systems"

Nina Kühn, Matrikelnummer 1651199
nina.kuehn@student.uni-siegen.de
University of Siegen
Siegen, Germany

Hannes Wunderlich, Matrikelnummer 1714577
hannes.wunderlich@student.uni-siegen.de
University of Siegen
Siegen, Germany

ABSTRACT

Standard evaluation of recommender systems typically uses static, single-number metrics, which fail to capture how algorithm performance evolves over time. This paper addresses this research problem by replicating and extending the work of Scheidt and Beel, who proposed a time-dependent evaluation approach [6]. Following their methodology, we first replicated their experiments on three datasets and then applied the time-dependent analysis to five additional datasets and three new algorithms, focusing on the nDCG metric. Our results confirm the original findings: algorithm performance and rankings are highly unstable and change significantly over time, particularly in a dataset's early stages, with no single algorithm proving universally superior. Although the study was limited by computational costs, our findings strongly support the conclusion that a time-aware evaluation is crucial for a more realistic and exact assessment of recommender systems, moving beyond static benchmarks.

KEYWORDS

Recommender Systems, Evaluation, Time-dependent Evaluation

1 INTRODUCTION

1.1 Background

The evaluation of recommender systems is a crucial and actively discussed field within the research community. A standard and widespread practice is to assess the effectiveness of algorithms using single-number metrics, such as nDCG or RMSE. These metrics are typically calculated once over an entire dataset, providing a static, aggregated view of an algorithm's average performance across the whole data collection period [6].

1.2 Research Problem

A limitation of this evaluation approach is that a single, static number only shows the average performance over the entire data collection period. It does not show whether and how performance develops over time. Algorithm effectiveness is not necessarily constant; it can change as the dataset grows and evolves over time. This can result in performance developing in different ways over time, such as improving steadily, declining gradually or fluctuating.

These developments can, in turn, change the ranking of algorithms when they are compared. As Scheidt and Beel highlight, if two algorithms perform differently over time, their ranking against each other might also change [6]. This means that a conclusion about which algorithm is the best could depend on the specific point in time when the evaluation is done. The original paper states that using single-number metrics is a common practice. An analysis of publications from the ACM RecSys 2020 conference revealed that 89% of the examined papers that evaluated algorithms presented results using single-number metrics, while only 11% showed performance over time [6].

1.3 Original Work

In their paper "Time-dependent Evaluation of Recommender Systems," Scheidt and Beel address this very problem [6]. The authors suggest that the common practice of using single-number metrics is potentially misleading, as it fails to capture how algorithm performance evolves over time.

To investigate this, they conducted a systematic analysis of six common recommender algorithms (funkSVD, biasedMF, UserKNN, ItemKNN, Bias and Most Popular) across ten different datasets, including variations of MovieLens, Amazon, and Netflix. Their core methodological innovation was to split each dataset into multiple temporal subsets. Instead of creating distinct time intervals, each subsequent subset included all data from the previous ones, thereby simulating a realistically growing dataset as would be encountered in a real-world application. At each time step, they evaluated the algorithms using nDCG, recall and RMSE.

The study by Scheidt and Beel showed that for 90% of the datasets, the performance of the algorithms changed over time. More critically, in 60% of the cases, the relative ranking of the algorithms also changed, especially during the early phases of data collection when less data was available. They also observed that larger datasets like Netflix tended to show more stable rankings over time compared to smaller ones, and that performance trends could differ depending on the evaluation metric used (e.g., nDCG vs. RMSE).

Based on these findings, Scheidt and Beel concluded that a time-dependent evaluation provides crucial insights that are lost in a single, aggregated metric. They suggest presenting algorithm performance over time, as this reveals important trends and changes in ranking that single-number metrics overlook, enabling more nuanced and informed evaluations.

1.4 Research Goal

The primary goal of this work is to replicate the experiments and validate the findings presented by Scheidt and Beel [6]. Specifically, we investigate whether their observations regarding time-dependent performance hold in our experimental setting and whether their evaluation approach can be generalized to different datasets and recommendation algorithms.

2 METHODOLOGY

2.1 Overview

Our project consists of two parts: the replication of the original study by the authors, and the extension of their methodology to new datasets and algorithms (reproduction). The original source code was made publicly available by the authors, and we used it as the foundation for our experiments.[1] Our own modified and extended version of the code is available at: https://github.com/Hannesw0211/TM2_Team7.

2.2 Replication

We closely followed the experimental setup described by Scheidt and Beel in the original paper [6]. The authors provided code to run six algorithms from the LensKit library, evaluate them using multiple metrics, and compute performance over time by splitting datasets based on timestamps. We used the original code to replicate experiments on the following datasets: MovieLens 100k[2], MovieLens 1M[3] and Amazon Instant Video[4] [2] [4]. For the Amazon Instant Video dataset, we noticed a discrepancy between the number of ratings stated in the original paper and the actual dataset available at the referenced URL. While the paper reports 135,000 ratings, the dataset currently contains 583,933 ratings. It is unclear whether the original authors used a subset of the data or whether the dataset has changed since the time of their experiments.

Each dataset was split into subsets based on either monthly or yearly intervals, and for each time point t, all ratings up to t were included to simulate a growing dataset. For every subset, 20% of the most recent ratings from each user were set aside as a test set. Models were trained from scratch at each time step using grid search with five-fold cross-validation.

While the original paper reported results for both nDCG and RMSE, we limited our evaluation to nDCG due to runtime constraints. In the provided codebase, models are optimized twice - once with respect to nDCG and once with respect to RMSE - using separate parameter grids. However, in the version of the code uploaded by the authors to GitHub, the method responsible for evaluating nDCG was configured to use the RMSE grid file (‘Grids_rmse.xls‘) for hyperparameter tuning. It remained unclear which grid configuration had been used to generate the final nDCG results presented in the paper. To ensure consistency and better alignment with our metric of interest, we replaced the RMSE grid with the other grid file (‘Grids.xls‘) included in the original paper’s repository.

The exact software versions and environments used in the original experiments were not documented. The specifications of the system and software environment used in our experiments are provided in Appendix A.

[1] https://github.com/ISG-Siegen/recsys-time-evaluation
[2] https://grouplens.org/datasets/movielens/100k/
[3] https://grouplens.org/datasets/movielens/1m/
[4] http://jmcauley.ucsd.edu/data/amazon/

Table 1: Overview of datasets used in replication

Dataset	Ratings	Timespan	Split
MovieLens 100k	100,000	1995–1998	Monthly (8)
MovieLens 1M	1,000,000	2000–2003	Yearly (4)
Amazon Instant Video	584,000	2007-2014	Yearly (8)

2.3 Reproduction

To test the generalizability of the original findings, we extended the evaluation to five additional datasets: ModCloth[5] [7], Amazon Subscription Boxes[6] [3], Amazon Magazine Subscriptions[7] [5], Amazon Beauty[8] [5], and MovieTweetings[9]. All datasets contain timestamps, enabling us to apply the same time-based evaluation procedure.

For MovieTweetings, we downsampled the dataset to 20% due to its large size. For Amazon Subscription Boxes, we relaxed the minimum number of ratings per user from three to one, as the dataset was very sparse. In line with the original paper, we excluded all dataset subsets that had fewer than 500 ratings to ensure model stability and comparability and skipped early years for some datasets where the number of ratings was insufficient for meaningful evaluation.

We reused the six algorithms from the original study (FunkSVD, BiasedMF, Bias, UserKNN, ItemKNN, MostPopular) and added three new algorithms: Non-negative Matrix Factorization (NMF), Probabilistic Matrix Factorization (PMF), and a Random recommender. For NMF and PMF, we implemented a custom model class compatible with the existing pipeline.

We repeated each experiment three times using a fixed random seed (42) for reproducibility. The evaluation was again focused on nDCG (and recall), as RMSE was omitted due to time constraints. Although the original paper did not visualize recall results, it noted that recall trends were similar to those of nDCG.

Table 2: Overview of datasets used in reproduction

Dataset	Ratings	Timespan	Split
ModCloth	100,000	2012-2017	Yearly (6)
Amazon Subscription Boxes	16,200	2019-2023	Yearly (5)
Amazon Magazine Subscriptions	90,000	2002-2018	Yearly (17)
Amazon Beauty	370,000	2007-2018	Yearly (12)
MovieTweetings (20%)	920,000	2013-2021	Yearly (9)

This extended experimental setup formed the basis for analyzing whether the trends observed in the original study also hold in other domains and with alternative recommendation methods.

3 RESULTS & DISCUSSION

This section presents and discusses the results of our replication and reproduction study, focusing on the time-dependent evaluation of recommender system algorithms. The aim is to assess whether

[5] https://cseweb.ucsd.edu/~jmcauley/datasets.html#market_bias
[6] https://amazon-reviews-2023.github.io/
[7] https://nijianmo.github.io/amazon/index.html
[8] https://nijianmo.github.io/amazon/index.html
[9] https://github.com/sidooms/MovieTweetings

the key findings reported by Scheidt and Beel [6] also hold across different datasets and extended algorithm configurations.

Our results indicate that algorithm performance is not static: both nDCG and recall values fluctuate significantly across years, reflecting changes in user behavior, item popularity, and dataset sparsity.

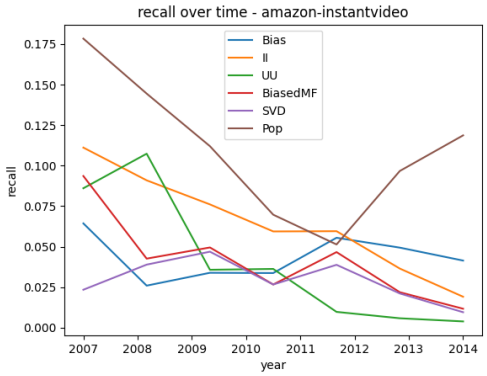


Figure 1: Fluctuation of nDCG and recall values across years

In several years, the ranking of the best-performing algorithm changed, underlining the importance of temporal dynamics in algorithm evaluation.

Furthermore, as shown in Figure 1 we observed that the gap between algorithms varied over time, with performance differences being most pronounced in earlier years when the dataset was smaller. As more data became available, most algorithms showed improved and more stable performance, though the relative order was not always preserved.

These findings support the claim that evaluating recommender systems over time yields richer insights than traditional single-number summaries. They emphasize the necessity of time-aware evaluation methods, especially when deploying algorithms in real-world dynamic environments.

The following subsections present the detailed results of our replication and reproduction, followed by a reflection on general patterns and implications.

3.1 Replication Results

In the process of our replication of the original paper, we limited our evaluation to nDCG and recall due to runtime constraints and there being minimal deviation.

Table 3: Overview of Algorithm Performance Changes

Dataset	Ranking	Trend	Range (%)
ML 100K	1(0)	Increasing	0.17-0.23(26,09%)
ML 1M	0(0)	Stable	0.18-0.19(5,26%)
A Instant	18(2)	Decreasing	0,051-0,178(71,35%)

Table 3 presents the performance variations of different algorithms. The first columns indicate the dataset used in each case.

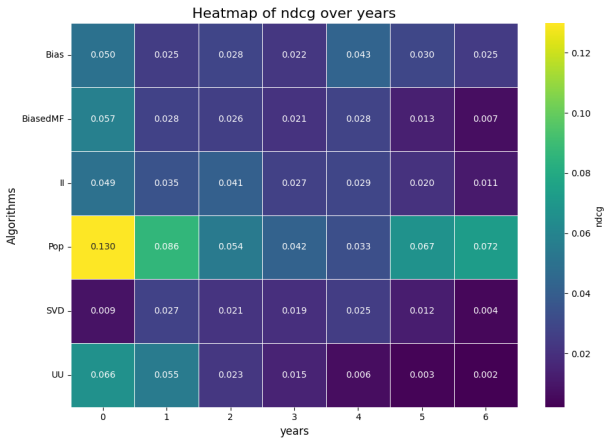


Figure 2: Heatmap of ndcg over the years (Instant Video)

The Ranking column reports the number of instances in which a reordering of algorithm performance occurred and in brackets counts the number of reordering of the best algorithm. The Trend column examines the temporal development of the best-performing algorithm. The Range column reflects the performance span of the top algorithm, representing the difference between its maximum and minimum performance values.

When analyzing the results of the Amazon Instant Video Dataset, we found substantial variation in both nDCG, Recall and algorithm rankings over time.

Performance Over Time. The substantial variation in both nDCG and Recall over time, indicating that the performance of recommendation algorithms is highly time-dependent: The nDCG values of individual algorithms fluctuated between 55% and 97% over time. The Recall values showed similarly strong dynamics, ranging between 60% and 96%. UserKNN (UU) experienced the highest variability with a nDCG range of 96.7% and Recall range of 96.4%, dropping from 0.066 to 0.002 in nDCG over the years. BiasedMF showed an nDCG drop of 88%, and Recall fell by 87.5%. Even the seemingly stable Most Popular algorithm (Pop) varied significantly, with nDCG decreasing by 75% and Recall by over 70%. These variations are not uniform across algorithms: some algorithms like Bias had relatively smaller fluctuations (nDCG range 55%), while others like SVD or ItemKNN were more volatile. Such volatile values also emerge from the original paper.

Algorithm Rankings. In addition to absolute performance changes, we observed frequent ranking shifts among the algorithms. In 2 out of 8 years (25%), the top-performing algorithm (in terms of nDCG) changed. Toward the later years (2012–2014), the performance curves began to stabilize for most algorithms, with Pop and BiasedMF often leading in recall and nDCG respectively.

Comparison to Original Results. Although the results of our reproduction exhibit similar patterns to those reported in the original paper, they are not identical. This discrepancy can be attributed to several factors. First, the datasets have evolved and expanded over

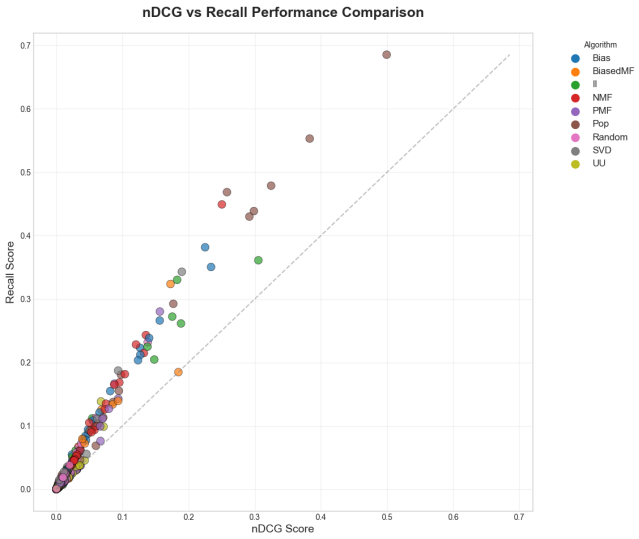


Figure 3: nDCG vs Recall Performance Comparison

time, leading to differences in their composition. Second, the original paper did not specify dataset or library versions, which may have resulted in version conflicts. Third, the hyperparameters for the algorithms appear to have been randomly selected from tables, potentially leading to variations in performance. Lastly, no fixed random seed was provided, further complicating the reproducibility of the experiments.

3.2 Reproduction Results

To ensure robustness in our experiments, each evaluation was repeated three times using a fixed random seed of 42. The reported values represent the mean across all runs, as noticeable variability was observed between individual executions.

Metric Selection and Correlation Analysis. Although both nDCG and Recall were computed, only nDCG is reported in detail. This decision is consistent with the original study, where Recall was omitted due to space constraints and due to its strong similarity in trend to nDCG. We observed a clear positive correlation between the two metrics. Algorithms that achieved high nDCG scores generally also scored highly in Recall. In our scatter plot comparing both metrics (see Figure 3), most points were positioned close to the $y = x$ diagonal, suggesting that the overall evaluation conclusions would not significantly differ between the two metrics. While Recall scores tended to be higher in absolute terms, the development over time mirrored that of nDCG, further justifying our decision to focus exclusively on nDCG.

Temporal Performance Trends. The original study found that performance changed over time in 90% of the evaluated datasets. In our reproduction, this was the case for all five datasets.

Appendix D visualizes the absolute nDCG scores of each algorithm over time for all datasets. Each line represents the actual development trajectory of one algorithm, making visible whether performance improved, declined, or fluctuated.

Overall, the temporal development was characterized by a general decline in nDCG values from early to late stages. This can be seen clearly in the aggregated heatmap visualization (Appendix B), where brighter colors represent higher scores and are concentrated in the early phases of each dataset. Among the best-performing algorithms in the early phase (0–20%) were *Most Popular* with a peak nDCG of 0.605 and *NMF* with 0.527. While *Most Popular* declined considerably over time, reaching a final-stage value of 0.166, *NMF* showed a slightly more stable trajectory through the middle phases (e.g., nDCG of 0.183 in the 40–60% range), although it too decreased toward the end. Other algorithms such as *Random* and *SVD* demonstrated consistently poor performance across all phases, suggesting their unsuitability for the datasets in question. Algorithms like *Bias*, *BiasedMF*, and *PMF* began with moderate nDCG values but showed noticeable degradation in later stages.

Algorithm Ranking Stability. In the original work, the authors observed that algorithm rankings changed in 60% of datasets and were especially unstable during the early phases. Our reproduction confirmed this pattern but extended it further: all datasets in our study experienced at least some changes in ranking over time. The level of volatility, however, varied significantly by dataset.

Among the evaluated datasets, Amazon Magazine Subscriptions exhibited the highest degree of ranking instability, with frequent and irregular changes in algorithm positions throughout the timeline. The Beauty dataset also showed considerable fluctuations, particularly in the middle phase of its lifecycle. In contrast, the ModCloth and Subscription Boxes datasets displayed relatively stable rankings over time, with only occasional shifts. MovieTweetings emerged as the most consistent dataset in terms of algorithm ranking, maintaining a largely unchanged order across the entire evaluation period. These dynamics are illustrated in Appendix C, where horizontal lines reflect stable rankings and frequent crossings indicate volatility.

Interestingly, some datasets transitioned from unstable to more stable phases. In the Beauty dataset, for example, rankings fluctuated considerably in the middle period but stabilized toward the end. In contrast, Magazine remained extremely unstable throughout.

Best-Performing Algorithms and Dataset Variability. No single algorithm was universally dominant across all datasets. *Most Popular* performed best on ModCloth and Subscription Boxes. *UserKNN* (UU), *PMF*, and *NMF* led on Beauty, Magazine, and MovieTweetings, respectively. This variation supports the observation from the original study that algorithm effectiveness is dataset-dependent and subject to change over time.

Table 4 summarizes the ranking dynamics, performance trends, and nDCG ranges for each dataset.

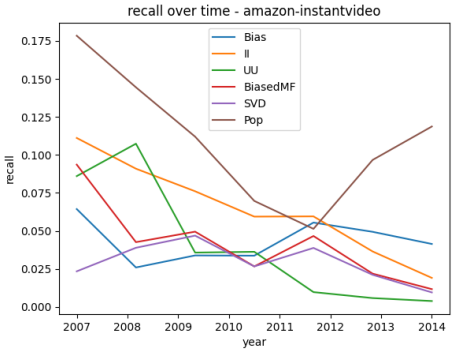
The original paper also hypothesized that larger datasets tend to produce more stable performance and ranking trajectories. While this was supported in their results, our reproduction suggests a more nuanced picture. The largest dataset in our evaluation, Amazon Magazine Subscriptions, covering 17 years, exhibited the most unstable behavior. In contrast, smaller datasets like Subscription Boxes, ModCloth, and MovieTweetings showed more stable development. This may be attributed to the overall smaller scale of our datasets compared to the original study, which used data sets

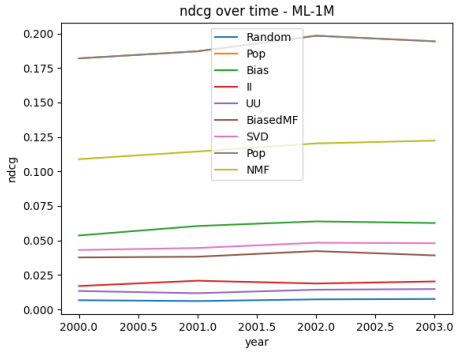
with up to 51 million ratings. Our largest dataset contained around 370,000 ratings.

3.3 General Observations and Insights

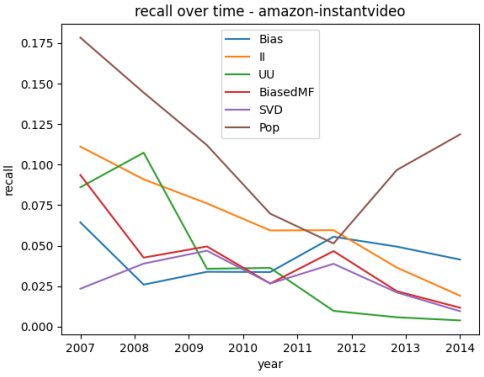
Our results strongly support the core argument presented by Scheidt and Beel [6]: evaluating recommender systems over time yields richer and more reliable insights than single-number performance summaries. Both our replication and reproduction revealed that algorithm performance is often unstable, particularly in the early stages of a dataset's lifecycle. This was most clearly observed in the Amazon Instant Video during the replication, where performance rankings changed frequently. Such volatility highlights that any claim about the "best" algorithm is highly dependent on the time of evaluation.

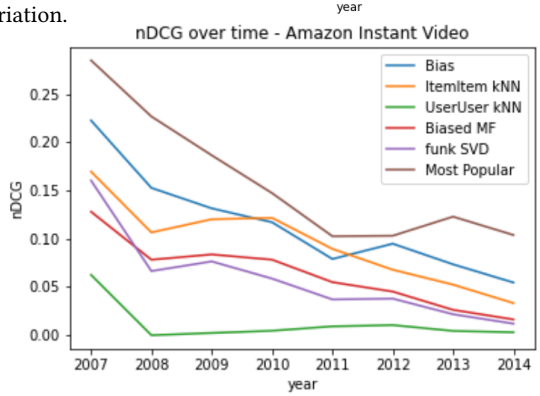
The same conclusion emerged from our extended reproduction study across five additional datasets. Here too, we observed that performance trends varied significantly between datasets. While the Amazon Magazine Subscriptions dataset showed highly unstable algorithm rankings throughout, the MovieTweetings and ModCloth datasets demonstrated much greater consistency. These observations reinforce the original claim that dataset characteristics — such as size, sparsity, and growth over time — play a central role in shaping algorithm behavior. While our replication results align with the original paper's claim that larger and denser datasets tend to exhibit more stable performance and rankings over time—as seen, for example, in the MovieLens 1M dataset — our reproduction results suggest that this pattern does not always hold. In fact, some of the larger datasets in our reproduction, such as Amazon Magazine Subscriptions, showed considerable instability, whereas smaller datasets like ModCloth or MovieTweetings demonstrated stable performance and rankings. This discrepancy indicates that additional factors, such as sparsity or dataset quality, may play a more substantial role than dataset size alone.





Another important observation concerns the evaluation metrics themselves. Our experiments, which focused on nDCG and Recall, showed that these two ranking-based metrics tend to evolve similarly over time. This confirms the original authors' claim that both metrics capture comparable dynamics. However, unlike the original study, we did not include RMSE in our evaluation due to computational limitations. As such, we cannot comment on whether error-based metrics would have revealed different temporal trends, though the original work noted metric-dependent variation.





Taken together, these findings from both our replication and reproduction underline the importance of time-aware evaluation methods. They cast doubt on the validity of static performance comparisons and call for a shift toward temporal benchmarking. While our reproduction showed that no algorithm consistently outperformed others over time, the replication results—such as those on MovieLens 1M—demonstrated stable rankings.

Ultimately, our results highlight the necessity of evaluating recommender systems as temporally evolving systems rather than static entities.

4 LIMITATIONS

We acknowledge several limitations in our work. A primary challenge was the selection of suitable datasets. Our goal was to replicate and extend the original study, which already incorporated several standard benchmark datasets commonly used in recommender systems research. This limited the availability of comparable datasets that met our criteria, such as having sufficient data variance and reliable timestamps to generate meaningful recommendations.

The high computational cost and long runtimes of the experiments were significant constraints throughout this study. Consequently, we were unable to include all datasets used by the original authors in our replication. For the same reason, we did not extend our analysis to larger-scale datasets, such as those with millions of ratings, which would have provided further insights into the scalability of the methods. Furthermore, to manage computational demands for the MovieTweetings dataset, we applied a downsampling strategy. We must note that a direct consequence of this preprocessing step is an increase in data sparsity, which can make the recommendation task inherently more challenging.

Due to these computational and time constraints, we did not perform an exhaustive hyperparameter optimization for the replication. Instead, we adopted the parameter settings reported in the original work. Similarly, our experiments were conducted without repetitions using multiple random seeds. Therefore, the reported results may be subject to statistical variance, and a more robust evaluation would involve averaging performance over several runs. Finally, in our evaluation, we focused on ranking-based metrics. We did not include the Root Mean Squared Error (RMSE) as an evaluation metric.

5 ACKNOWLEDGMENTS

This work was conducted as part of the Machine Learning Praktikum at the University of Siegen [1].

REFERENCES

[1] Joeran Beel and Lukas Wegmeth. 2025. Machine Learning Praktikum. *Universisät Siegen* (2025).
[2] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *WWW* (2016). Available: https://cseweb.ucsd.edu/~jmcauley/pdfs/www16a.pdf.
[3] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).
[4] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. *SIGIR* (2015). Available: https://cseweb.ucsd.edu/~jmcauley/pdfs/sigir15.pdf.
[5] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP).*
[6] Teresa Scheidt and Joeran Beel. 2021. Time-dependent Evaluation of Recommender Systems. (2021).
[7] Mengting Wan, Jianmo Ni, Rishabh Misra, and Julian McAuley. 2020. Addressing Marketing Bias in Product Recommendations. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM).*

A SYSTEM SPECIFICATIONS

All experiments were conducted on the following hardware and software setup:

Device 1.

- Operating System: Windows 10
- CPU: AMD64 Family 25 Model 33
- RAM: 16GB
- Python Version: 3.9.8
- LensKit Version: 0.14.1
- NumPy Version: 1.21.5
- Pandas Version: 1.3.5
- scikit-learn Version: 1.0.2

Device 2.

- Operating System: Windows 10
- CPU: Intel64 Family 6 Model 158
- RAM: 16GB
- Python Version: 3.9.8
- LensKit Version: 0.14.4
- NumPy Version: 1.26.4
- Pandas Version: 2.2.3
- scikit-learn Version: 1.6.1

Device 3.

- Operating System: Windows 11
- CPU: Ryzen 7 5800X
- RAM: 16GB DDR4
- Python Version: 3.9.8
- LensKit Version: 0.14.4
- NumPy Version: 1.26.4
- Pandas Version: 2.2.3
- scikit-learn Version: 1.6.1

B REPRODUCTION RESULTS: HEATMAP

C REPRODUCTION RESULTS: RANKING OF ALGORITHMS

D REPRODUCTION RESULTS: RANKING OF ALGORITHMS

E REPRODUCTION RESULTS: TABLE

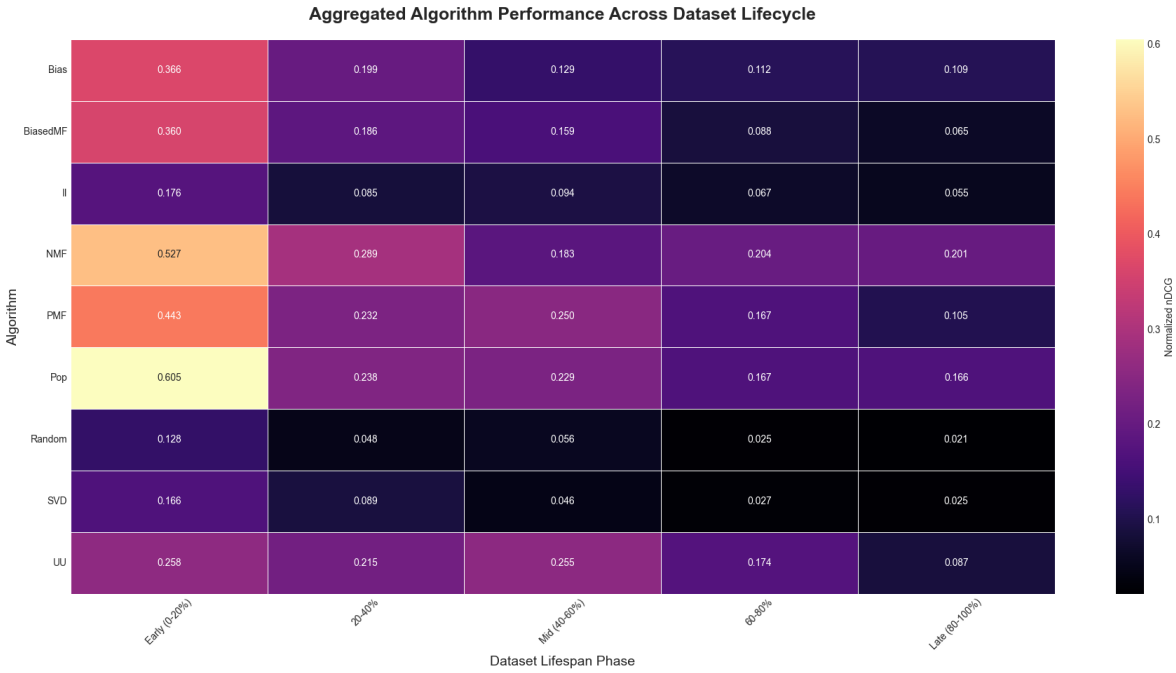


Figure 4: Heatmap of aggregated nDCG performance for all algorithms (y-axis) across five dataset lifecycle phases (x-axis). Brighter colors indicate higher performance. The figure reveals a general decline in performance from early to late dataset phases across all algorithm–dataset combinations.

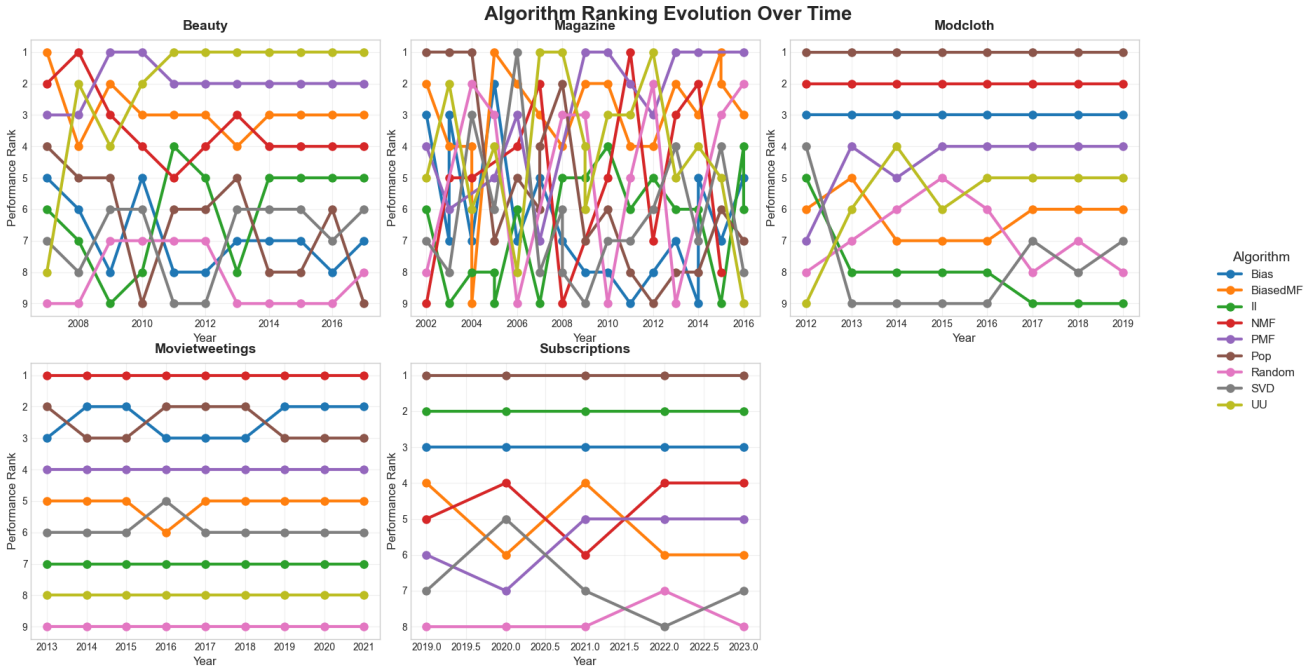


Figure 5: Ranking evolution plots for each dataset. Each line represents the position of an algorithm over time (y-axis: 1 = best). Horizontal lines indicate stable rankings, while frequent changes denote instability.

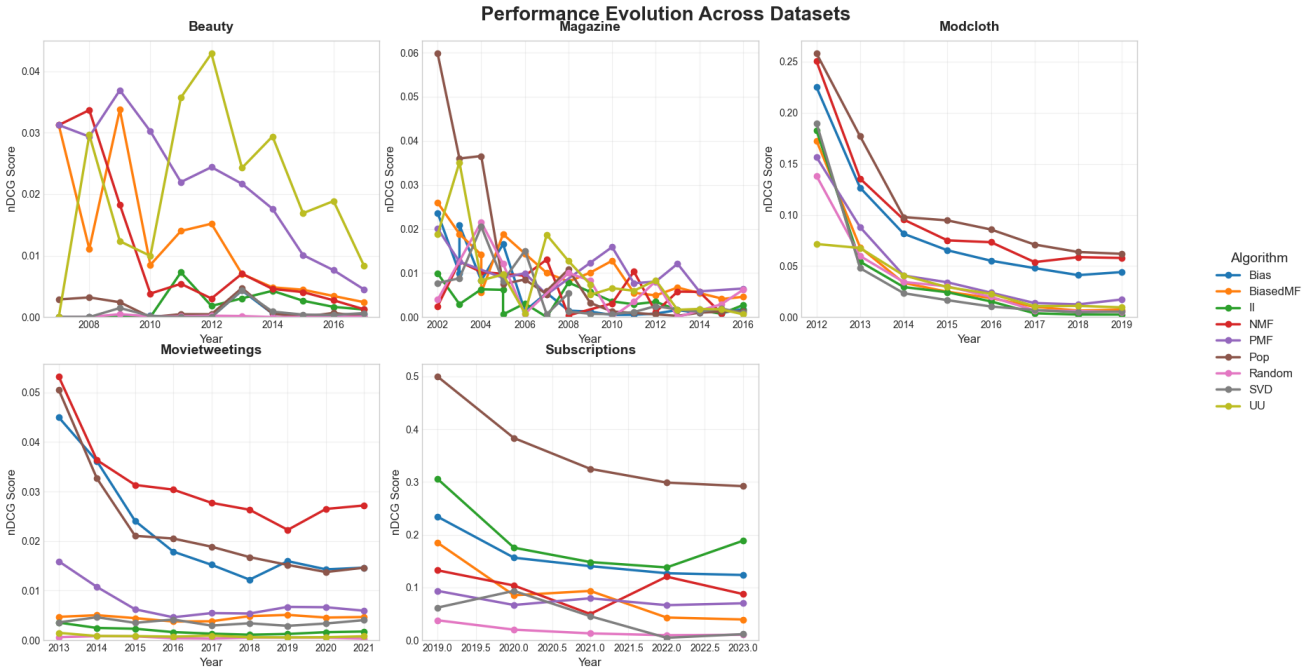


Figure 6: Absolute nDCG values of all algorithms over time for each dataset. Each colored line traces the time-dependent performance of a specific algorithm. Falling lines indicate performance degradation, while rising lines denote improvement.

Table 4: Summary of nDCG Results in the Reproduction Study

Dataset	changed best Algo	Trend of Best Algo	nDCG Range of best Algo	Best Algorithm
Beauty	Yes	Mixed	0.000–0.043 (0%)	UserKNN
Magazine Subscriptions	Yes	Mixed	0.005–0.020 (292%)	PMF
ModCloth	No	Decreasing	0.062–0.258 (317%)	Most Popular
MovieTweetings (20%)	No	Decreasing	0.022–0.053 (139%)	NMF
Subscription Boxes	No	Decreasing	0.292–0.499 (71%)	Most Popular