Submission date: 04/11/2025
Resubmission date: 15/04/2026
Camera-ready submission date: 15/06/2026

1st round notification: 09/03/2026
2nd round notification: 08/06/2026
Available online: 30/07/2026
Publication date: 30/07/2026

Section: [regular article]

Reminders in Session-Based Recommender Systems: A Multi-Domain Evaluation

Douglas Rorie Tanno[1], Gabriel Libardi Lulu[1], Vinicius Sylvestre Simm[1], Matheus Henrique Cecilio Leme[1], Marcos Aurelio Domingues[1]

[1]Department of Informatics – State University of Maringá (UEM)
Maringá, Paraná – Brazil

{pg404806, ra134728, pg404811, pg55308, madomingues}@uem.br

Abstract. *Online services expose users to vast amounts of content, leading to information overload and making it difficult to find relevant items. Recommender systems address this by personalizing user experiences based on preferences and interaction history. This work investigates Session-Based Recommender Systems (SBRS) with reminders, which reuse previously viewed items to enhance recommendation relevance. Experiments across six domains show that reminders improve performance in legal, employment, and e-commerce contexts, while having a smaller impact in music, tourism, and news, highlighting opportunities for further exploration.*

Keywords. *Recommender Systems; Reminders; Session-Based Recommender Systems*

1. Introduction

E-commerce platforms, streaming services, and social networks generate large volumes of information, which can easily overwhelm users. Recommender systems alleviate this overload by predicting items likely to be relevant, supporting decision-making, and facilitating content discovery. These systems leverage explicit preferences, historical interactions, similarities with other users, browsing behavior, contextual signals (e.g., time, location, device), and diverse data modalities such as text, images, audio, and multimedia [Ibrahim et al. 2025, Xu et al. 2026].

The increasing availability of heterogeneous data and the growth of online platforms have led to the development of sophisticated recommendation algorithms capable of capturing both short-term interests and long-term preferences. Modern approaches not only aim to improve accuracy but also address challenges such as diversity, novelty, fairness, and transparency, which are crucial for maintaining user trust and engagement. Additionally, the rapid pace of user behavior evolution in dynamic domains, such as news consumption, music streaming, and e-commerce, necessitates adaptive recommendation strategies that can react to shifting interests in near real-time [Saifudin and Widiyaningtyas 2024, Masciari et al. 2024].

A common strategy is to suggest items related to those previously explored, aligning recommendations with recent user activity to increase engagement and relevance.

Cite as: Tanno, D. R., Lulu, G. L., Simm, V. S., Leme, M. H. C., Domingues, M. A. (2026). Reminders in Session-Based Recommender Systems: A Multi-Domain Evaluation. iSys: Revista Brasileira de Sistemas de Informação (iSys - Journal of Information Systems), 19(1), 9:1-9:30. DOI: 10.5753/isys.2026.6809

Session-Based Recommender Systems (SBRS) focus on short-term preferences by analyzing sequences of interactions within a session, which is a coherent set of actions occurring within a limited time frame. Unlike traditional systems that use the full user history, SBRS rely solely on current-session interactions, making them particularly effective when preferences change rapidly, such as in last-minute purchases or dynamic media consumption [Zhang et al. 2025, Zhang et al. 2026].

In addition to traditional recommendation strategies, the incorporation of *reminders* has demonstrated notable effectiveness in recommender systems. This strategy reintroduces items previously viewed by the user within the current session, highlighting products or content of recent interest. Many online platforms integrate *reminders* into their recommendation lists to support decision-making, reduce cognitive load, and enhance user trust. Moreover, *reminders* can aid in the creation of future shopping lists and serve as a temporary solution in cold-start situations, where limited user information is available. This mechanism is particularly valuable in domains such as music and video, encouraging users to revisit previously consumed items that might otherwise be forgotten [Jannach et al. 2017, Domingues et al. 2022, Tanno et al. 2025].

Beyond improving immediate recommendations, *reminders* may also foster long-term engagement and retention by facilitating the continuation of purchase decisions or content consumption, thereby promoting sustained interaction with the platform [Jannach et al. 2017].

In this work, we investigate the use of *reminders* in recommender systems across multiple application domains. We hypothesize that incorporating *reminders* can enhance recommendation relevance while improving user satisfaction. To evaluate this, we conduct experiments with SBRS utilizing *reminders* in the legal, music, employment, tourism, and news domains, as well as in e-commerce platforms, where this strategy has been most applied. By examining these diverse contexts, we aim to determine whether the advantages of *reminders* extend beyond e-commerce and can improve recommendation performance in other domains. The effectiveness of SBRS with *reminders* was assessed by comparison with baseline models that do not employ this strategy.

The remaining of this paper is organized as follows: The related work is described in Section 2. Section 3 details the methodology, covering datasets, data preprocessing, training and test set splitting, SBRS models, *reminder* strategies, and evaluation setup. Section 4 presents and discusses the experimental results. Finally, conclusions and future directions are presented in Section 5.

2. Related Work

Research on the use of *reminders* in Session-Based Recommender Systems (SBRS) remains limited, with most studies concentrating on the e-commerce domain.

Lerche et al. (2016) conducted a comparison of several recommendation strategies using e-commerce datasets, including Zalando, China-Gadgets, and TMall. Their *reminder* strategy, *IRec*, was employed as a baseline and evaluated against approaches including *IInt*, *ISim*, and *SSim*. *IRec* recommends items based on the timestamp of the user's most recent interaction, whereas *IInt* leverages the frequency of item views in the

user's history. *ISim* identifies items aligned with the current purchase goal by computing cosine similarity between item descriptions, and *SSim* searches for past sessions with the same purchase goal, also using cosine similarity.

The evaluation considered, for each target purchase, the p sessions preceding a time interval of d days, with p set to six and d ranging from 0 to 3 days. To improve recommendation quality, a *Feature Filter* (FF) was introduced to dynamically exclude *reminders* from categories that have already been purchased within a session. For $d = 0$, *IRec* achieved high Hit Rate (HR), reflecting users' tendency to view items shortly before purchase. *IInt* and *ISim* exhibited lower performance, while *SSim* performed best for Zalando's highly engaged users, suggesting that users often abandon sessions and return later. For $d = 3$, overall HR and MRR declined, but the combination of *ISim* with FF provided the best results by filtering previously purchased categories. In China-Gadgets, *IRec* performed well due to the temporal ordering of interactions.

Beyond e-commerce, Domingues et al. (2022) investigated *reminders* in the legal domain using the JusBrasilRec dataset. They applied a hybrid *reminder* strategy combining *IRec* and *SSim* with SBRS models. Results indicated that the hybrid approach did not consistently improve performance and occasionally led to degradation, highlighting the importance of parameter tuning when applying *reminders*.

Overall, the limited body of research on *reminders* in SBRS underscores that this is an emerging area. Existing studies predominantly focus on single-domain e-commerce scenarios, with minimal exploration of other contexts. Our work seeks to address this gap by systematically evaluating *reminder* strategies across multiple application domains.

3. Methodology

This section describes the methodology employed to conduct the experiments in this work. The main steps are summarized in Figure 1 and are further detailed throughout the following subsections.

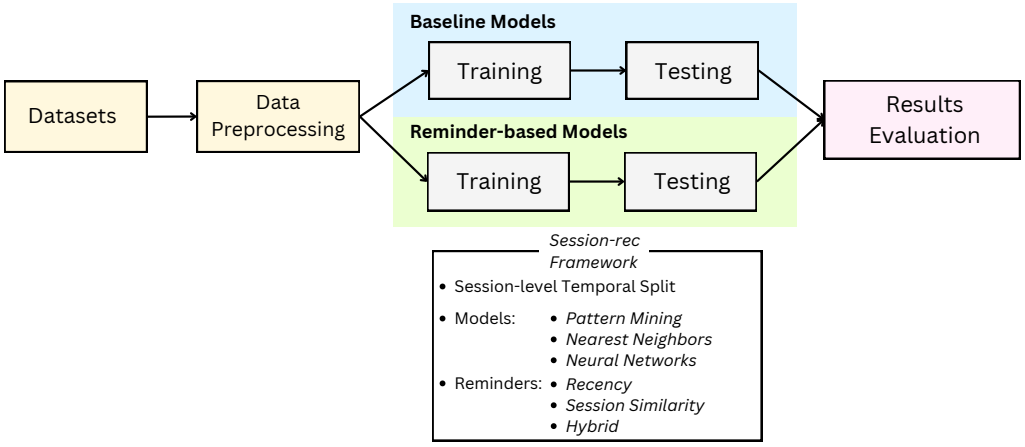

Figure 1. Overview of the main steps involved in the development of this work.

3.1. Datasets

For the experiments, we used six public datasets frequently adopted in SBRS research. The datasets are related to legal, music, employment, tourism, news, and e-commerce domains. Each dataset was divided into consecutive time-based slices, with the number and duration of slices adjusted according to the dataset's overall time span and interaction volume. More details about the slice process adopted in our work can be found in Subsection 3.3. The datasets used are:

- **JusBrasilRec:** Contains interaction data from 2,310,247 users across 4,225,874 items, grouped into 5,415,623 sessions over a 30-day period on the JusBrasil legal platform. Each record includes session and user IDs, user type (logged-in or anonymous), document ID and type, and a timestamp [Domingues et al. 2023]. For evaluation purposes, we divided the dataset into five chronological slices, each covering a 5-day span.
- **Xing:** From ACM RecSys Challenge 2016, with about 1.5 million users, 12 million interactions (clicks, bookmarks, replies), and 327,000 job postings. Job metadata includes title, description, career level, industry, and location; user profiles have anonymized experience and region info [Abel et al. 2016]. Split into five slices of 16 days.
- **Music4All:** Includes 15,602 users and 109,269 songs, with listening histories, metadata (artist, album, release year), lyrics, 30-second audio clips, user and genre tags, and Spotify audio features such as danceability, energy, and popularity [Pegoraro Santana et al. 2020]. Divided into five slices of 21 days.
- **Trivago:** Used in the ACM RecSys Challenge 2019, it contains user interactions in sessions on the Trivago hotel search platform. The dataset includes anonymized information from approximately 700,000 users, totaling nearly 16 million interactions. Each session contains the sequence of accommodations viewed and the indication of the item clicked at the end of the interaction [Knees et al. 2019]. The data were partitioned into four slices of 7 days.
- **MIND:** Released for the Microsoft News Personalization Competition, it is a large-scale news recommendation dataset collected from anonymized behavior logs of the Microsoft News website. It contains impression logs of user-news interactions, including clicks and non-clicks, along with rich metadata such as news titles, abstracts, categories, and entities. The dataset covers interactions from November 2019 and involves 1 million users and 161,013 English news articles [Wu et al. 2020]. We divided it into three slices of 2 days.
- **RetailRocket:** Collected from a real-world e-commerce platform, this dataset includes user behavior data and item attributes such as price, category, vendor, and product type. It contains 212,182 actions in 59,962 sessions across 31,968 items [Zykov 2022]. We divided it into five slices of 27 days.

3.2. Data Preprocessing

To ensure the quality of the data, all datasets underwent preprocessing prior to the experiments. Table 1 presents the statistics of the original datasets before any preprocessing steps, highlighting the scale and heterogeneity of the raw data.

Sessions containing only a single interaction were removed, as they provide limited information for training recommendation models. Likewise, sessions with more than 50 interactions were excluded, since they may represent atypical user behavior that does not reflect standard platform usage. Additionally, duplicate sessions were eliminated to preserve the uniqueness of user interaction sequences and avoid redundancy. Items with fewer than five occurrences in the dataset were also removed to reduce sparsity and ensure that only sufficiently represented items were considered during training. Furthermore, users with fewer than three sessions were filtered out, ensuring a minimum of three sessions per user. This preprocessing strategy follows a protocol similar to that adopted in prior work [Domingues et al. 2022].

Table 2 presents a summary of the datasets after preprocessing, including the total number of interactions, distinct items, sessions, session length statistics, and the data collection timespan[1], illustrating the impact of the filtering steps on data sparsity and session structure.

Table 1. Dataset statistics before preprocessing.

Statistics	JusBrasilRec	Xing	Music4All	Trivago	MIND	RetailRocket
Number of Interactions	22,442,232	8,826,678	5,109,444	5,934,159	16,568,353	1,314,043
Number of Items	4,225,874	1,029,480	99,596	394,238	94,443	132,923
Number of Sessions	5,415,623	784,699	617,021	1,080,988	1,493,003	382,084
Items per Session (min.)	2	1	1	1	1	1
Items per Session (mean)	4.14	11.25	8.28	5.49	11.10	3.44
Items per Session (max.)	50	4148	544	660	813	50
Timespan in Days	29	79	109	7	6	137

Table 2. Dataset statistics after preprocessing.

Statistics	JusBrasilRec	Xing	Music4All	Trivago	MIND	RetailRocket
Number of Interactions	3,637,949	255,291	3,998,499	689,501	1,024,266	187,341
Number of Items	100,052	19,898	48,957	39,740	5,894	11,980
Number of Sessions	905,824	52,850	522,387	91,516	337,579	42,732
Items per Session (min.)	2	2	2	2	2	2
Items per Session (mean)	4.02	4.83	7.65	7.53	3.03	4.38
Items per Session (max.)	50	50	50	50	43	47
Timespan in Days	29	79	109	7	6	134

The comparison between Tables 1 and 2 highlights the substantial impact of the preprocessing steps across all datasets. Overall, there is a significant reduction in the number of interactions, items, and sessions, as the filtering of infrequent items and extreme session lengths leads to a more compact and dense representation of user behavior.

The removal of rare items drastically reduces the item space (e.g., from millions to hundreds of thousands in datasets such as JusBrasilRec), effectively mitigating sparsity and improving the reliability of learned patterns. Additionally, session filtering results in

[1]The preprocessed datasets are available at `https://github.com/DouglasTanno/reminders-sbrs/`

more homogeneous interaction sequences, as reflected by the consistent minimum session length of two and the bounded maximum length across datasets. However, the impact of these preprocessing steps varies across domains. For instance, Xing exhibits a strong reduction in average session length (from 11.25 to 4.83), suggesting that longer sessions in the original data may contain noisy or less informative interactions. In contrast, datasets such as Music4All and RetailRocket retain relatively higher average session lengths, indicating more stable user interaction patterns. These differences highlight the importance of domain characteristics when applying preprocessing strategies in SBRS.

Such variations also suggest that each dataset may introduce specific biases that influence the recommendation task. Datasets with shorter sessions and limited temporal coverage, such as MIND and Trivago, tend to reflect more dynamic and short-lived user behavior. In contrast, datasets with longer sessions, such as Music4All and RetailRocket, provide richer interaction sequences, which may better capture user intent but can also introduce additional noise.

Moreover, while reducing the item space mitigates sparsity and stabilizes the learning process, it may also bias the data toward more frequent items and highly active sessions. As a result, the processed datasets provide a more stable and less sparse representation of user behavior, while potentially underrepresenting less frequent interactions and irregular session patterns.

Finally, domain-specific characteristics further influence user behavior. For instance, interactions in legal and e-commerce platforms tend to be more goal-oriented, whereas domains such as music and news often involve exploratory or rapidly changing preferences. These factors should be considered when interpreting the results, as they may affect how well different approaches generalize across domains.

3.3. Training and Test Set Splitting

Each dataset was split into temporal slices, with the window size d (in days) chosen to fit the overall timespan of the dataset. For each slice, a user-level split was applied, where the most recent session of each user was assigned to the test set and the earlier sessions were used for training, as shown in Figure 2. This approach maintains the chronological order of interactions, reflecting realistic settings for recommender systems.

3.4. SBRS Models

The experiments were conducted using the *session-rec* recommendation framework[2], which provides a variety of models employed in SBRS with and without *reminders*—the latter referred to as baseline models.

The recommender models used in our work can be grouped into three main categories based on their underlying principles: *Pattern Mining*, *Nearest Neighbors*, and *Neural Networks* [Jannach et al. 2017, Domingues et al. 2023].

Pattern Mining-Based Models explore item co-occurrence patterns to identify frequent associations in user interaction data. These models recommend items that often

[2]https://github.com/rn5l/session-rec

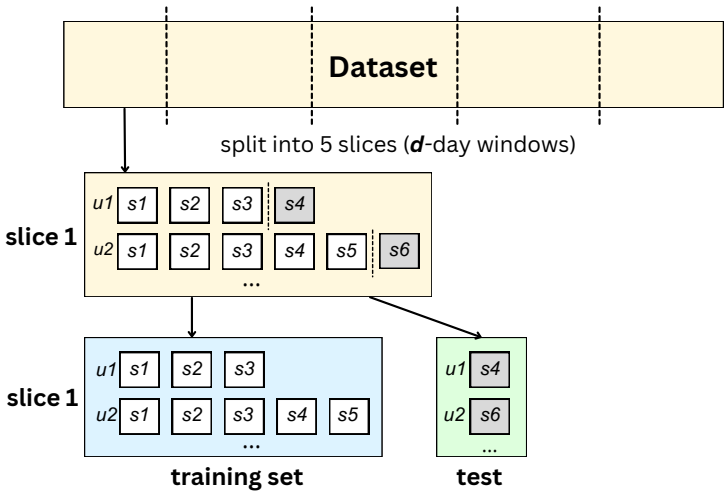

Figure 2. Illustration of the data splitting setup into 5 slices (d-day windows). For each user u_n within a slice, the last session s_n^{last} is allocated to the test set, while the previous sessions $s_n^1, \ldots, s_n^{last-1}$ are used for training.

appear together, leveraging recurrent behaviors to enhance the relevance of recommendations [Jannach et al. 2017, Kamehkhosh et al. 2017]. The following model was included in this category:

- `sr`: Based on the *Sequential Rules*, this model builds upon the classical *Association Rules* (`ar`) approach, which identifies frequent item co-occurrences in the form of rules $i \rightarrow j$, indicating that item j is likely to follow item i. The `sr` model extends `ar` by incorporating the sequential order of items within sessions. It applies a decay function that penalizes larger distances between items, thus giving more weight to short-term sequential dependencies observed in training sessions [Kamehkhosh et al. 2017].

Nearest Neighbors-Based Models, despite their simplicity, have demonstrated competitive performance in various session-based recommendation tasks [Kamehkhosh et al. 2017, Verstrepen and Goethals 2014, Jannach and Ludewig 2017]. Three variants were selected:

- `vsknn`: The *Vector Multiplication Session-Based KNN* is a variant of `knn` that emphasizes recent items when calculating session similarity. The current session is encoded as a real-valued vector, where only the last item receives weight 1, and the others are weighted by a linear decay function based on their position. Similarity is computed using the dot product, which gives more influence to recent interactions [Ludewig and Jannach 2018].
- `stan`: The *Sequence and Time-aware Neighborhood* model incorporates three factors: (1) item position in the current session, (2) recency of past sessions, and (3) position of recommendable items in similar sessions. Decay functions are applied to assign higher weights to more recent interactions [Garg et al. 2019].
- `vstan`: This model integrates the mechanisms of both `stan` and `vsknn`. In addition to considering session recency and item position, it introduces a sequence-

aware scoring scheme and applies an IDF-based weighting strategy to penalize overly frequent items, where IDF (Inverse Document Frequency) is commonly used to reduce the influence of popular items in information retrieval [Ludewig et al. 2021].

Neural Network-Based Models represent the most recent and complex approaches for session-based recommendation. Two models were selected for our work:

- `gru4rec`: The first neural model proposed for session-based recommendation, employing Gated Recurrent Units (GRUs) to mitigate the vanishing gradient problem in sequential learning tasks. The model predicts the next likely item based on the sequential dynamics of the session [Hidasi and Karatzoglou 2018].
- `narm`: The *Neural Attentive Recommendation Machine* extends `gru4rec` by introducing a hybrid encoder equipped with an attention mechanism. This mechanism captures both sequential signals and item-level similarity to the last interaction. Candidate item scores are computed using a bilinear matching scheme over the unified session representation [Li et al. 2017].

All models were initially evaluated using their default hyperparameter settings. Subsequently, a comprehensive set of experiments was conducted, where each model was tested with a wide range of parameter configurations. The best-performing setup for each model and dataset was selected from the validation set. In some cases, default values were retained when no significant gains were observed with alternative configurations.

One of the key parameters adjusted in the framework was `remind_mode`, which determines the position of *reminder* items in the recommendation list. This parameter can be set to `top` or `end` (default). When set to `top`, *reminder* items are placed at the top of the recommendation list, giving them higher priority. In contrast, when set to `end`, *reminder* items are added to the bottom of the list, after the main recommendations. Following the general tuning procedure, both configurations were evaluated, and the selected setting corresponds to the best-performing option for each dataset. The `remind_mode` was set to `top` for JusBrasilRec, Xing, Trivago, and RetailRocket, as it performed better in these datasets. For Music4All and MIND, however, `remind_mode` was set to `end`, where it produced the best results.

Another important parameter adjusted was `remind_sessions_num`, which controls how many of the user's most recent sessions are used to select *reminder* items. While the default value is 6, we evaluated this parameter over a range from 1 to 10. For the Xing dataset, we found that using only the most recent session (`remind_sessions_num` = 1) yielded better results, likely due to its users' shorter interaction histories.

Finally, for `gru4rec`, the number of training epochs and batch size were adapted per dataset based on validation performance and convergence speed. Epoch values were evaluated in the range of 10 to 100, and batch sizes in $\{6, 16, 32, 64\}$, with the best-performing configuration selected for each dataset. The final setup consisted of 10 epochs for JusBrasilRec, 50 for Xing, and 30 for Music4All, Trivago, MIND, and RetailRocket. A batch size of 6 was used for Xing, 16 for MIND, and 32 for the remaining datasets.

3.5. Reminder Strategies

In our work, we have adopted the *reminder* strategies available in the *session-rec* framework: recency, session similarity, and hybrid. These *reminders* follow the concept of Adaptive Reminders [Lerche et al. 2016]. It is important to note that the session similarity and hybrid strategies are only available for Nearest Neighbors-Based Models within the framework, as they rely on computing similarity scores between sessions based on `knn`.

To formalize these techniques, we define the following notation. Let H_u represent the recent interaction history of a user u, extracted from their previous sessions. A set of candidate items H_I^u is sampled from H_u, which consists of sessions composed of interaction tuples H_V^u in the form of $\langle$item, action, timestamp$\rangle$. The type of action may vary according to the domain—such as clicks, viewing an item, adding a song to a playlist, or applying for a job—and provides contextual information for identifying meaningful *reminders* [Lerche et al. 2016].

The recency *reminder* strategy is based on the *IRec* (Item Recency) concept, which prioritizes items that have been recently interacted with by the user. The underlying idea is that items from recent interactions are more likely to be relevant for future recommendations, as they reflect the user's current interests and preferences. This approach assumes that user preferences evolve rapidly, and by prioritizing recent interactions, the recommendations stay aligned with the user's up-to-date preferences [Lerche et al. 2016].

To quantify the recency of interactions, the *timestamps* of interactions are used to calculate a *ranking* score score_{ui}^{IRec} for a user u and an item i. This score reflects the temporal relevance of an item based on the user's interaction history. Specifically, the score is determined by identifying the most recent interaction for the item, as defined in Equation 1:

$$\text{score}_{ui}^{IRec} = \max_{t \in H_u^V(i)} (time(t)) \tag{1}$$

where $H_u^V(i)$ is the set of interaction tuples in H_u^V related to item i, and $time()$ is a function that returns the timestamp of a past interaction tuple t.

The session similarity *reminder* strategy is based on the *SSim* (Session Similarity) technique, which aims to identify past sessions of a user that exhibit similar intent to the current session. This approach assumes that if previous sessions contain items semantically close to those in the current session, then other items from those sessions may also be of interest [Lerche et al. 2016].

To compute session similarity, the cosine similarity between items in the current session C_u and items in a past session V_s is averaged, as shown in Equation 2:

$$\text{sim}_{session}(u, s) = \frac{\sum_{i \in V_s} \sum_{j \in C_u} \text{sim}_{cos}(i, j)}{|V_s| \cdot |C_u|} \tag{2}$$

where s is a past session of user u, V_s is the set of items in session s, and C_u is the set of

items in the current session.

The top k most similar sessions, denoted $\text{top}_k^S(u)$, are selected. The union of items from these sessions, $items(\text{top}_k^S(u))$, forms the candidate set. To rank the candidate items, the *IInt* (Interaction Intensity) score is used, which reflects how frequently an item was viewed in the user's interaction history:

$$\text{score}_{ui}^{SSim} = \text{score}_{ui}^{IInt} \cdot \mathbf{1}_{items(\text{top}_k^S(u))}(i) \tag{3}$$

The *IInt* score for a user u and an item i is defined as the number of past interactions with item i, as given in Equation 4:

$$\text{score}_{ui}^{IInt} = |H_u^V(i)| \tag{4}$$

This scoring assumes that a higher frequency of interaction with an item indicates a stronger user interest. In cases where multiple items receive the same *IInt* score, the *IRec* score can be used to break ties based on the most recent interaction timestamps [Lerche et al. 2016].

The hybrid *reminder* strategy combines the strengths of both *recency*-based and *session similarity*-based approaches by integrating their respective *reminder* scores into a unified recommendation score. This fusion aims to capture not only the immediate context of the user's current session but also historical behavioral patterns across previous sessions [Lerche et al. 2016].

Internally, the hybrid strategy computes separate *reminder* scores using the *IRec* and *SSim* techniques. Each score is then integrated with the base recommendation score (from the main model) using a weighted sum, as shown in Equation 5:

$$\text{score}_{ui}^{Hybrid} = \sum_{x \in \{base, IRec, SSim\}} w_x \cdot \text{score}_{ui}^{x} \tag{5}$$

where each component $x \in \{\textit{base}, \textit{IRec}, \textit{SSim}\}$ is associated with a weight w_x that controls its contribution to the final score. To ensure comparability across components, all scores score_{ui}^{x} are normalized by their respective maximum values prior to aggregation.

This strategy provides a flexible mechanism to adapt the recommendation list to both short-term user interests (via recency) and longer-term preferences (via session similarity), which can be particularly effective in dynamic environments with recurrent users [Lerche et al. 2016].

In this work, the weights associated with the hybrid *reminder* strategy were fixed to equal values, assigning the same contribution to both recency and session similarity components. This choice was made to ensure a balanced integration of short-term and session-level signals, while avoiding additional hyperparameter tuning and maintaining a consistent comparison across models and datasets.

The session similarity and hybrid *reminder* strategies involve computing cosine similarity between items across sessions (Equation 2) and performing nearest neighbor search to identify the top-k similar sessions. The similarity computation requires pairwise comparisons between items in the current session and past sessions, leading to a computational cost proportional to $O(|C_u| \cdot |V_s|)$ per session pair. Additionally, retrieving the top-k similar sessions requires evaluating multiple candidate sessions, which may increase computational overhead. However, these operations are restricted to a limited set of recent user sessions, keeping the method tractable in practice.

3.6. Evaluation Setup

SBRS aim to generate a ranked list of items based on a specific session, and their effectiveness is evaluated by how well they predict items within that session. To simulate the user's journey, all items are withheld and revealed iteratively. Specifically, the task focuses on predicting the next item given the first n interactions [Ludewig and Jannach 2018].

Two evaluation scenarios were adopted in our work: Next Item and Rest of Session.

In the Next Item scenario, the model receives the first n interactions of a session and must predict the $(n+1)$-th item. The value of n is incrementally increased to reflect different points within the session, simulating how the model performs at various stages of a user's navigation [Ludewig and Jannach 2018]. For each prediction, we compute Hit Rate and Mean Reciprocal Rank (MRR).

Hit Rate@k measures the proportion of sessions in which the ground-truth next item appears among the top-k recommendations. It is computed per session as a binary metric and averaged over all sessions, ignoring the exact rank within the top-k (Equation 6). From a user perspective, this metric reflects whether the system is able to present at least one relevant item within the visible recommendations, which is essential for a satisfactory experience [Hidasi et al. 2015, Hidasi and Karatzoglou 2018].

$$\text{Hit Rate@}k = \frac{\#\textit{Sessions with at least one relevant item in top-}k}{\textit{Total }\#\textit{Sessions}} \tag{6}$$

MRR@k evaluates the average reciprocal rank of the first relevant item in the recommendation list. For each session, it considers the position where the first relevant item appears, rewarding higher placements with larger reciprocal values. The final score is obtained by averaging these reciprocal ranks across all sessions, thus reflecting the model's ability to prioritize relevant items early in the ranking (Equation 7). This metric reflects how quickly a relevant item can be found, reducing the effort required to locate useful recommendations [Hidasi et al. 2015, Ludewig and Jannach 2018].

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\textit{rank}_i} \tag{7}$$

where *rank*$_i$ denotes the position of the first relevant item in the i-th session, and N is the total number of sessions.

In the Rest of Session scenario, the task is to predict all remaining items in the session after the first n interactions [Ludewig and Jannach 2018]. Here, we evaluate the ranked output using Precision, Recall, NDCG (Normalized Discounted Cumulative Gain), and MAP (Mean Average Precision).

Precision@k measures the ability of a model to generate correct predictions within the top-k recommended items. It is computed as the number of relevant items in the top-k recommendations divided by k, as shown in Equation 8. This metric reflects the proportion of useful items presented to the user, indicating how focused the recommendation list is on relevant content [Hidasi and Karatzoglou 2018].

$$\text{Precision@}k = \frac{\textit{Number of relevant items in the top-}k\textit{ recommendations}}{k} \tag{8}$$

Recall@k measures the fraction of relevant items that appear in the top-k recommendations. It captures how many of the relevant items are successfully retrieved by the model, reflecting its ability to avoid missing useful suggestions [Hidasi and Karatzoglou 2018], as shown in Equation 9.

$$\text{Recall@}k = \frac{\textit{Number of relevant items in the top-}k\textit{ recommendations}}{\textit{Total number of relevant items}} \tag{9}$$

NDCG@k evaluates the ranking quality of the top-k recommendations by assigning a higher weight to relevant items appearing earlier in the list. This metric emphasizes the importance of placing more relevant items at higher positions in the ranking [Hidasi and Karatzoglou 2018]. It is computed as the ratio of the Discounted Cumulative Gain (DCG@k) to the Ideal Discounted Cumulative Gain (IDCG@k), as shown in Equation 10.

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{IDCG@}k} \tag{10}$$

where DCG@k represents the discounted cumulative gain of the top-k recommended items, calculated as:

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{\text{relevance}_i}{\log_2(i+1)} \tag{11}$$

and IDCG@k corresponds to the maximum possible DCG@k achievable with a perfect ranking of the relevant items. In Equation 11, relevance$_i$ denotes the relevance of the item at position i.

MAP@k evaluates the overall ranking quality by aggregating precision scores at the positions of relevant items across all sessions [Ludewig and Jannach 2018]. Unlike Precision@k, it also considers the order of relevant items, providing a more comprehensive assessment of recommendation quality by rewarding rankings where relevant items appear earlier [Hidasi and Karatzoglou 2018].

For a single session, the Average Precision (AP) is computed as the mean of precision values at the ranks where each relevant item appears in the top-k recommendations, as shown in Equation 12.

$$\text{AP} = \frac{\sum_{k=1}^{n} \textit{Precision@k} \times \textit{Indicator}(\textit{relevant item at rank } k)}{\textit{Total number of relevant items in the session}} \tag{12}$$

Here, Precision@k is the precision at cutoff k, and the Indicator function equals 1 if the k-th item is relevant, and 0 otherwise.

MAP@k is then obtained by averaging the AP values over all N sessions, as shown in Equation 13:

$$\text{MAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{13}$$

where AP_i is the Average Precision for the i-th session.

MAP@k aggregates precision scores across all correct predictions, offering a comprehensive view of ranking quality [Ludewig and Jannach 2018].

In addition to accuracy-based metrics, we also consider Coverage and Popularity Bias, which capture aspects related to the variety and distribution of recommendations. Coverage measures the fraction of unique items from the catalog that appear in recommendation lists, while Popularity Bias reflects the extent to which the model favors popular items [Jannach et al. 2015, Jannach and Ludewig 2017].

To compute Coverage, a recommendation list is generated for each test session using the model. All items recommended at least once across all sessions are collected, and the number of unique items is counted. Coverage is then calculated by dividing the number of unique recommended items by the total number of items in the catalog, as shown in Equation 14. A higher Coverage indicates that the model explores a larger portion of the item space, reducing the concentration on a limited set of items [Jannach et al. 2015, Jannach and Ludewig 2017].

$$\text{Coverage} = \frac{\textit{Number of unique recommended items}}{\textit{Total number of items in the catalog}} \tag{14}$$

Popularity Bias is computed in a two-step process. First, the popularity of each item in the catalog is determined by counting the number of occurrences in all training sessions. Min-max normalization is then applied to obtain a normalized popularity score between 0 and 1 for each item, as shown in Equation 15. Next, for each

test session, a recommendation list is generated using the model, and the mean normalized popularity of the recommended items is computed. Finally, the average of these scores across all recommendation lists is calculated to obtain the overall Popularity Bias (Equation 16). Lower Popularity Bias indicates a tendency to recommend less popular items, promoting a more balanced distribution of recommendations [Jannach et al. 2015, Jannach and Ludewig 2017].

$$\text{Normalized Popularity} = \frac{\textit{Item Popularity} - \textit{Min Popularity}}{\textit{Max Popularity} - \textit{Min Popularity}} \tag{15}$$

$$\text{Popularity Bias} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{K}\sum_{j=1}^{K}\textit{Normalized Popularity}_{ij}\right) \tag{16}$$

where N is the number of test sessions (i.e., one recommendation list per session) and K is the number of items in each recommendation list.

All evaluations were conducted at a cutoff of $k = 10$ (i.e., the top 10 recommendations), following common practice in recommender systems to limit cognitive load and reflect realistic interface constraints [Ludewig and Jannach 2018].

4. Results and Discussion

This section presents the experimental results obtained on the six datasets used in our work. For clarity and organization, we report the results for each dataset in separate subsections.

Throughout this section, we refer to the *reminders* versions of the models using a suffix that indicates the strategy employed: `-recency` for models that incorporate the recency strategy, `-session` for those based on session similarity, and `-hybrid` for models that combine the hybrid strategy.

Tables 3 to 14 summarize the averaged results for the JusBrasilRec, Xing, Music4All, Trivago, MIND and RetailRocket datasets. For each dataset, we present two tables: one for the Next Item scenario, Coverage and Popularity Bias, and another for the Rest of Session scenario.

The best-performing values for each evaluation metric are highlighted in bold. In case of ties, the value with the smallest standard deviation was chosen; if both the mean and the standard deviation are tied, all tied values are highlighted. If all results are identical, none is highlighted.

A two-sided paired t-test was performed between each *reminder*-based model and its corresponding baseline model across all temporal slices to assess statistical significance. The resulting p-values are explicitly reported in the tables and can be interpreted according to conventional thresholds.

In addition to the tables, Figures 3 to 8 present heatmaps depicting the percentage difference of each *reminder*-based model relative to its baseline for all metrics. In all

figures, green represents improvement, while red represents a decrease relative to the baseline.

Across all datasets, models incorporating *reminders* frequently achieved performance that was competitive with or superior to their respective baselines. We discuss the results for each dataset in the following subsections.

All experiments were conducted on a machine equipped with an Intel Core i9-13900HX processor and 32 GB of RAM. In terms of computational performance, the evaluated *reminder* strategies introduced only a marginal overhead compared to the baseline methods. Prior work [Lerche et al. 2016] reports that *reminder*-based approaches can generate recommendation lists in less than 10 ms per session, and our observations indicate similar response times in practice, as no substantial impact on response time was observed. These findings suggest that incorporating *reminder* mechanisms does not significantly increase runtime complexity.

4.1. JusBrasilRec

As shown in Tables 3 and 4, incorporating *reminder*-based strategies led to consistent improvements on the JusBrasilRec dataset. The most significant gains occurred in Coverage and Popularity Bias, particularly for the `vstan` *reminder*-based strategies. These variants reached Coverage values up to 0.952 and Popularity Bias between 0.068 and 0.075, compared to 0.448 and 0.018 for the baseline. As depicted in Figure 3, these correspond to relative gains of approximately 79.5% to 112.5% in Coverage and 277.8% to 316.7% in Popularity Bias.

Table 3. Results for the JusBrasilRec dataset in the Next Item scenario, Coverage and Popularity Bias.

	Hit Rate			**MRR**			**Coverage**			**Pop. Bias**		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.747	0.002	-	**0.578**	0.002	-	0.794	0.009	-	0.083	0.008	-
stan-recency	0.754	0.005	0.01	0.349	0.129	0.01	0.885	0.047	0.01	**0.086**	0.008	0.22
stan-session	**0.755**	0.003	0.01	0.277	0.008	0.01	**0.940**	0.023	0.01	0.085	0.008	0.01
stan-hybrid	0.747	0.005	0.05	0.517	0.139	0.05	0.827	0.068	0.36	0.083	0.008	0.54
vstan (baseline)	0.743	0.003	-	**0.611**	0.005	-	0.448	0.121	-	0.018	0.001	-
vstan-recency	**0.754**	0.001	0.01	0.292	0.004	0.01	0.908	0.006	0.01	0.073	0.006	0.01
vstan-session	0.751	0.002	0.01	0.274	0.004	0.01	**0.952**	0.003	0.01	**0.075**	0.006	0.01
vstan-hybrid	0.741	0.002	0.08	0.589	0.002	0.01	0.804	0.009	0.01	0.068	0.006	0.01
vsknn (baseline)	0.745	0.002	-	0.563	0.002	-	0.796	0.010	-	0.087	0.009	-
vsknn-recency	**0.757**	0.002	0.01	0.289	0.004	0.01	**0.908**	0.006	0.01	**0.088**	0.008	0.67
vsknn-session	0.755	0.002	0.01	0.541	0.002	0.01	0.803	0.009	0.01	0.086	0.009	0.01
vsknn-hybrid	0.745	0.002	0.29	**0.564**	0.002	0.01	0.796	0.01	0.01	0.087	0.009	0.01
gru4rec (baseline)	0.68	0.002	-	**0.523**	0.004	-	0.92	0.005	-	0.042	0.004	-
gru4rec-recency	**0.708**	0.001	0.01	0.278	0.003	0.01	**0.955**	0.003	0.01	**0.055**	0.004	0.01
narm (baseline)	0.718	0.003	-	**0.541**	0.006	-	0.937	0.006	-	0.081	0.008	-
narm-recency	**0.735**	0.003	0.01	0.284	0.003	0.01	**0.965**	0.004	0.01	**0.082**	0.008	0.01
sr (baseline)	0.665	0.003	-	**0.523**	0.004	-	0.772	0.010	-	0.072	0.008	-
sr-recency	**0.699**	0.002	0.01	0.275	0.003	0.01	**0.901**	0.006	0.01	**0.077**	0.007	0.01

These results suggest that *reminder*-based strategies substantially increase the diversity of recommended items and reduce the dominance of a small subset of highly

Table 4. Results for the JusBrasilRec dataset in the Rest of Session scenario.

	Precision			Recall			NDCG			MAP		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.105	0.000	-	0.481	0.007	-	**0.460**	0.005	-	0.054	0.001	-
stan-recency	**0.106**	0.001	0.01	0.488	0.009	0.01	0.357	0.055	0.01	0.054	0.001	0.01
stan-session	**0.106**	0.001	0.01	**0.489**	0.007	0.01	0.325	0.006	0.01	**0.055**	0.001	0.01
stan-hybrid	0.105	0.001	0.01	0.482	0.009	0.2	0.432	0.061	0.25	0.054	0.001	0.28
vstan (baseline)	0.097	0.000	-	**0.494**	0.004	-	**0.479**	0.003	-	0.054	0.000	-
vstan-recency	**0.106**	0.000	0.01	0.488	0.007	0.16	0.332	0.004	0.01	**0.055**	0.001	0.10
vstan-session	0.105	0.001	0.01	0.486	0.007	0.05	0.323	0.004	0.01	0.054	0.001	0.63
vstan-hybrid	0.103	0.000	0.01	0.475	0.007	0.01	0.458	0.005	0.01	0.053	0.001	0.11
vsknn (baseline)	0.105	0.000	-	0.481	0.007	-	0.456	0.005	-	0.054	0.001	-
vsknn-recency	**0.108**	0.001	0.01	**0.493**	0.007	0.01	0.332	0.004	0.01	**0.055**	0.001	0.01
vsknn-session	0.107	0.000	0.01	0.489	0.007	0.01	**0.458**	0.006	0.01	**0.055**	0.001	0.01
vsknn-hybrid	0.105	0.000	0.96	0.481	0.007	0.01	0.456	0.005	0.01	0.054	0.001	0.97
gru4rec (baseline)	0.093	0.000	-	0.434	0.006	-	**0.414**	0.005	-	0.048	0.001	-
gru4rec-recency	**0.099**	0.000	0.01	**0.458**	0.006	0.01	0.314	0.003	0.01	**0.051**	0.001	0.01
narm (baseline)	0.100	0.000	-	0.461	0.006	-	**0.437**	0.004	-	0.051	0.001	-
narm-recency	**0.104**	0.000	0.01	**0.477**	0.006	0.01	0.323	0.003	0.01	**0.053**	0.001	0.01
sr (baseline)	0.090	0.000	-	0.425	0.007	-	**0.409**	0.006	-	0.047	0.001	-
sr-recency	**0.097**	0.000	0.01	**0.453**	0.006	0.01	0.311	0.003	0.01	**0.050**	0.001	0.01

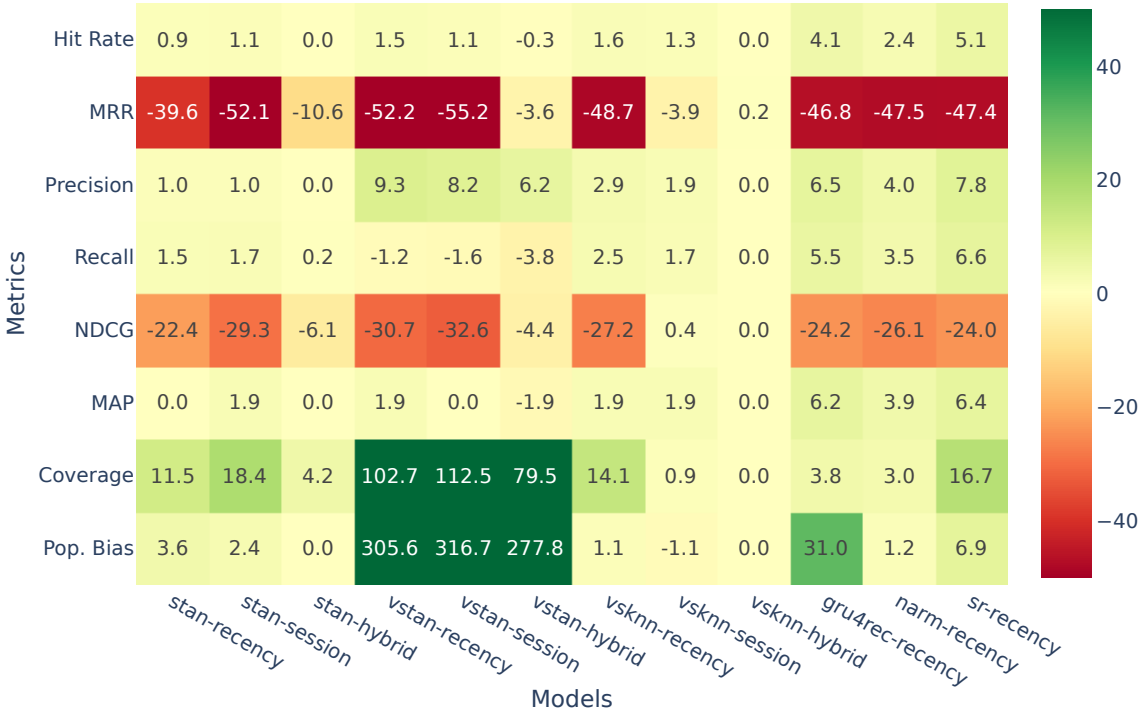

Figure 3. Percent change of each *reminder*-based model relative to its baseline for the JusBrasilRec dataset.

frequent items. This behavior is particularly relevant in the legal domain, where user interactions are typically goal-oriented and involve exploring multiple related documents within short sessions, rather than repeatedly consuming the same popular items.

Improvements were also observed across the `stan` and `vsknn` *reminder*-based strategies, achieving Coverage gains of up to 18.4% and 14.1%, respectively, compared to their baselines. Additionally, `vsknn-recency` and `vsknn-session` improved Hit Rate, Precision, and Recall, suggesting an increased likelihood of retrieving relevant items without restricting the recommendation space. The `gru4rec-recency` strategy obtained the highest Coverage (0.955), with a moderate increase in Popularity Bias (0.055), further reinforcing the trend toward broader item exploration.

Despite these improvements, metrics focused on ranking quality, such as MRR and NDCG, showed limited variation. The baseline `vstan` still achieved the highest MRR (0.611), indicating that *reminder*-based strategies primarily affect Coverage and Popularity Bias rather than ranking precision. Most improvements were statistically significant, except for a few hybrid variants that did not differ meaningfully from their baselines.

4.2. Xing

The impact of *reminders* was particularly pronounced in the Xing dataset, as shown in Tables 5 and 6. Coverage reached its peak for `gru4rec-recency` (0.997), followed by Hit Rate for `vstan-session` (0.349). Recall was highest for `vstan-recency` and `vstan-session` (0.258), Precision peaked for `vsknn-recency` (0.066), and Popularity Bias was highest for `vsknn` (0.113).

Ranking-oriented metrics remained limited. MRR scores were generally low, with the highest values observed for `vstan` and `vstan-hybrid` (0.193), followed by `stan` and `stan-hybrid` (0.184). Most other *reminder*-based models, especially recency and session similarity strategies, achieved lower MRR, with some even underperforming their baselines. NDCG improved for `sr-recency` (0.171), `narm-recency` (0.167), and `gru4rec-recency` (0.172) compared to 0.125, 0.097, and 0.121, respectively, but overall values remained modest. MAP reached 0.029 for `gru4rec-recency`, `sr-recency`, `vsknn-recency`, and `vstan-recency/session`, reflecting moderate gains over baselines.

These results suggest that, in the employment domain, recency may act as a strong signal for next-item prediction, with recent interactions playing a central role in user behavior. At the same time, the limited improvements in ranking-oriented metrics indicate that refining the exact ordering of recommended items has a smaller impact compared to ensuring that relevant items are included in the recommendation list.

One possible explanation is the sparse and episodic nature of user interactions in job search scenarios, where users tend to engage in short bursts of activity with less consistent preference patterns. This may contribute to the relatively low performance in ranking-specific metrics, even when Coverage and recall-oriented metrics improve. Again, most improvements were statistically significant, although some hybrid models showed no meaningful differences relative to their baselines.

Figure 4 shows that `gru4rec-recency`, `narm-recency`, and `sr-recency` generally outperformed their baselines, particularly in MAP, Precision, Recall, and NDCG, albeit with varying magnitudes across metrics.

Table 5. Results for the Xing dataset in the Next Item scenario, Coverage and Popularity Bias.

	Hit Rate			MRR			Coverage			Pop. Bias		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.311	0.019	-	**0.184**	0.012	-	0.985	0.004	-	0.097	0.068	-
stan-recency	0.329	0.026	0.02	0.140	0.009	0.01	**0.990**	0.004	0.01	0.098	0.068	0.34
stan-session	**0.332**	0.024	0.01	0.142	0.008	0.01	**0.990**	0.004	0.01	**0.099**	0.068	0.10
stan-hybrid	0.311	0.019	0.01	**0.184**	0.012	0.37	0.985	0.004	0.01	0.097	0.068	0.38
vstan (baseline)	0.31	0.017	-	**0.193**	0.011	-	0.991	0.002	-	0.088	0.065	-
vstan-recency	0.345	0.020	0.01	0.128	0.009	0.01	**0.995**	0.002	0.01	**0.093**	0.064	0.09
vstan-session	**0.349**	0.018	0.01	0.133	0.008	0.01	0.994	0.002	0.01	**0.093**	0.064	0.02
vstan-hybrid	0.310	0.017	0.01	**0.193**	0.011	0.01	0.991	0.002	0.01	0.088	0.065	0.01
vsknn (baseline)	0.311	0.015	-	**0.175**	0.011	-	0.989	0.002	-	**0.113**	0.077	-
vsknn-recency	**0.341**	0.016	0.01	0.137	0.029	0.02	**0.994**	0.003	0.01	0.111	0.076	0.10
vsknn-session	0.322	0.012	0.01	**0.175**	0.010	0.71	0.989	0.002	0.34	0.111	0.076	0.04
vsknn-hybrid	0.311	0.015	0.37	**0.175**	0.011	0.44	0.989	0.002	0.01	**0.113**	0.077	0.13
gru4rec (baseline)	0.211	0.004	-	**0.112**	0.004		0.980	0.001	-	0.08	0.059	-
gru4rec-recency	**0.277**	0.017	0.01	0.106	0.009	0.17	**0.997**	0.001	0.01	**0.098**	0.063	0.01
narm (baseline)	0.172	0.013	-	0.087	0.007		0.928	0.01	-	0.107	0.074	-
narm-recency	**0.261**	0.019	0.01	**0.103**	0.008	0.01	**0.992**	0.003	0.01	**0.109**	0.070	0.60
sr (baseline)	0.211	0.005	-	**0.114**	0.003		0.966	0.004	-	0.091	0.067	-
sr-recency	**0.278**	0.017	0.01	0.105	0.008	0.08	**0.995**	0.001	0.01	**0.103**	0.067	0.01

Table 6. Results for the Xing dataset in the Rest of Session scenario.

	Precision			Recall			NDCG			MAP		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.053	0.003	-	0.218	0.017	-	0.176	0.013	-	0.024	0.002	-
stan-recency	**0.059**	0.004	0.01	0.239	0.023	0.01	**0.191**	0.012	0.01	**0.027**	0.003	0.01
stan-session	**0.059**	0.003	0.01	**0.240**	0.023	0.01	**0.191**	0.012	0.01	**0.027**	0.003	0.01
stan-hybrid	0.053	0.003	0.37	0.218	0.017	0.37	0.176	0.013	0.31	0.024	0.002	0.37
vstan (baseline)	0.053	0.003	-	0.216	0.016	-	0.177	0.012	-	0.024	0.002	-
vstan-recency	**0.065**	0.003	0.01	**0.258**	0.017	0.01	0.189	0.012	0.04	**0.029**	0.002	0.01
vstan-session	**0.065**	0.002	0.01	**0.258**	0.017	0.01	**0.191**	0.011	0.01	**0.029**	0.002	0.01
vstan-hybrid	0.053	0.003	0.01	0.216	0.016	0.01	0.177	0.012	0.01	0.024	0.002	0.01
vsknn (baseline)	0.057	0.002	-	0.223	0.015	-	0.178	0.012	-	0.025	0.002	-
vsknn-recency	**0.066**	0.005	0.01	**0.256**	0.016	0.01	**0.190**	0.012	0.05	**0.029**	0.002	0.02
vsknn-session	0.059	0.002	0.02	0.232	0.012	0.02	**0.188**	0.008	0.01	0.026	0.001	0.01
vsknn-hybrid	0.057	0.002	0.37	0.223	0.015	0.37	0.178	0.012	0.33	0.025	0.002	0.64
gru4rec (baseline)	0.038	0.001	-	0.150	0.004	-	0.121	0.003	-	0.016	0	-
gru4rec-recency	**0.065**	0.004	0.01	**0.236**	0.013	0.01	**0.172**	0.011	0.01	**0.029**	0.002	0.01
narm (baseline)	0.033	0.002	-	0.125	0.01	-	0.097	0.006	-	0.014	0.001	-
narm-recency	**0.062**	0.004	0.01	**0.226**	0.013	0.01	**0.167**	0.011	0.01	**0.027**	0.002	0.01
sr (baseline)	0.040	0.001	-	0.153	0.006	-	0.125	0.004	-	0.017	0.001	-
sr-recency	**0.065**	0.004	0.01	**0.236**	0.013	0.01	**0.171**	0.011	0.01	**0.029**	0.002	0.01

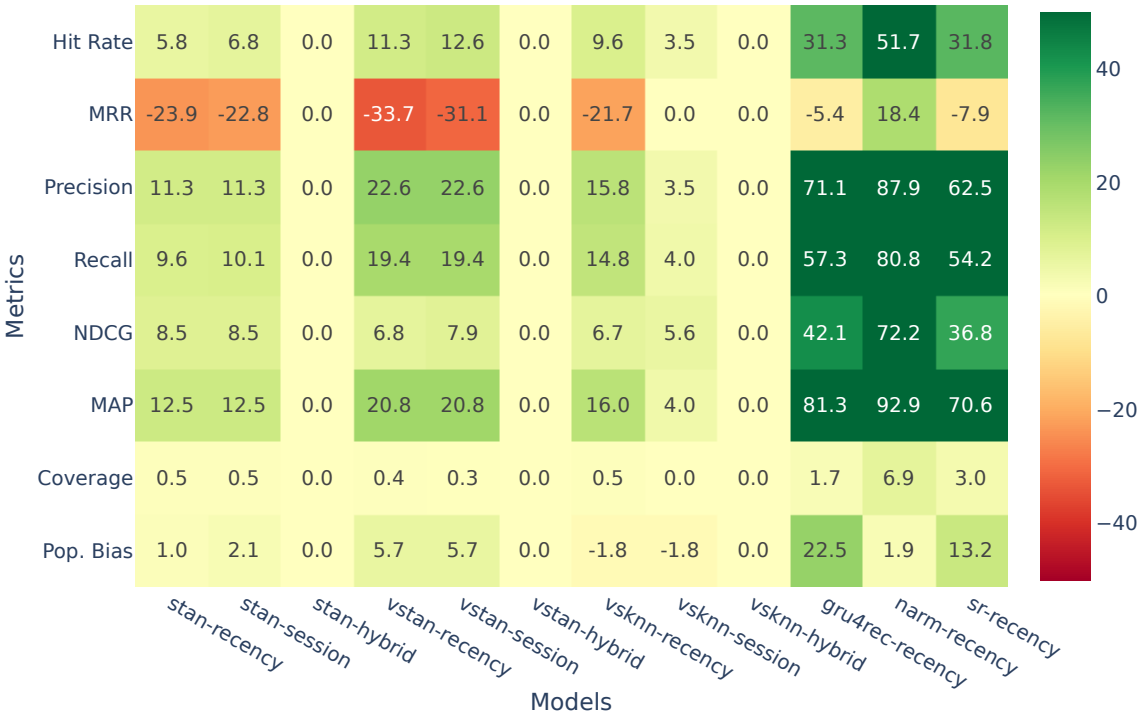

Figure 4. Percent change of each *reminder*-based model relative to its baseline for the Xing dataset.

4.3. Music4All

The results on the Music4All dataset indicate that all recommendation models achieved relatively low performance, as shown in Tables 7 and 8. Among the models, `stan-recency`, `vsknn-recency`, and `narm` reached slightly higher Hit Rates and Coverage, while `vsknn` exhibited the highest Popularity Bias, although differences were minor. As depicted in Figure 5, improvements over the baselines were generally small, with modest gains observed mainly in MAP, while Popularity Bias occasionally worsened for some *reminder*-based strategies.

Overall, most improvements were minor and not statistically significant, suggesting that the highly diverse and less predictable nature of user interactions in music consumption limits the effectiveness of *reminder*-based models. In particular, users may exhibit broader and more exploratory listening patterns, reducing the impact of recent interactions as a strong signal for future recommendations. As a result, strategies based on *reminders* may provide limited benefits compared to other domains, where user behavior tends to be more structured.

4.4. Trivago

The results on the Trivago dataset indicate that, overall, recommendation performance was moderate across all models, as shown in Tables 9 and 10. Hit Rate remained relatively high (0.706–0.863), suggesting that models were often able to retrieve relevant items, whereas MRR and NDCG varied more widely (0.303–0.743 and 0.333–0.420, respectively), indicating greater difficulty in accurately ranking these items.

Table 7. Results for the Music4All dataset in the Next Item scenario, Coverage and Popularity Bias.

	Hit Rate			MRR			Coverage			Pop. Bias		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.478	0.032	-	0.230	0.016	-	0.828	0.029	-	**0.096**	0.055	-
stan-recency	**0.479**	0.033	0.08	**0.232**	0.016	0.01	**0.831**	0.029	0.01	0.088	0.048	0.04
stan-session	**0.479**	0.033	0.08	**0.232**	0.016	0.01	**0.831**	0.029	0.01	0.088	0.048	0.04
stan-hybrid	**0.479**	0.033	0.08	**0.232**	0.016	0.01	0.830	0.029	0.01	0.088	0.048	0.04
vstan (baseline)	0.481	0.032	-	0.241	0.016	-	0.892	0.023	-	0.061	0.040	-
vstan-recency	0.481	0.032	0.01	**0.241**	0.015	0.78	0.892	0.023	0.45	0.061	0.040	0.38
vstan-session	0.481	0.032	0.01	**0.241**	0.015	0.78	0.892	0.023	0.45	0.061	0.040	0.38
vstan-hybrid	0.481	0.032	0.01	0.241	0.016	0.01	0.892	0.023	0.01	0.061	0.040	0.01
vsknn (baseline)	0.461	0.026	-	0.211	0.014	-	0.754	0.033	-	**0.121**	0.073	-
vsknn-recency	**0.464**	0.025	0.01	**0.212**	0.013	0.77	0.759	0.032	0.01	0.114	0.070	0.01
vsknn-session	**0.464**	0.025	0.01	**0.212**	0.013	0.77	**0.760**	0.032	0.01	0.114	0.070	0.01
vsknn-hybrid	**0.464**	0.025	0.01	**0.212**	0.013	0.79	0.759	0.032	0.01	0.114	0.070	0.01
gru4rec (baseline)	0.460	0.031	-	0.322	0.025	-	0.949	0.010	-	0.036	0.026	-
gru4rec-recency	0.460	0.031	0.37	0.322	0.025	0.38	0.949	0.010	0.37	0.036	0.026	0.33
narm (baseline)	0.445	0.032	-	**0.296**	0.026	-	**0.812**	0.022	-	0.082	0.049	-
narm-recency	0.445	0.033	0.85	0.295	0.025	0.02	0.811	0.025	0.78	0.082	0.049	0.35
sr (baseline)	0.463	0.031	-	0.313	0.024	-	0.884	0.017	-	**0.080**	0.049	-
sr-recency	0.463	0.031	0.66	0.313	0.024	0.64	**0.885**	0.017	0.34	0.080	0.050	0.25

Table 8. Results for the Music4All dataset in the Rest of Session scenario.

	Precision			Recall			NDCG			MAP		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.160	0.020	-	0.251	0.028	-	0.249	0.023	-	0.043	0.006	-
stan-recency	0.159	0.020	0.14	0.251	0.028	0.55	**0.249**	0.022	0.71	0.043	0.006	0.81
stan-session	0.159	0.020	0.14	0.251	0.028	0.55	**0.249**	0.022	0.71	0.043	0.006	0.81
stan-hybrid	0.159	0.020	0.14	0.251	0.028	0.56	**0.249**	0.022	0.71	0.043	0.006	0.81
vstan (baseline)	0.158	0.019	-	0.252	0.027	-	0.253	0.021	-	0.043	0.006	-
vstan-recency	0.158	0.019	0.58	0.252	0.027	0.52	0.253	0.021	0.28	0.043	0.006	0.20
vstan-session	0.158	0.019	0.58	0.252	0.027	0.52	0.253	0.021	0.28	0.043	0.006	0.37
vstan-hybrid	0.158	0.019	0.01	0.252	0.027	0.01	0.253	0.021	0.01	0.043	0.006	0.01
vsknn (baseline)	0.170	0.021	-	0.255	0.022	-	0.252	0.021	-	0.045	0.005	-
vsknn-recency	**0.171**	0.021	0.01	**0.257**	0.022	0.01	**0.253**	0.020	0.07	**0.046**	0.005	0.01
vsknn-session	**0.171**	0.021	0.01	**0.257**	0.022	0.01	**0.253**	0.020	0.06	**0.046**	0.005	0.01
vsknn-hybrid	**0.171**	0.021	0.01	**0.257**	0.022	0.01	**0.253**	0.020	0.06	**0.046**	0.005	0.01
gru4rec (baseline)	0.139	0.013	-	0.227	0.018	-	0.256	0.018	-	0.037	0.003	-
gru4rec-recency	0.139	0.013	0.45	0.227	0.018	0.34	0.256	0.018	0.40	0.037	0.003	0.37
narm (baseline)	0.153	0.020	-	0.234	0.023	-	**0.263**	0.023	-	0.040	0.006	-
narm-recency	0.153	0.020	0.95	**0.235**	0.023	0.90	0.263	0.024	0.33	0.040	0.006	0.92
sr (baseline)	0.160	0.021	-	0.252	0.025	-	0.285	0.026	-	0.046	0.006	-
sr-recency	0.160	0.021	0.75	0.252	0.025	0.95	0.285	0.026	0.23	0.046	0.006	0.86

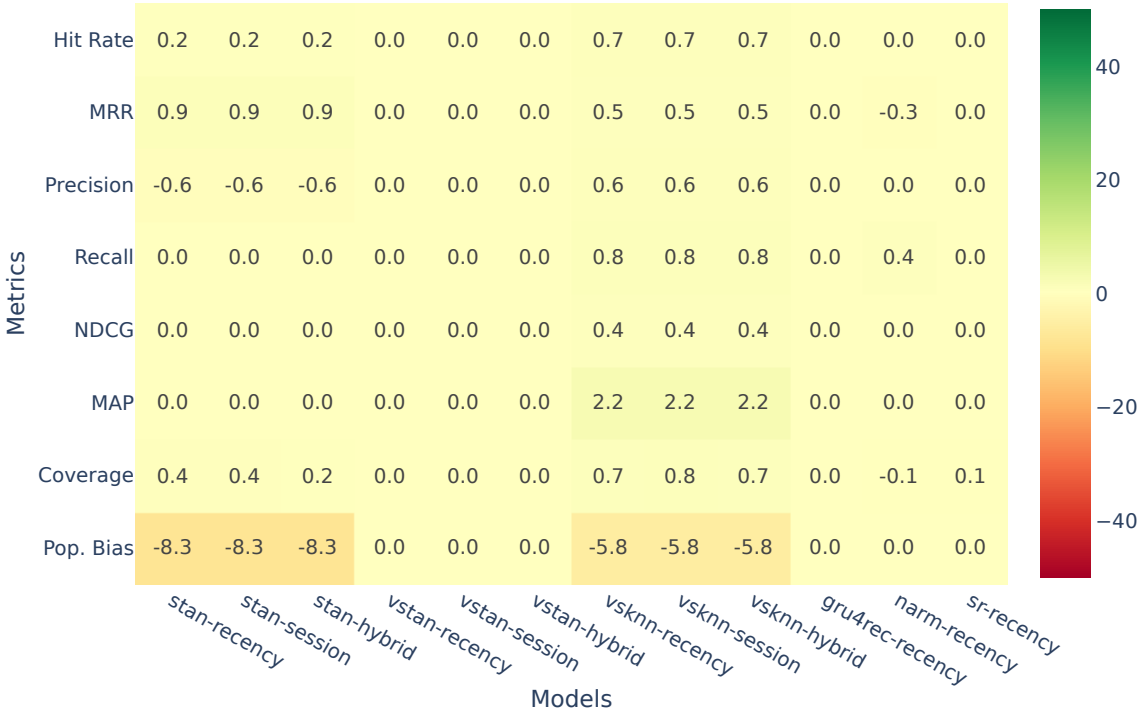

Figure 5. Percent change of each *reminder*-based model relative to its baseline for the Music4All dataset.

Table 9. Results for the Trivago dataset in the Next Item scenario, Coverage and Popularity Bias.

	Hit Rate			MRR			Coverage			Pop. Bias		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.847	0.005	-	**0.735**	0.004	-	0.924	0.006	-	0.109	0.015	-
stan-recency	0.862	0.005	0.01	0.441	0.010	0.01	0.961	0.004	0.01	0.109	0.016	0.30
stan-session	**0.863**	0.005	0.01	0.435	0.009	0.01	**0.965**	0.004	0.01	**0.110**	0.016	0.11
stan-hybrid	0.847	0.005	0.01	**0.735**	0.004	0.01	0.924	0.006	0.01	0.109	0.015	0.01
vstan (baseline)	0.846	0.005	-	**0.743**	0.004	-	0.925	0.006	-	0.106	0.015	-
vstan-recency	**0.861**	0.006	0.01	0.441	0.010	0.01	0.962	0.003	0.01	0.107	0.016	0.28
vstan-session	**0.861**	0.006	0.01	0.435	0.009	0.01	**0.965**	0.004	0.01	**0.108**	0.016	0.10
vstan-hybrid	0.846	0.005	0.01	**0.743**	0.004	0.01	0.925	0.006	0.01	0.106	0.015	0.01
vsknn (baseline)	0.843	0.006	-	**0.643**	0.006	-	0.926	0.006	-	**0.114**	0.016	-
vsknn-recency	**0.855**	0.006	0.01	0.425	0.010	0.01	**0.963**	0.003	0.01	0.114	0.017	0.58
vsknn-session	0.846	0.005	0.01	0.570	0.003	0.01	0.927	0.006	0.09	0.113	0.016	0.04
vsknn-hybrid	0.843	0.006	0.01	**0.643**	0.006	0.01	0.926	0.006	0.01	**0.114**	0.016	0.01
gru4rec (baseline)	0.752	0.004	-	**0.606**	0.005	-	0.977	0.002	-	0.098	0.016	-
gru4rec-recency	**0.790**	0.007	0.01	0.410	0.011	0.01	**0.990**	0.001	0.01	**0.101**	0.016	0.01
narm (baseline)	0.762	0.002	-	**0.589**	0.004	-	0.986	0.001	-	0.113	0.018	-
narm-recency	**0.794**	0.006	0.01	0.407	0.011	0.01	**0.992**	0.000	0.01	**0.113**	0.017	0.50
sr (baseline)	0.706	0.002	-	**0.591**	0.005	-	0.867	0.010	-	0.099	0.014	-
sr-recency	**0.760**	0.005	0.01	0.399	0.009	0.01	**0.940**	0.005	0.01	**0.103**	0.015	0.04

Table 10. Results for the Trivago dataset in the Rest of Session scenario.

	Precision			Recall			NDCG			MAP		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.131	0.001	-	0.378	0.013	-	**0.420**	0.010	-	0.044	0.002	-
stan-recency	**0.139**	0.001	0.01	0.392	0.014	0.01	0.377	0.012	0.01	**0.046**	0.002	0.01
stan-session	**0.139**	0.001	0.01	**0.393**	0.014	0.01	0.376	0.011	0.01	**0.046**	0.002	0.01
stan-hybrid	0.131	0.001	0.01	0.378	0.013	0.01	**0.420**	0.010	0.01	0.044	0.002	0.01
vstan (baseline)	0.130	0.001	-	0.376	0.014	-	**0.416**	0.010	-	0.043	0.002	-
vstan-recency	**0.138**	0.001	0.01	0.390	0.014	0.01	0.377	0.012	0.01	**0.046**	0.002	0.01
vstan-session	**0.138**	0.001	0.01	**0.391**	0.015	0.01	0.375	0.012	0.01	**0.046**	0.002	0.01
vstan-hybrid	0.130	0.001	0.01	0.376	0.014	0.01	**0.416**	0.010	0.01	0.043	0.002	0.01
vsknn (baseline)	0.130	0.002	-	0.375	0.014	-	**0.399**	0.010	-	0.043	0.002	-
vsknn-recency	**0.138**	0.002	0.01	**0.389**	0.014	0.01	0.365	0.011	0.01	**0.046**	0.002	0.01
vsknn-session	0.133	0.002	0.01	0.378	0.013	0.01	0.391	0.010	0.01	0.044	0.001	0.01
vsknn-hybrid	0.130	0.002	0.01	0.375	0.014	0.01	**0.399**	0.010	0.01	0.043	0.002	0.01
gru4rec (baseline)	0.112	0.001	-	0.332	0.012	-	**0.361**	0.010	-	0.037	0.001	-
gru4rec-recency	**0.125**	0.001	0.01	**0.359**	0.014	0.01	0.346	0.012	0.01	**0.041**	0.002	0.01
narm (baseline)	0.110	0.001	-	0.331	0.012	-	**0.356**	0.009	-	0.037	0.001	-
narm-recency	**0.123**	0.001	0.01	**0.358**	0.014	0.01	0.343	0.012	0.01	**0.041**	0.002	0.01
sr (baseline)	0.097	0.001	-	0.303	0.011	-	**0.340**	0.008	-	0.033	0.001	-
sr-recency	**0.116**	0.001	0.01	**0.341**	0.013	0.01	0.333	0.011	0.01	**0.039**	0.001	0.01

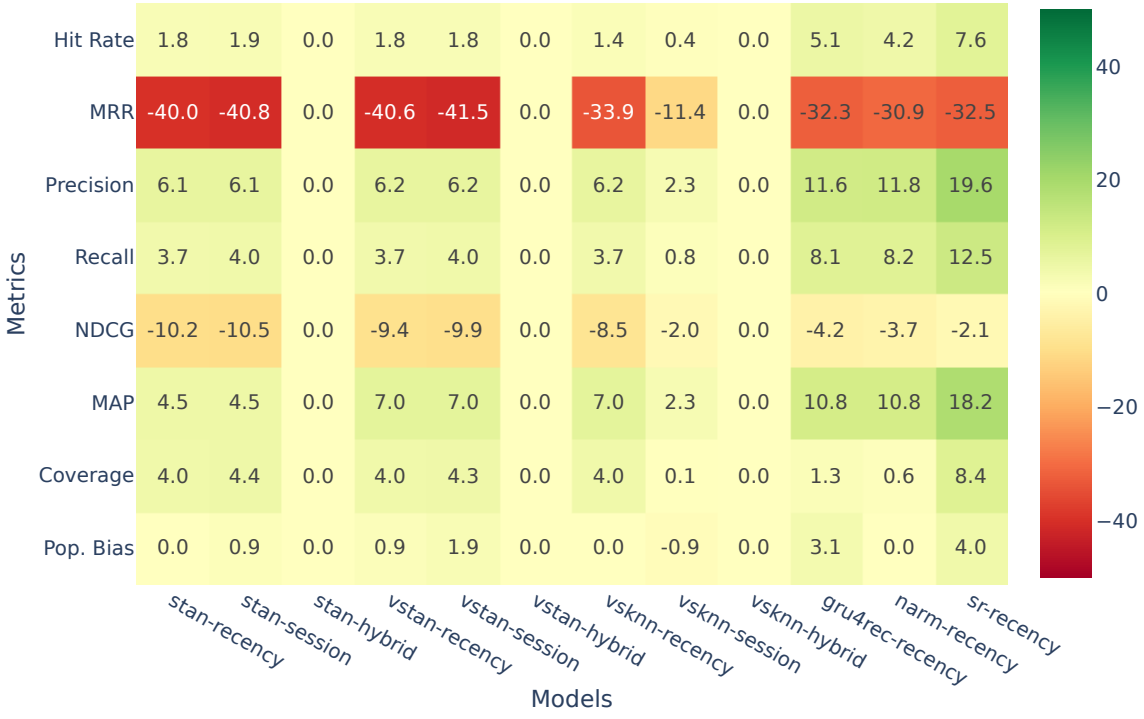

Figure 6. Percent change of each *reminder*-based model relative to its baseline for the Trivago dataset.

Recency- and session similarity *reminder* strategies led to modest improvements in Hit Rate and Coverage—for instance, `stan-session` increased Coverage from 0.924 to 0.965, and `vsknn-recency` raised Hit Rate from 0.843 to 0.855—while gains in MRR and NDCG remained limited. This suggests that *reminder*-based strategies are effective in reinforcing recently or contextually relevant items, but contribute less to refining their exact position in the ranking.

These effects also varied across models, indicating that the impact of *reminders* depends on how each algorithm leverages session and recency information. As depicted in Figure 6, most improvements were statistically significant, with the most noticeable gains observed for `sr`, `narm`, and `gru4rec`, particularly in MAP and with smaller increases in Precision and Recall. Overall, these results suggest that *reminder*-based strategies may provide consistent, though moderate, benefits in this domain.

4.5. MIND

The results on the MIND dataset indicate that all baseline and *reminder*-based models achieved very low performance, as shown in Tables 11 and 12. Hit Rate ranged from 0.054 for `gru4rec` to 0.221 for `narm`, while MRR remained between 0.015 and 0.091, with the lowest observed for `gru4rec` and the highest for `narm`. Precision values varied between 0.010 for `gru4rec` and 0.042 for `narm`, and Recall ranged from 0.054 to 0.221. Ranking-specific metrics were also limited, with NDCG spanning 0.029 for `gru4rec` to 0.150 for `narm`, and MAP ranging from 0.005 for `gru4rec` to 0.023 for `narm`. Coverage was highest for `gru4rec` (0.973) and lowest for `narm` (0.456), while Popularity Bias ranged from 0.019 to 0.237, indicating substantial differences in recommendation distribution across models.

Table 11. Results for the MIND dataset in the Next Item scenario, Coverage and Popularity Bias.

	Hit Rate			**MRR**			**Coverage**			**Pop. Bias**		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	**0.200**	0.024	-	**0.050**	0.006	-	0.901	0.005	-	**0.234**	0.053	-
stan-recency	0.189	0.023	0.02	0.048	0.006	0.04	**0.910**	0.005	0.01	0.208	0.051	0.01
stan-session	0.189	0.023	0.02	0.048	0.006	0.02	**0.910**	0.005	0.03	0.208	0.051	0.01
stan-hybrid	0.189	0.023	0.02	0.048	0.006	0.04	**0.910**	0.005	0.00	0.208	0.051	0.01
vstan (baseline)	0.176	0.021	-	0.045	0.005	-	**0.956**	0.005	-	0.162	0.042	-
vstan-recency	0.176	0.021	0.22	0.045	0.005	0.19	0.956	0.006	0.42	0.162	0.042	0.15
vstan-session	0.176	0.021	0.22	0.045	0.005	0.19	0.956	0.006	0.42	0.162	0.042	0.15
vstan-hybrid	0.176	0.021	0.22	0.045	0.005	0.18	**0.956**	0.005	0.42	0.162	0.042	0.18
vsknn (baseline)	**0.196**	0.023	-	0.050	0.006	-	0.887	0.006	-	**0.227**	0.053	-
vsknn-recency	0.194	0.022	0.01	0.050	0.006	0.09	**0.897**	0.007	0.01	0.221	0.053	0.01
vsknn-session	0.194	0.023	0.01	0.050	0.006	0.17	0.895	0.007	0.01	0.221	0.053	0.01
vsknn-hybrid	0.194	0.023	0.01	0.050	0.006	0.17	0.894	0.007	0.01	0.220	0.053	0.01
gru4rec (baseline)	0.054	0.011	-	0.015	0.004	-	**0.973**	0.003	-	0.019	0.006	-
gru4rec-recency	0.054	0.011	0.42	0.015	0.004	0.82	0.972	0.004	0.15	0.019	0.006	0.91
narm (baseline)	0.221	0.026	-	**0.091**	0.012	-	**0.456**	0.044	-	0.233	0.055	-
narm-recency	**0.221**	0.025	0.91	0.090	0.012	0.10	0.379	0.095	0.12	**0.237**	0.058	0.28
sr (baseline)	**0.197**	0.025	-	0.081	0.011	-	0.706	0.017	-	0.220	0.044	-
sr-recency	0.196	0.025	0.45	0.081	0.011	0.32	**0.718**	0.022	0.07	**0.221**	0.044	0.20

Table 12. Results for the MIND dataset in the Rest of Session scenario.

	Precision			Recall			NDCG			MAP		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	**0.036**	0.004	-	**0.200**	0.024	-	**0.111**	0.014	-	**0.020**	0.003	-
stan-recency	0.035	0.004	0.02	0.190	0.022	0.02	0.108	0.013	0.04	0.019	0.002	0.03
stan-session	0.035	0.004	0.04	0.190	0.022	0.02	0.108	0.013	0.02	0.019	0.002	0.04
stan-hybrid	0.035	0.004	0.02	0.190	0.022	0.02	0.108	0.013	0.04	0.019	0.002	0.03
vstan (baseline)	0.032	0.004	-	0.176	0.021	-	0.100	0.011	-	0.018	0.002	-
vstan-recency	0.032	0.004	0.25	0.176	0.021	0.28	0.100	0.011	0.26	0.018	0.002	0.28
vstan-session	0.032	0.004	0.25	0.176	0.021	0.28	0.100	0.011	0.26	0.018	0.002	0.28
vstan-hybrid	0.032	0.004	0.79	0.176	0.021	0.18	0.100	0.011	0.26	0.018	0.002	0.18
vsknn (baseline)	0.036	0.004	-	**0.197**	0.022	-	0.112	0.013	-	0.020	0.002	-
vsknn-recency	0.036	0.004	0.02	0.194	0.022	0.00	0.112	0.013	0.04	0.020	0.002	0.02
vsknn-session	0.036	0.004	0.02	0.194	0.023	0.00	0.112	0.013	0.07	0.020	0.002	0.04
vsknn-hybrid	0.036	0.004	0.02	0.194	0.023	0.00	0.112	0.013	0.07	0.020	0.002	0.04
gru4rec (baseline)	0.010	0.002	-	0.054	0.011	-	0.029	0.007	-	0.005	0.001	-
gru4rec-recency	0.010	0.002	0.10	0.054	0.011	0.58	0.029	0.007	0.64	0.005	0.001	0.43
narm (baseline)	0.042	0.005	-	0.221	0.025	-	**0.150**	0.018	-	0.023	0.003	-
narm-recency	0.042	0.005	0.40	0.221	0.025	0.81	0.149	0.017	0.35	0.023	0.003	0.20
sr (baseline)	0.036	0.005	-	0.197	0.025	-	0.133	0.017	-	0.020	0.003	-
sr-recency	0.036	0.005	0.33	0.197	0.025	0.49	0.133	0.017	0.49	0.020	0.003	0.51

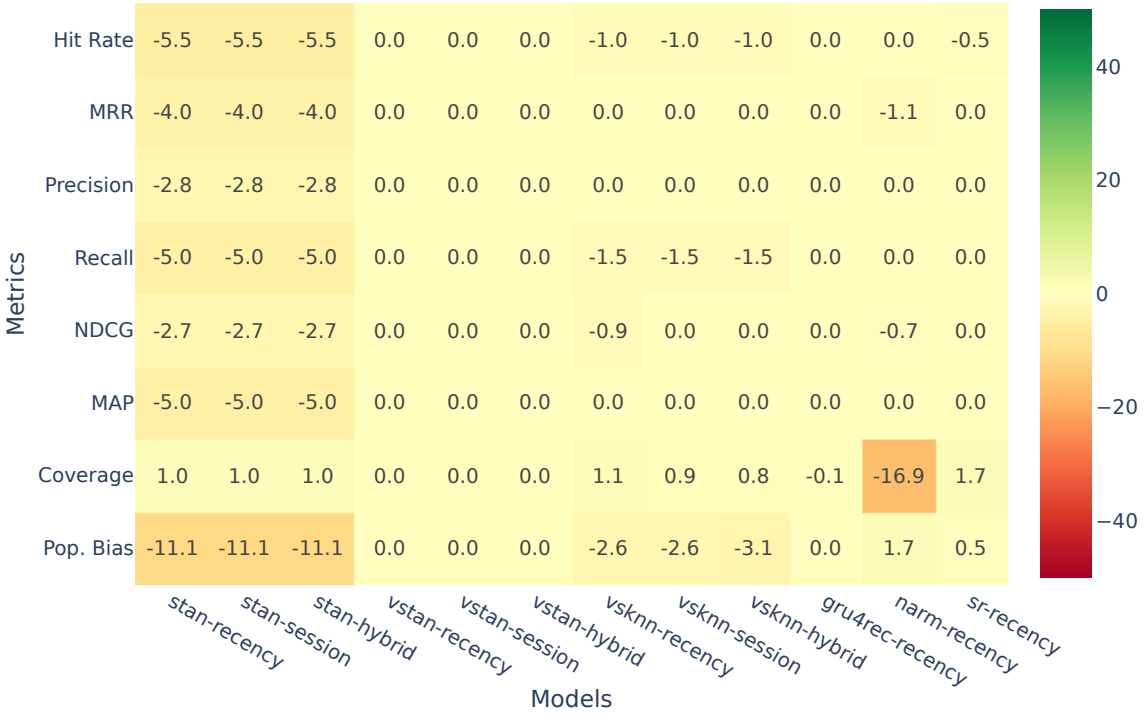

Figure 7. Percent change of each *reminder*-based model relative to its baseline for the MIND dataset.

As depicted in Figure 7, *reminder*-based strategies had minimal overall impact. Slight gains in Coverage were observed for all `vsknn` and `stan` strategies and for `sr-recency`, whereas `narm-recency` showed a decrease. Most `stan` strategies negatively affected Hit Rate, MRR, and other ranking measures, indicating that adding reminders did not consistently enhance model performance.

In summary, these results suggest that the highly dynamic and short-lived nature of user interactions with news articles may limit the effectiveness of *reminders*, where recent interactions may quickly become outdated, reducing their usefulness as signals for future recommendations. As a result, *reminder*-based strategies provide limited benefits, with most statistically significant differences reflecting small or even negative changes rather than meaningful improvements in recommendation quality.

4.6. RetailRocket

The results on the RetailRocket dataset, shown in Tables 13 and 14, highlight that recency *reminder*-based strategies achieved the highest performance across multiple metrics. Hit Rate peaked at 0.782–0.783 for `stan-recency`, `stan-session`, `vstan-recency`, `vstan-session`, and `vsknn-recency`, while Coverage reached a maximum of 0.997 for `gru4rec-recency`. Precision, Recall, NDCG, and MAP also showed their highest values for recency *reminder*-based strategies, with `gru4rec-recency` achieving 0.116, 0.531, 0.492, and 0.061, respectively.

Table 13. Results for the RetailRocket dataset in the Next Item scenario, Coverage and Popularity Bias.

	Hit Rate			MRR			Coverage			Pop. Bias		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.751	0.020	-	**0.561**	0.028	-	0.978	0.00	-	0.073	0.008	-
stan-recency	**0.782**	0.022	0.01	0.519	0.035	0.01	0.989	0.00	0.01	0.073	0.008	0.87
stan-session	**0.782**	0.023	0.01	0.518	0.032	0.01	**0.992**	0.00	0.01	**0.074**	0.008	0.63
stan-hybrid	0.751	0.020	0.01	**0.561**	0.028	0.01	0.978	0.00	0.01	0.073	0.008	0.01
vstan (baseline)	0.748	0.019	-	**0.561**	0.028	-	0.981	0.00	-	0.070	0.007	-
vstan-recency	**0.781**	0.022	0.01	0.509	0.035	0.01	0.990	0.00	0.01	0.070	0.008	0.77
vstan-session	**0.781**	0.022	0.01	0.508	0.032	0.01	**0.993**	0.00	0.01	**0.071**	0.008	0.38
vstan-hybrid	0.748	0.019	0.01	**0.561**	0.028	0.01	0.981	0.00	0.01	0.070	0.007	0.01
vsknn (baseline)	0.748	0.018	-	**0.552**	0.029	-	0.981	0.00	-	**0.080**	0.009	-
vsknn-recency	**0.782**	0.021	0.01	0.508	0.035	0.01	**0.992**	0.00	0.01	0.079	0.010	0.30
vsknn-session	0.764	0.019	0.01	0.548	0.030	0.06	0.983	0.00	0.02	0.079	0.009	0.24
vsknn-hybrid	0.748	0.018	0.01	**0.552**	0.029	0.06	0.981	0.00	0.01	**0.080**	0.009	0.01
gru4rec (baseline)	0.641	0.017	-	0.455	0.027	-	0.990	0.00	-	0.059	0.006	-
gru4rec-recency	**0.720**	0.022	0.01	**0.489**	0.035	0.01	**0.997**	0.00	0.01	**0.061**	0.006	0.01
narm (baseline)	0.666	0.029	-	0.489	0.034	-	0.977	0.00	-	0.072	0.007	-
narm-recency	**0.730**	0.028	0.01	**0.493**	0.037	0.11	**0.989**	0.00	0.01	0.072	0.007	0.72
sr (baseline)	0.612	0.020	-	0.451	0.025	-	0.959	0.01	-	0.069	0.007	-
sr-recency	**0.703**	0.025	0.01	**0.482**	0.035	0.01	**0.984**	0.00	0.01	**0.070**	0.008	0.19

In contrast, MRR and Popularity Bias remained close to baseline values, with MRR peaking at 0.561 for `stan`, `stan-hybrid`, `vstan`, and `vstan-hybrid`, and Popularity Bias reaching 0.080 for `vsknn` and `vsknn-hybrid`. This indicates that

Table 14. Results for the RetailRocket dataset in the Rest of Session scenario.

	Precision			Recall			NDCG			MAP		
Models	mean	std	p-val	mean	std	p-val	mean	std	p-val	mean	std	p-val
stan (baseline)	0.120	0.004	-	0.543	0.018	-	0.514	0.02	-	0.063	0.002	-
stan-recency	**0.128**	0.004	0.01	**0.571**	0.019	0.01	**0.522**	0.02	0.05	**0.067**	0.002	0.01
stan-session	0.127	0.004	0.01	0.570	0.020	0.01	0.520	0.02	0.04	**0.067**	0.002	0.01
stan-hybrid	0.120	0.004	0.01	0.543	0.018	0.01	0.514	0.02	0.01	0.063	0.002	0.01
vstan (baseline)	0.119	0.004	-	0.540	0.017	-	0.511	0.02	-	0.063	0.002	-
vstan-recency	**0.127**	0.004	0.01	**0.569**	0.019	0.01	**0.521**	0.02	0.01	**0.067**	0.002	0.01
vstan-session	**0.127**	0.004	0.01	**0.569**	0.019	0.01	0.520	0.02	0.01	**0.067**	0.002	0.01
vstan-hybrid	0.119	0.004	0.01	0.540	0.017	0.01	0.511	0.02	0.01	0.063	0.002	0.01
vsknn (baseline)	0.120	0.004	-	0.540	0.016	-	0.515	0.02	-	0.063	0.002	-
vsknn-recency	**0.129**	0.004	0.01	**0.571**	0.018	0.01	0.521	0.02	0.09	**0.068**	0.002	0.01
vsknn-session	0.124	0.004	0.01	0.554	0.017	0.01	**0.529**	0.02	0.01	0.065	0.002	0.01
vsknn-hybrid	0.120	0.004	0.01	0.540	0.016	0.01	0.515	0.02	0.01	0.063	0.002	0.01
gru4rec (baseline)	0.099	0.003	-	0.467	0.015	-	0.431	0.02	-	0.052	0.001	-
gru4rec-recency	**0.116**	0.003	0.01	**0.531**	0.019	0.01	**0.492**	0.02	0.01	**0.061**	0.002	0.01
narm (baseline)	0.104	0.004	-	0.485	0.024	-	0.458	0.03	-	0.055	0.003	-
narm-recency	**0.117**	0.004	0.01	**0.538**	0.023	0.01	**0.498**	0.03	0.01	**0.062**	0.003	0.01
sr (baseline)	0.093	0.003	-	0.446	0.018	-	0.422	0.02	-	0.050	0.002	-
sr-recency	**0.112**	0.004	0.01	**0.518**	0.020	0.01	**0.483**	0.03	0.01	**0.060**	0.002	0.01

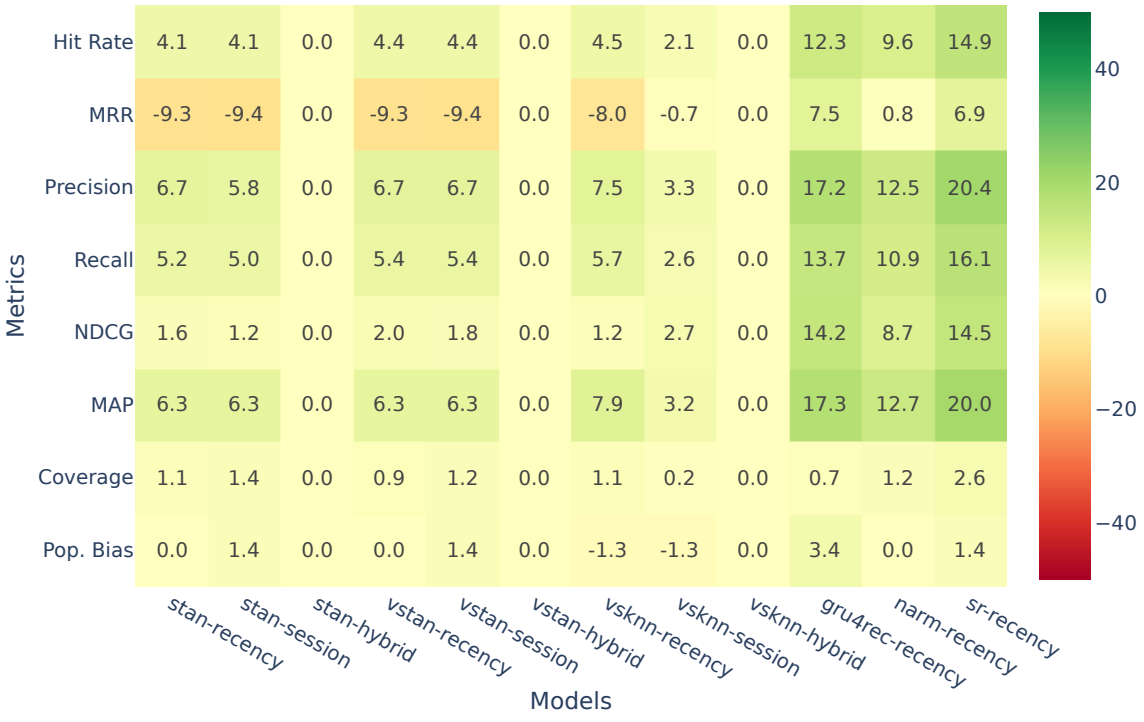

Figure 8. Percent change of each *reminder*-based model relative to its baseline for the RetailRocket dataset.

while *reminder*-based strategies improve the ability to retrieve relevant items, they have a limited effect on refining their exact ranking or altering the distribution toward less popular items.

As shown in Figure 8, recency *reminder*-based strategies for `sr`, `narm`, and `gru4rec` yielded the largest gains over their baselines. Hit Rate increased to 0.703, 0.730, and 0.720, respectively; Precision reached 0.112, 0.117, and 0.116; Recall improved to 0.518, 0.538, and 0.531; NDCG rose to 0.483, 0.498, and 0.492; and MAP reached 0.060, 0.062, and 0.061.

Overall, these results highlight that capturing recent activity is particularly beneficial in e-commerce scenarios, where user intent is often strongly influenced by short-term interactions within a session. In this context, recent clicks and views provide strong signals of immediate interest, making *reminder*-based strategies especially effective. Most of these improvements were statistically significant, confirming the effectiveness of incorporating recent user interactions. This finding is consistent with previous work by [Lerche et al. 2016], who also reported significant gains using *reminders* in e-commerce recommendation experiments.

5. Conclusions and Future Directions

In this work, we investigated the effectiveness of Session-Based Recommender Systems (SBRS) with *reminders*, which incorporate previously viewed items to improve recommendation precision and relevance.

Overall, the experimental results highlight the effectiveness of incorporating *reminder* strategies into SBRS. Performance improvements were remarkably consistent in domain-specific scenarios, such as the Xing and JusBrasilRec datasets, where the gains were not only the highest but also statistically significant, reinforcing the robustness of the observed effects. Similarly, in the RetailRocket dataset, reminder-based models achieved the most significant improvements across all domains, consistently outperforming the baselines on multiple metrics. This finding may be explained by the strong influence of recent user activity in e-commerce sessions.

In contrast, the Music4All, Trivago, and MIND datasets showed comparatively smaller gains, indicating that these models might have less impact in domains with diverse, episodic, or highly session-specific interactions. These observations suggest that the effectiveness of *reminder*-based strategies is highly domain-dependent, particularly influenced by how strongly past user actions correlate with short-term intent.

These findings suggest that incorporating information from previous sessions should be treated as a context-dependent design choice rather than a universally beneficial strategy. Differences in user activities and interaction dynamics across domains may influence how effectively such information can support recommendation services.

The recency, session similarity, and hybrid *reminder* strategies may prove effective in enriching the contextual representation of user sessions. These mechanisms can leverage latent temporal dynamics and inter-session relationships, potentially improving model responsiveness to user behavior patterns. As such, they represent a promising di-

rection for advancing the performance and adaptability of SBRS.

These findings also provide indirect evidence regarding our initial hypothesis on user satisfaction. Although user satisfaction was not measured explicitly, improvements in metrics such as Hit Rate, Recall, and Coverage suggest that reminder-based strategies increase the likelihood of presenting relevant and diverse items to users. At the same time, the limited gains in ranking-based metrics such as MRR and NDCG indicate that while users may be exposed to more useful options, the precise ordering of recommendations may not be consistently improved. Therefore, the results partially support our hypothesis, suggesting that reminder-based strategies can enhance aspects related to perceived usefulness, although their impact on overall user satisfaction may depend on the domain and the importance of ranking quality.

Future work could explore alternative *reminder* strategies and investigate their applicability across a wider variety of domains and session characteristics. Further studies are also needed to assess the integration of *reminders* into different types of recommendation models, as well as to better understand how different *reminder* configurations impact recommendation quality. In particular, exploring different weighting schemes in the hybrid *reminder* strategy, beyond the equal weighting adopted in this work, may provide insights into how the balance between recency and session similarity influences recommendation performance.

6. Acknowledgement

This work was supported by the National Council for Scientific and Technological Development (CNPq), Brazil, under a Master's scholarship Process No. 170391/2023-0, and financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Finance Code 001.

References

[Abel et al. 2016] Abel, F., Benczúr, A., Kohlsdorf, D., Larson, M., and Pálovics, R. (2016). Recsys challenge 2016: Job recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 425–426.

[Domingues et al. 2022] Domingues, M. A., de Moura, E. S., Marinho, L. B., and da Silva, A. (2022). Benchmarking session-based and session-aware recommender systems for jusbrasil. *Anais Estendidos do XXVIII Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 145–148.

[Domingues et al. 2023] Domingues, M. A., de Moura, E. S., Marinho, L. B., and da Silva, A. (2023). A large scale benchmark for session-based recommendations on the legal domain. *Artificial Intelligence and Law*, pages 1–36.

[Garg et al. 2019] Garg, D., Gupta, P., Malhotra, P., Vig, L., and Shroff, G. (2019). Sequence and time aware neighborhood for session-based recommendations: Stan. *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1069–1072.

[Hidasi and Karatzoglou 2018] Hidasi, B. and Karatzoglou, A. (2018). Recurrent neural networks with top-k gains for session-based recommendations. *Proceedings of the*

27th ACM international conference on information and knowledge management, pages 843–852.

[Hidasi et al. 2015] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

[Ibrahim et al. 2025] Ibrahim, O. A. S., Younis, E. M., Mohamed, E. A., and Ismail, W. N. (2025). Revisiting recommender systems: an investigative survey. *Neural Computing and Applications*, 37(4):2145–2173.

[Jannach et al. 2015] Jannach, D., Lerche, L., Kamehkhosh, I., and Jugovac, M. (2015). What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25:427–491.

[Jannach and Ludewig 2017] Jannach, D. and Ludewig, M. (2017). When recurrent neural networks meet the neighborhood for session-based recommendation. *Proceedings of the eleventh ACM conference on recommender systems*, pages 306–310.

[Jannach et al. 2017] Jannach, D., Ludewig, M., and Lerche, L. (2017). Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. *User Modeling and User-Adapted Interaction*, 27:351–392.

[Kamehkhosh et al. 2017] Kamehkhosh, I., Jannach, D., and Ludewig, M. (2017). A comparison of frequent pattern techniques and a deep learning method for session-based recommendation. *RecTemp@ RecSys*, pages 50–56.

[Knees et al. 2019] Knees, P., Deldjoo, Y., Moghaddam, F. B., Adamczak, J., Leyson, G.-P., and Monreal, P. (2019). Recsys challenge 2019: Session-based hotel recommendations. In *Proceedings of the 13th ACM conference on recommender systems*, pages 570–571.

[Lerche et al. 2016] Lerche, L., Jannach, D., and Ludewig, M. (2016). On the value of reminders within e-commerce recommendations. *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 27–35.

[Li et al. 2017] Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., and Ma, J. (2017). Neural attentive session-based recommendation. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428.

[Ludewig and Jannach 2018] Ludewig, M. and Jannach, D. (2018). Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 28:331–390.

[Ludewig et al. 2021] Ludewig, M., Mauro, N., Latifi, S., and Jannach, D. (2021). Empirical analysis of session-based recommendation algorithms: a comparison of neural and non-neural approaches. *User Modeling and User-Adapted Interaction*, 31(1):149–181.

[Masciari et al. 2024] Masciari, E., Umair, A., and Ullah, M. H. (2024). A systematic literature review on ai-based recommendation systems and their ethical considerations. *IEEE Access*, 12:121223–121241.

[Pegoraro Santana et al. 2020] Pegoraro Santana, I. A., Pinhelli, F., Donini, J., Catharin, L., Mangolin, R. B., da Costa, Y. M. e. G., Delisandra Feltrim, V., and Domingues, M. A. (2020). Music4all: A new music database and its applications. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 399–404.

[Saifudin and Widiyaningtyas 2024] Saifudin, I. and Widiyaningtyas, T. (2024). Systematic literature review on recommender system: Approach, problem, evaluation techniques, datasets. *IEEE Access*, 12:19827–19847.

[Tanno et al. 2025] Tanno, D. R., Simm, V. S., Lulu, G. L., and Domingues, M. (2025). Exploring session-based recommender systems with reminders. In *Anais do XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025)*, pages 1–8.

[Verstrepen and Goethals 2014] Verstrepen, K. and Goethals, B. (2014). Unifying nearest neighbors collaborative filtering. *Proceedings of the 8th ACM Conference on Recommender systems*, pages 177–184.

[Wu et al. 2020] Wu, F., Qiao, Y., Chen, J.-H., Wu, C., Qi, T., Lian, J., Liu, D., Xie, X., Gao, J., Wu, W., and Zhou, M. (2020). MIND: A large-scale dataset for news recommendation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

[Xu et al. 2026] Xu, J., Chen, Z., Yang, S., Li, J., Wang, W., Hu, X., Hoi, S., and Ngai, E. (2026). A survey on multimodal recommender systems: Recent advances and future directions. *IEEE Transactions on Multimedia*.

[Zhang et al. 2025] Zhang, X., Xu, B., Li, C., He, B., Lin, H., Ma, C., and Ma, F. (2025). A survey on side information-driven session-based recommendation: From a data-centric perspective. *IEEE Transactions on Knowledge and Data Engineering*.

[Zhang et al. 2026] Zhang, X., Xu, B., Ma, F., Wang, Z., Yang, L., and Lin, H. (2026). Rethinking contrastive learning in session-based recommendation. *Pattern Recognition*, 169:111924.

[Zykov 2022] Zykov, R. (2022). Retailrocket recommender system dataset. `https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset`. Accessed: 2025-05-01.