Multi-funnel Recommender System for Cold Item Boosting

Ahmed Khaili[1], Kostia Kofman[1], Edgar Cano[1], Andrew Mende[1] and Adva Hadrian[1]

[1]*Booking.com*

Abstract

In order to achieve a healthy two sided marketplace, a recommender system balancing the user and the item interests must be deployed. One of the challenges for such a system is to give new items a fair chance to compete against an existing supply. In the context of an online travel agency, this challenge arises not only from the classical cold start problem in recommender systems but also from the customer's reluctance to book new properties due to high uncertainty about their quality, which is often alleviated by past customer reviews. In this paper, we share how a multi-funnel recommender system was developed to address this specific challenge and how the effect of such intervention can be measured on both the customers and the supply. Along with the main recommender delivering a personalized ranking maximizing customers conversion, a second funnel focusing on only cold properties can be used to generate an alternative optimistic ranking. Then these two rankings are merged in a way to balance between short term conversion rate and long term metrics assessing the supply health and properties performance. We show this intervention has different effects when segmenting both customers and properties and suggest future directions given these observations.

Keywords

Recommender system, Cold-start Recommendation, Tourism Recommendation System

1. Introduction

The growth of e-commerce has transformed how customers discover and engage with products and services. With expanding catalogs and diverse customer needs, these platforms face the challenge of helping users find relevant content efficiently. Recommender systems (RS) address this by providing personalized suggestions, enhancing customer experience, and driving engagement [1]. RS rely on past interactions to make accurate predictions. For new customers and items, the system cannot learn good representations due to the lack of past data, leading to sub-optimal recommendations. This is what's known as the cold start problems which affect both users and items. In online travel agencies (OTA), many customers are in a continuous cold start state [2]. This makes the success of any RS depending on addressing this challenge. One popular measure of success for RS is based on transactional data, specifically, conversion rate. However, focusing only on conversion rates can lead to recommending top performers,

Workshop on Recommenders in Tourism (RecTour 2024), October 18th, 2024, co-located with the 18th ACM Conference on Recommender Systems, Bari, Italy.

✉ ahmed.khaili@booking.com (A. Khaili); kostia.kofman@booking.com (K. Kofman); edgar.cano@booking.com (E. Cano); andrew.mende@booking.com (A. Mende); adva.hadrian@booking.com (A. Hadrian)
0009-0007-2035-733X (A. Khaili); 0009-0004-4690-6147 (K. Kofman); 0009-0000-1849-8001 (E. Cano); 0000-0003-2238-9427 (A. Hadrian)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

ignoring new and niche items. This ”closed-loop feedback” reinforces popularity bias, limiting exploration and affecting recommendation diversity and fairness [3, 4]. Addressing the item cold start problem can enhance inventory representation and attract customers seeking long-tail items. [1, 5]

Furthermore, resolving this issue can drive marginal profit by enabling the sale of long-tail items in smaller quantities, as demonstrated by platforms such as Alibaba, Amazon, and Netflix [6].

At Booking.com, ensuring newcomers get initial traction is important. The tourism industry is highly seasonal, and extra supply is crucial during peak seasons when availability is low. In our marketplace, we refer to property owners as partners. Engaging partners, motivates them to optimize listings and offer exclusive discounts. Partners without early user engagement often neglect their listings, making them less appealing. Our data shows that generating initial traction within 30 days is crucial for long-term partners' engagement. Engaged partners create better offerings, follow best practices, stay active in the platform by offering more rooms, and increase the inventory pool. Moreover, besides the potential bias in an RS against new properties, customers themselves might be reluctant to book a new property without past reviews from other customers even if it was shown to them.

To address RS bias against new properties, we augmented our RS with an additional funnel dedicated to cold properties. This multi-funnel design ensures full control over the fraction of new properties promoted to a given customer, balancing short-term bookings with long-term marketplace health. We define cold items as properties in their first 30 days of joining the marketplace. These properties are ranked both among themselves and with other properties. We employ two models: a main model for all properties and a cold model specifically for cold properties. The cold model has a similar architecture to the main model but does not rely on a subset of the features set, specifically those that encode implicit property information learned through embedding layers. Besides that, learning to rank models often use explicit item-level features encoding past customer interactions with the item, such as lagged conversion rate. To handle the absence of this data due to the lack of interaction history with the cold property, estimated values from similar properties are used. Finally, a de-duplication logic combines the full and cold ranks, ensuring a balanced ratio of cold and warm items is used.

To test the hypothesis that increasing the visibility of cold properties in search results improves their short-term performance and long-term retention, thereby reducing churn, we designed a two-sided marketplace experiment. This involves two separate A/B tests, one on customers: in our case travelers and one on the supply side, in our case our partners. To measure any short-term negative impact on bookings, half of the potential travelers might see the boosted cold properties in search results, while the other half will not. In the partner-facing experiment, half of the cold properties will receive a visibility boost on the platform, while the other half will not. This will help measure the long-term benefits of boosting, such as reduced churn and improved performance. The rest of the paper is structured as follows:

- In section 2, we go over related work from the industry and academia.
- In section 3, we give a brief background over the overall ranking funnel before the introduction of the cold item funnel.
- In section 4, we describe the new cold item funnel and how it fits in the overall RS.

- In section 5, we describe the experimentation setup.
- In section 6, we summarize the offline and online results and we suggest future direction to further improve this multi-funnel recommender system.

2. Related Work

Many prior works from both academia and industry have tackled the cold start problem in recommender systems from the perspective of items. Some approaches rely on inverse propensity scores and similar methods to debias both the training and the offline evaluation of new policies [7]. Others focus on imputing missing information for new items, such as in [8, 9], where a method for learning good item representations was proposed. Transfer learning from other data sources, like social networks and session logs, has also been used to improve ranking accuracy for cold items [10]. Graph neural networks (GNNs) have been proposed to develop robust embeddings for cold items, capturing both structural and semantic details essential for accurate recommendations in data-sparse scenarios [11]. Additionally, dedicating a few slots for pure exploration can help alleviate closed-loop effects in recommender systems [4]. The authors of [12] suggest a multi-funnel fresh content recommendation system, dedicating a specific slot in customers' feeds to new contents. Additional approaches tackle the problem by recommending a tail item most likely to be clicked alongside a head item already clicked, while considering contextual information [13].

For our application, propensity score-related methods and dedicating slots to pure exploration can be costly due to the required level of exploration. Estimating cold property representations is also challenging due to the lack of customer interaction and limited information. In the context of a two-sided marketplace, having separate funnels for exploration and exploitation helps balance short-term customer conversion rates with partner performance and supply health metrics. However, the high cost of exploration in an online travel agency (OTA) setting and the restrictions set by partners on the supply of eligible cold properties can result in recommending irrelevant cold properties. Therefore, we implemented a multi-funnel recommender system where cold item exploration depends on its optimistic estimated relevance relative to the full supply's estimated relevance. This approach balances exploration costs with the need to recommend relevant cold properties.

3. Background on the Overall System

Web scale applications such as Booking.com utilize vast amounts of information about the customer's past interactions, the search context and the item explicit/implicit information. Most of those inputs are either categorical or transformed into categorical features, resulting in large sparse inputs.

An important aspect in personalized and contextualized recommender systems is the ability to model feature interactions, a simple example can be the traveler type, e.g solo, couple, family, etc. interacted with the property type, e.g hotel, entire villa, etc. We expect customers will prefer specific property types given their search context. In reality, those preferences depend on additional factors such as the travel destination, price, customer history and many others.

Such systems suffer from additional complexities, such as processing high cardinality features, ranging between hundreds to millions of unique categorical values.

Throughout the years our ranking model evolved from a linear model [14] in which feature interactions were engineered manually and processing high cardinality features was possible by applying the hashing trick, to gradient boosting decision trees algorithms in which there was no need to model the feature interactions explicitly but limitations on the amount of training data and the type of features that can be used became an issue.

The current ranking model is based on the Deep Cross Networks (DCN) architecture [15], the DCN architecture is designed to learn effectively and explicitly feature interactions of a bounded degree (depending on the number of cross layers in the model). Combining the architecture with one of the common ways to handle large and sparse feature space, i.e embedding layers we are able to address both the sparsity and high cardinality of the input features from one hand and learn explicitly the relevant interactions between those features.

Selected feature groups which are important for the understanding of following sections are listed in table 1.

Table 1
Selected Features in Model

Feature Group	Feature Description	Feature Transformation
itemID	Unique id of the item	The unique id is mapped in a lookup table, one-hot encoded and passed through an embedding layer
itemID CVR	Conversion rate from impression to book of item	Value is bucketized, one-hot encoded and passed through an embedding layer
Customer context	Search context of customer, e.g length of stay, search destination, device type, etc.	Values are either one-hot encoded and passed through an embedding layer or first bucketized

4. Personalized Cold Start Properties Boosting

As mentioned above, besides the bias in the model against cold properties, customers are reluctant to book these properties without prior reviews from other customers. For this reason, we wanted a separate tailored pipeline to cold properties in order to be able to have full control on the fraction of cold properties being boosted in every search in order to balance between short term conversion and metrics assessing the long term marketplace's health.

Figure 1 is the diagram of the full multi-funnel ranking pipeline. The highlighted yellow part is the pipeline described above prior to the introduction of the cold start item funnel. The highlighted blue part of the pipeline, the cold start funnel, is the main focus of this paper. In this section, we will be diving in each component of it.

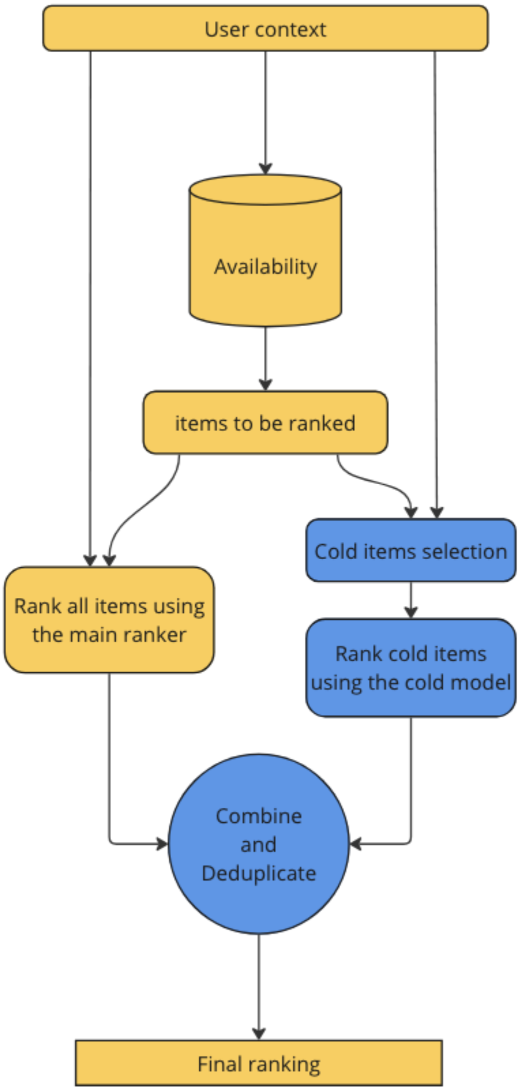

Figure 1: Multi-funnel RS Architecture.

4.1. Cold Item Selection

As a first iteration, and for the reason mentioned above, we defined cold items as any properties that joined the platform less than 30 days ago. Note that, in addition to being ranked along warm items using the main model, they are also ranked exclusively among themselves using the cold model. This design is because according to this definition of cold items, some of them can still organically appear in the top slot of the ranking without any additional intervention. Any additional properties' boosting will be applied to the rest of the cold properties.

4.2. Cold Model

The cold model follows the same architecture of the main model with one exception: The itemID feature was dropped during its training. It was trained on the same dataset. Learning to rank DNN models typically rely on the item id as feature to learn an implicit item representation through an embedding layer. Cold items however lack enough historical data to generate a meaningful embeddings for them. Many approaches are available in the literature to address this problem by estimating cold items embedding using existing items [8, 9, 11]. However, some of these approaches require customers' interactions with these cold items, while others require training multiple models. Given that, in our case, a separate pipeline will be used specifically for cold items only, and because of all the additional complexity required to estimate such an embedding, we decided to simply drop this feature from the cold model. Besides resulting in a less complex training process for the cold model, we observed no effect from dropping this feature when evaluating on cold properties only, and it resulted in a lighter model at serving.

It's a common practice for learning to rank models to use item level features encoding explicit past customer interactions with the item in question, for example lagged conversion rate by customer cohort and historical review scores. However these features are not completely defined for cold items. Dropping them completely from the cold model negatively affects some property cohorts more than others. This is because the performance distribution varies from one cohort to another significantly, and relying on only descriptive property's features will affect unequally the top performers of each cohort. To tackle this, these features were kept in the cold model. At training, their actual values are used, and at serving, they were replaced by an estimate based on similar counterparts from the warm properties pool.

4.3. Deduplication Logic

Once a cold list and a full ranked list are combined, and before showing the final list to a given customer, it's necessary to deduplicate the cold items since they will appear in both the cold and full lists.

This deduplication logic is as following:

Given:

- $\boldsymbol{A}$: eligible properties to be ranked.
- $\boldsymbol{C}$: eligible cold properties to be ranked: a subset of A.
- $\boldsymbol{R}$: full rank : R_i a ranking score for each property $p_i \in A$.
- $\boldsymbol{r}$: cold rank : r_i a ranking score for each property $p_i \in C$.
- $\boldsymbol{N}$: a hyperparameter controlling the maximum cold properties to be boosted

For cold properties, we define :

- Boosted cold properties : $p_i \in C$ where $r_i > R_i$.
- Non boosted cold properties : $p_i \in C$ where $r_i <= R_i$.

The deduplication logic is as follow:

- Take the top $\boldsymbol{N}$ boosted cold properties, and replace R_i in $\boldsymbol{R}$ with r_i from $\boldsymbol{r}$.
- Sort the new list $\boldsymbol{R}$.

Given that we rely on merging two ranking scores from two independent ML models. It was essential to add a calibration layer assuring the same distribution of the ranking score.

5. Experiment Design

Given that this is a two sided market, its necessarily to evaluate the impact for both sides:

1. **Traveler side**: which refer to customers booking their accommodations
2. **Supply Side**: which refers to the cold properties in our platform.

For the supply side it is expected that by boosting the ranking of cold properties, they will get more bookings in the short term and in the long term they will increase their performance and retention.

For the traveler side it's not expected that the treatment will have a positive impact, this is because potential customers might be skeptical to book cold properties that lack the proof of quality (eg. Reviews). In fact boosting the cold supply might have a negative impact, this is why it's necessary to set an acceptable loss, this loss (on the traveler side) should be outweighed by the benefit on the supply side.

To measure the impact on both sides: the traveler and the supply, it is aimed to run a pseudo two-sided marketplace experiment. Two experiments will be run, one after the other, starting with a traveler facing intervention which will quantify the short term loss, followed by a supply side intervention to measure the long benefit. Overall the short term loss should be outweighed by the positive impact on the supply. It was decided to run one experiment after the other to increase power as we are maximizing the sample size that will be exposed to the treatment.

5.1. Traveler Facing Experiment

An A/B experiment was set up where half of the potential travelers would be exposed to the new boosting mechanism while the other half will not, then we will measure the difference between these two groups.
Hypothesis: By introducing the boosting mechanism it is expected to have a negative impact on conversion, this impact is considered short term.
Once the experiment was concluded and the acceptable loss was measured the supply side experiment started.

5.2. Supply Facing Experiment

A follow up A/B experiment was set-up, where half of the cold supply will be eligible to be boosted while the other half will not get the boost. In here it's aimed to measure the long term impact of the boost even after the boosting timeframe has run out.

Hypothesis: By boosting the new supply we aim to increase retention and long term performance, the value of the benefit from an increase in retention measured in this experiment should be greater than the negative effect in conversion measured in the traveler facing intervention. Even when the boost was temporary (first 30 days of being on the platform) the impact in retention and performance was measured long after this, the goal here was to capture the long term impact.

6. Results

6.1. Offline Results

While removing the itemID embedding resulted in around a 1% drop in ndcg@10, when restricting the evaluation set to only cold items, we do not see a significant change in the same metric. This confirms that having the itemID as a feature without any embedding estimations for cold items benefits only warm items.

As mentioned above a property's relevance depends heavily on the context. In addition to using the main model as a baseline for ranking cold supply, a simple baseline consisting of ranking by least expensive is also used offline to evaluate the cold model. Below in table 2 is how theses two baselines compare to the cold model.

Table 2
Cold Model Against Baselines

Evaluation data	NDCG@10 Cold Model Relative to Main model	NDCG@10 Cold Model Relative to least expensive
Cold reservations - Cold supply	0.8% ± 1.04%	25% ± 1.6%
All reservations - full supply	-0.9% ± 0.03%	-

6.2. Online results

From previous online experiments, we measured the drop in conversion rate relative to the number of boosted cold properties (figure 2), based on this data, we selected an appropriate value of N for the algorithm in 4.3.

As it was expected, this intervention resulted in a drop in conversion rate measured through a customer facing A/B experiment. Results are in table 3.

After observing a notable disparity between the results of different customer cohorts, a detailed offline analysis of the customer facing experiment data revealed that there might be a room for improvement by using uplift modeling to predict searches that are unlikely to convert when cold properties are boosted. And by hiding this boost from these searches, it is possible to achieve a smaller decline in conversion rate without significantly decreasing the reservations

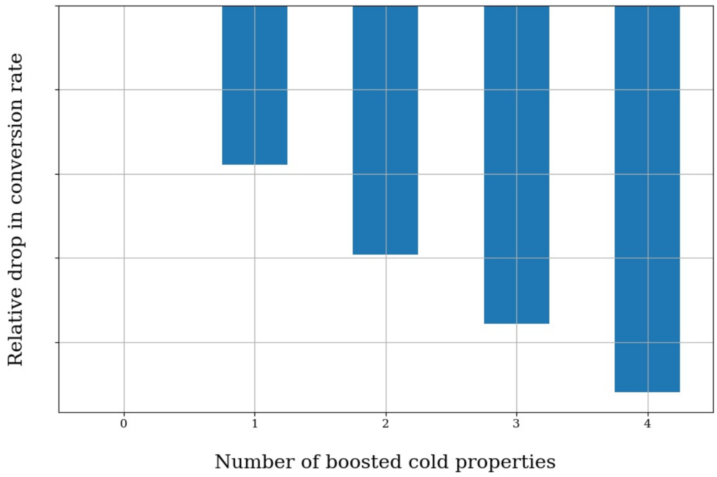

Figure 2: Drop in Conversion Rate Relative to Number of Cold Properties Boosted

Table 3
Relative Drop in Conversion Rate from the Customer Facing experiment

Experiment	Relative Impact on Conversion Rate
Customer facing	-0.09% ±0.07%

for cold properties. In Figure 3 we show the estimated benefit of this personalized boosting through uplift modeling.

In the partner facing experiment, we examined the effect of boosting on a partner's retention rate. Looking at the supply as a whole the results were inconclusive, however, by looking at partner's cohorts separately(Figure 4) we see positive signals in one important group, to validate these signals the experiment would need to be re-run with that cohort/group as the main group of interest.

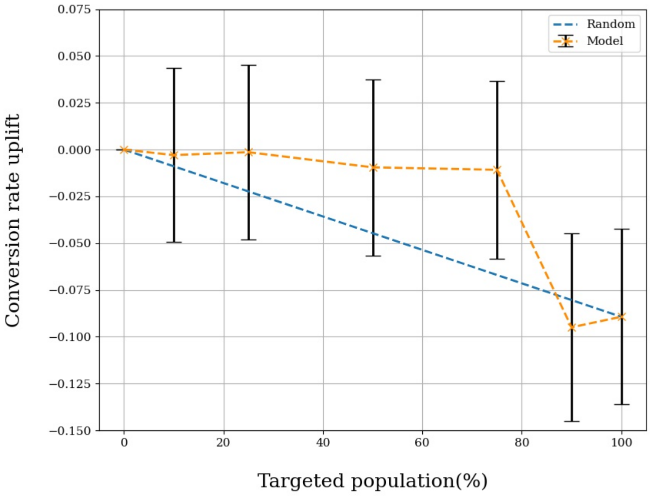

Figure 3: Conversion Rate Uplift Relative to the Targeted Population

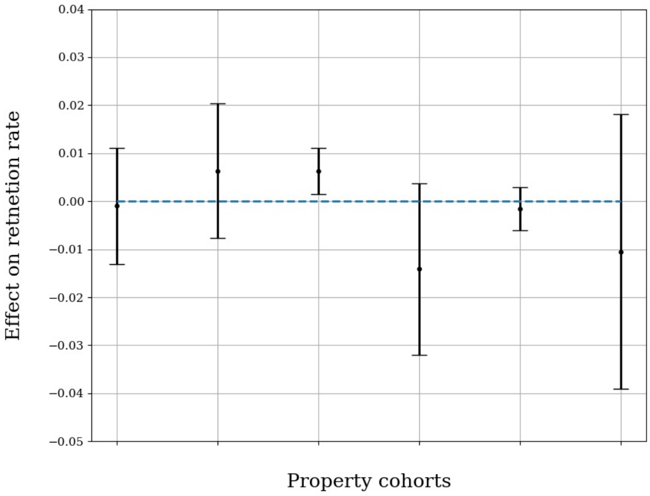

Figure 4: Effect on property's Churn Rate by property's Cohort

7. Conclusion and Next Steps

Promoting new items is crucial for the long-term health and sustainability of recommender systems, particularly within online travel agencies. Ensuring that new properties receive adequate exposure is essential for keeping the marketplace dynamic and attractive to both travelers and property owners. However, addressing the property's cold start problem presents significant challenges, primarily due to the limited information available on them and the absence of past customer reviews, which are key trust signals for prospective customers. This paper proposed a multi-funnel recommender system targeting cold properties by implementing a dedicated ranking funnel for them, which is integrated with the main ranking funnel. Cold properties are ranked independently and then combined with the main ranking via a deduplication logic in order to balance the need for short-term customer conversion with long-term supply health. The experimental setup demonstrated the system's ability to improve new properties engagement, with measured trade-offs between short-term customer conversion rate and long-term partner retention.

As next steps, future improvements could focus on personalizing the boosting mechanism to target only less sensitive customers to new properties. Restricting the boosting to only property cohorts that are more likely to benefit from an increased visibility is also an option. By refining this approach, it will be possible to further reduce the short-term impact on conversion rates while improving new supply's long-term health metrics.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems 46 (2013) 109–132. doi:10.1016/J.KNOSYS.2013.03.012.

[2] L. Bernardi, J. Kamps, J. Kiseleva, M. J. Mueller, The continuous cold start problem in e-commerce recommender systems, CEUR Workshop Proceedings 1448 (2015) 30–33. URL: https://arxiv.org/abs/1508.01177v1.

[3] H. Abdollahpouri, M. Mansoury, R. Burke, B. Mobasher, The unfairness of popularity bias in recommendation, CEUR Workshop Proceedings 2440 (2019). URL: https://arxiv.org/abs/1907.13286v3.

[4] A. H. Jadidinejad, C. MacDonald, I. Ounis, Using exploration to alleviate closed loop effects in recommender systems, SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020) 2025–2028. URL: https://dl.acm.org/doi/10.1145/3397271.3401230. doi:10.1145/3397271.3401230/SUPPL_FILE/3397271.3401230.MP4.

[5] M. Volkovs, G. Yu, T. Poutanen, Dropoutnet: Addressing cold start in recommender systems, Advances in neural information processing systems 30 (2017).

[6] C. Anderson, Why the future of business is selling less of more, 2006.

[7] L. Yang, Y. Cui, Y. Xuan, C. Wang, S. B. P. of the 12th ..., undefined 2018, Unbiased offline recommender evaluation for missing-not-at-random implicit feedback, dl.acm.orgL Yang, Y Cui, Y Xuan, C Wang, S Belongie, D EstrinProceedings of the 12th ACM conference on recommender systems, 2018•dl.acm.org (2018) 279–287. URL: https://dl.acm.org/doi/abs/10.1145/3240323.3240355. doi:10.1145/3240323.3240355.

[8] F. Pan, S. Li, X. Ao, P. Tang, Q. He, Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings, SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019) 695–704. URL: https://dl.acm.org/doi/10.1145/3331184.3331268. doi:10.1145/3331184.3331268/SUPPL_FILE/CITE2-12H00-D3.MP4.

[9] Y. Zhu, R. Xie, F. Zhuang, K. Ge, Y. Sun, X. Zhang, L. Lin, J. Cao, Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks,

SIGIR 2021 - Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (2021) 1167–1176. URL: https://dl.acm.org/doi/10.1145/3404835.3462843. doi:10.1145/3404835.3462843/SUPPL_FILE/192.MP4.

[10] J. Gope, S. K. Jain, A survey on solving cold start problem in recommender systems, in: 2017 International Conference on Computing, Communication and Automation (ICCCA), IEEE, 2017, pp. 133–138.

[11] S. Liu, I. Ounis, C. MacDonald, Z. Meng, A heterogeneous graph neural model for cold-start recommendation, SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020) 2029–2032. URL: https://dl.acm.org/doi/10.1145/3397271.3401252. doi:10.1145/3397271.3401252/SUPPL_FILE/3397271.3401252.MP4.

[12] J. Wang, H. Lu, S. Z. Google, B. L. Google, H. W. Google, D. G. Google, B. L. Google, S. B. Google, E. H. C. Google, C. J. G. Google, S.-L. W. Google, L. B. Google, M. C. Google, S. Zhang, B. Locanthi, H. Wang, D. Greaves, B. Lipshitz, S. Badam, E. H. Chi, C. J. Goodrow, S.-L. Wu, L. Baugher, M. Chen, Fresh content needs more attention: Multi-funnel fresh content recommendation, dl.acm.orgJ Wang, H Lu, S Zhang, B Locanthi, H Wang, D Greaves, B Lipshitz, S Badam, EH ChiProceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and ..., 2023•dl.acm.org (2023) 5082–5091. URL: https://dl.acm.org/doi/abs/10.1145/3580305.3599826. doi:10.1145/3580305.3599826.

[13] T. Didi, I. Guy, A. Livne, A. Dagan, L. Rokach, B. Shapira, Promoting tail item recommendations in e-commerce, UMAP 2023 - Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization 23 (2023) 194–203. URL: https://dl.acm.org/doi/10.1145/3565472.3592968. doi:10.1145/3565472.3592968.

[14] T. Mavridis, S. Hausl, A. Mende, R. Pagano, Beyond algorithms: Ranking at scale at booking. com., in: ComplexRec-ImpactRS@ RecSys, Virtual, 2020.

[15] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, E. Chi, Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems, in: Proceedings of the web conference 2021, 2021, pp. 1785–1797. doi:10.1145/3442381.3450078.