EvalRS: a rounded evaluation of recommender systems

Jacopo Tagliabue[1,2,*,†], Federico Bianchi[3,†], Tobias Schnabel[4,†], Giuseppe Attanasio[5,†], Ciro Greco[1,2,†], Gabriel de Souza P. Moreira[6,†] and Patrick John Chia[7,†]

[1]South Park Commons, New York, NY, USA
[2]Coveo Labs, New York, NY, USA
[3]Stanford University, Stanford, CA, USA
[4]Microsoft, Redmond, WA, USA
[5]Bocconi University, Milan, Italy
[6]NVIDIA, São Paulo, Brazil
[7]Coveo, Montreal, Canada

Abstract

Much of the complexity of recommender systems (RSs) comes from the fact that they are used as part of highly diverse real-world applications which requires them to deal with a wide array of user needs. However, research has focused almost exclusively on the ability of RSs to produce accurate item rankings while giving little attention to the evaluation of RS behavior in real-world scenarios. Such narrow focus has limited the capacity of RSs to have a lasting impact in the real world and makes them vulnerable to undesired behavior, such as the reinforcement of data biases. We propose `EvalRS` as a new type of challenge, in order to foster this discussion among practitioners and build in the open new methodologies for testing RSs "in the wild".

Keywords

recommender systems, behavioral testing, open source

1. Introduction

Recommender systems (RSs) are embedded in most applications we use today. From streaming services to online retailers, the accuracy of a RS is a key factor in the success of many products. Evaluation of RSs has often been done considering point-wise metrics, such as *HitRate* (HR) or *nDCG* over held-out data points, but the field has recently begun to recognize the importance of a more rounded evaluation as a better proxy to real-world performance [1].

We designed `EvalRS` as a new type of data challenge in which participants are asked to test their models incorporating quantitative as well as behavioral insights. Using a popular open dataset – *Last.fm* – we go beyond single aggregate numbers and instead require participants to optimize for a wide range of recommender systems properties. The contribution of this challenge is two-fold:

1. We propose and standardize the data, evaluation loop and testing for RSs over a popular use case (user-item recommendations for music consumption [2]), thus releasing in the open domain a first unified benchmark for this topic.
2. We bring together the community on evaluation from both an industrial and research point of view, to foster an inclusive debate for a more nuanced evaluation of RSs.

In this paper, we describe the conceptual and practical motivations behind `EvalRS`, provide context on the organizers, related events and relevant literature, and explain the evaluation methodology we champion. For participation rules, up-to-date implementation details and all the artifacts produced before and during the challenge, please refer to the `EvalRS` official repository.[1]

2. Motivation

`EvalRS` at *CIKM 2022* complements the existing challenge landscape and it is driven by two different perspectives: the first one coming from academic research, the second one from the industrial development of RSs. We examined these in turn.

EvalRS 2022: CIKM EvalRS 2022 DataChallenge, October 21, 2022, Atlanta, GA

*Corresponding author.

†TS proposed the format and methodology and worked with JT and FB towards a first draft. PC led the implementation and contributed most of the `RecList` code. GA, CG, FB and PC researched, iterated and operationalized behavioral tests. GM reviewed the API and implemented baselines, while GA, JT and FB prepared tutorials for participants. *Everybody* helped with drafting the paper, rules and guidelines. JT and FB acted as senior PIs in the project. JT and CG started this work at Coveo Labs, New York, NY, USA.

tagliabue.jacopo@gmail.com (J. Tagliabue)

[1]https://github.com/RecList/evalRS-CIKM-2022.

2.1. A Research Perspective

Although undeniable progress was made in the past years, concerns have been raised about the status of research advancements in the field of recommendations, particularly with respect to ephemeral processes in motivating architectural choices and lack of reproducibility [3]. This challenge draws attention to a further – and potentially deeper – issue: even if the "reproducibility crisis" is solved, we are still mostly dealing with point-wise quantitative metrics as the only benchmarks for RSs. As reported by Sun et al. [4], the dominating metrics used in the evaluation of recommender systems published at top-tier conferences (RecSys, SIGIR, CIKM) are standard information retrieval metrics, such as *MRR*, *Recall*, *HITS*, *NDCG* [5, 6, 7, 8, 9].

While it is undoubtedly convenient to summarize the performance of different models via *one* score, this lossy projection discards a lot of important information on model behavior: for example, given the power-law distribution in many real-world datasets ([10, 11, 12]), marginal improvements on frequent items may translate in noticeable accuracy gains, even at the cost of significantly degrading the experience of subgroups. Metrics such as coverage, serendipity, and bias [13, 14, 15] are a first step in the right direction, but they still fall short of capturing the full complexity of deploying RSs.

Following the pioneering work of [16] in Natural Language Processing, we propose to supplement standard retrieval metrics with new tests: in particular, we encourage practitioners to go beyond the false dichotomy "quantitative-and-automated" vs "qualitative-and-manual", and find a middle ground in which behavioral *desiderata* can be expressed transparently in code [1].

2.2. An Industrial Perspective

RSs in practice differ from RSs used in research in crucial ways. For example, in research, a static dataset is used repeatedly, and there is no real interactivity between the model and users: prediction over a given point in time x_t in the test set doesn't change what happens at x_{t+1}[2]. Even without considering the complexity of reproducing real-world interactions for benchmarking purposes, we highlight four important themes from our experience in building RSs at scale in production scenarios:

- *Cold-start performance*: new/rare items and users are challenging for many models across industries [19, 20]. In e-commerce, for instance, while most "similar products" predictions will happen over frequent items, in reality, new users and items can represent a big portion of them with significant business consequences: the cold-start problem is believed to affect 50% of users [21] in a context where field studies found that 40% of shoppers would stop shopping if shown non-relevant recommendations [22].
- *Use cases and industry idiosyncrasies*: different use cases in different industries present different challenges. For instance, recommendations for complementary items in e-commerce need to account for the fact that if item A is a good complementary candidate for item B, the reverse might not hold (e.g. an *HDMI cable* is a good complementary item for a *4k TV*, but not vice versa). Music recommendations need to deal with the issue of "hubness", where popular items act as hubs in the top-N recommendation list of many users without being similar to the users' profiles and making other items invisible to the recommender [23]. Such use-case specific traits are particularly important when designing effective testing procedures and often require considerable domain knowledge.
- *Not all mistakes are equal*: point-wise metrics are unable to distinguish different *types* of mistakes; this is especially problematic for recommender systems, as even a single mistake may cause great social and reputational damage [24].
- *Robustness matters as much as accuracy*: while historically a significant part of industry effort can be traced back to a few key players, there is a blooming market of Recommendation-as-a-Service systems designed to address the needs of "reasonable scale" systems [25]. Instead of vertical scaling and extreme optimization, SaaS providers emphasize horizontal scaling through multiple deployments, highlighting the importance of models that prove to be flexible and robust across many dimensions (e.g., traffic, industry, etc.).

While not related to model evaluation *per se*, decision-making processes in the real world would also take into account the different resources used by competing approaches: time (both as time for training and latency for serving), computing (CPU vs GPU), CO2 emissions are all typically included in an industry benchmark.

3. EvalRS Challenge

We propose to supplement standard retrieval metrics over held out data points with *behavioral tests*: in behavioral tests, we treat the target model as a black-box and supply only input-output pairs (for example, query

[2]This is especially important in the context of sequential recommender[17], which arguably resembles more reinforcement learning than supervised inference with pseudo-feedback [18].

user and desired recommended song). In particular, we leverage a recent open-source package, `RecList` [1], to prepare a suite of tests for our target dataset (Section 3.1). In putting forward our tests, we operationalize the intuitions from Section 2 through a general plug-and-play API to facilitate model comparison and data preparation, and by providing convenient abstractions and ready-made recommenders used as baselines.

3.1. Use Case and Dataset

`EvalRS` is a user-item recommendation challenge in the music domain: participants are asked to train a model that, given a user id, recommends an appropriate song out of a known set of songs. The ground truth necessary to compute all the test metrics, quantitative *and* behavioral, is provided by our leave-one-out framework: for each user, we remove a song from their listening history and use it as the ground truth when evaluating the models.

We provide test abstractions and an evaluation script designed for *LFM*, a transformed version of *LFM-1b dataset* [2] – a dataset focused on music consumption on *Last.fm*. We chose the *LFM-1b dataset* as the primary data source after a thorough comparisons of popular datasets for a unique combination of features. Given our focus on rounded evaluation and the importance of joining prediction / ground truth with meta-data, *LFM* is an ideal dataset, as it provides rich song (artist, album information) and user (country, age, gender,[3] time on platform) meta-data.

We applied principled data transformations to make `EvalRS` amenable to a larger audience whilst preserving the rich information in the original dataset. We detail the data transformation process and our motivations:

- First, we removed `users` and `artists` which have few interaction since they are likely to be too sparse to be informative. Following the suggestions in, we apply k-core [26] filtering to the bipartite interaction graph between `users` and `artists`, setting $k = 10$ (i.e. we retain vertices with a minimum degree of k).
- After the aforementioned processing, the dataset still contained over 900M events, which motivated further filtering of the data. In particular, we keep only the *first* interaction a `user` had with a given `track`, and for each `user` we retain only their $N = 500$ most recent unique `track` interactions. We supplement the information lost during this pruning step by providing the interaction count between a `user` and a `track`.

[3]Gender in the original dataset is a binary variable. This is a limitation, as it gives a stereotyped representation of gender. Our intent is not to make normative claims about gender.

Table 1
Descriptive statistics for *LFM* dataset.

Items	Value
Users	119,555
Artists	62,943
Albums	1,374,121
Tracks	820,998
Listening Events	37,926,429
User-Track History Length (25/50/75 pct)	241/346/413

- We then performed another iteration of k-core filtering, this time on the `user-track` interaction graph, with $k = 10$ to retain only users and tracks which are informative.
- Lastly, the original dataset contained missing meta-data (e.g. there were `track_id` in the events data which did not have corresponding track metadata). We removed tracks, albums, artists and events which had missing information.
- We summarize the final dataset statistics in Table 1.

Taken together, these features allow us to fulfill `EvalRS` promise of offering a challenging setting and a rounded evaluation. While a clear motivation behind the release of *LFM-1b dataset* was to offer "additional user descriptors that reflect their music taste and consumption behavior", it is telling that both the modelling and the evaluation by the original authors are still performed without any real use of these rich meta-data [27]. By taking a fresh look at an existing, popular dataset, `EvalRS` challenges practitioners to think about models not just along familiar quantitative dimensions, but also along non-standard scores closer to human perception of relevance and fairness.

3.2. Evaluation Metrics

Submission are evaluated according to our randomized loop (Section 3.3) over the testing suite released with the challenge. At a first glance, tests can be roughly divided in three main groups:

- **Standard RSs metrics**: these are the typical point-wise metrics used in the field (e.g. MRR, HR@K) – they are included as sanity checks and as a informative baseline against which insights gained through the behavioral tests can be interpreted.
- **Standard metrics on a per-group or slice basis**: as shown for example in [1], models which are indistinguishable on the full test set may exhibit very different behavior across data slices.

It is therefore crucial to quantify model performance for specific input and target groups, i.e. is there a performance difference between males and females? Is there an accuracy drops when artists are not very popular?

- **Behavioral tests**: this group may include perturbance tests (i.e. if we modify a user's history by swapping *Metallica* with *Pantera*, how much will predictions change?), and error distance tests (i.e. if the ground truth is *Shine On You Crazy Diamond* and the prediction is *Smoke on the Water*, how severe is this error?).

Based on this taxonomy, we now survey the tests implemented in the `RecList` powering `EvalRS`, with reference to relevant literature and examples from the target datasets. For implementation details please refer to the official repository.[4]

3.2.1. Standard RSs metrics

Based on popular metrics in the literature, we picked two standard metrics as a quantitative baseline and sanity check for our `RecList`:

- **Mean Reciprocal Rank** (MRR) as a measure of where the first relevant element retrieved by the model is ranked in the output list. Besides being considered a standard rank-aware evaluation metric, we chose MRR because it is particularly simple to compute and to interpret.
- **Hit Rate** (HR), defined as *Recall at k* ($k = 100$), i.e. the proportion of relevant items found in the top-k recommendation.

3.2.2. Standard metrics on a per-group or slice basis

Models are tested to address a wide spectrum of known issues for recommender systems, for instance: fairness (e.g. a model should have equal outcomes for different groups, e.g. [28, 29, 30]), robustness (e.g. a model should produce good outcomes also for long-tail items, such as items with less history or belonging to less represented categories, e.g. [31]), industry-specific use-cases (e.g. in the case of music, your model should not consistently penalize niche or simply less known artists).

All the tests in this group are based on *Miss Rate* (MR), defined as ratio between the prediction errors (i.e. model predictions do not contain the ground truth) and the number of predictions. Slices can be generalized as n partitions (e.g. Countries with UK/US/IT/FR and others is split is N partitions) of the test data forming n-ary classes. The absolute difference between the MR obtained on each slice and the the MR obtained on the original test set is averaged and negated (so that a higher value implies better performance in the metric) to obtain the final score for each test. The slice-based tests considered for the final scores are:

- **Gender balance**. This test is meant to address fairness towards gender [32]. Since the dataset only provides binary gender, the test will minimize the difference between the MR obtained on users who specified Female as gender and the MR obtained on the original test set. In other words, the smaller the difference, the fairer the model towards potential gender biases.
- **Artist popularity**. This test is meant to address a known problem in music recommendations: niche (or simply less known) artists and users who are less interested in highly popular content are often penalized by recommender systems [33, 34]. This point appears even more important when we consider that several music streaming services (e.g. Spotify, Tidal) also act as marketplaces for artists to promote their music. Since splitting the test set in two would draw an arbitrary line between popular vs. unpopular artists, failing to capture the actual properties of the distribution. Instead, we split the test set into bins with equal size after logarithmic scaling.
- **User country**. Music consumption is subject to many country dependent factors, such as language differences, local sub-genres and styles, local licensing and distribution laws, cultural influences of local traditional music, etc [35]. We capture this diversity by slicing the test set based on the top-10 countries by user counts.
- **Song popularity**. This test measures the model performance on both popular tracks and on songs with fewer listening events. The test is designed to address both robustness to long tail items and cold-start scenarios, so we pooled together both less popular and newer songs. Again, we used logarithmic bucketing with base 10 to divide the test set in order to avoid arbitrary thresholds.
- **User history**. The test can be viewed as a robustness/cold-start test, in which we sliced the dataset based on the length of user history on the platform. To create slices, we use the user play counts (i.e. the sum of play counts per user) and we use logarithmic bucketing in base 10 to divide the test set in order to avoid arbitrary thresholds.

3.2.3. Behavioral and qualitative tests

Our final set of tests is *behavioral* in nature, and tries to capture (with some assumptions) how models differ based on qualitative aspects:

[4] https://github.com/RecList/evalRS-CIKM-2022.

- **Be less wrong**. It is important that RSs maintain a reasonable standard of relevance even when the predictions are not accurate. For instance, if the ground truth for a recommendation is the rap song *'Humble'* by Kendrick Lamar, a model might suggest another rap song from the same year (*'The story of O.J.'* by Jay-Z), or a famous pop song from the top chart of that year (*'Shape of You'* by Ed Sheeran). There is still a substantial difference between these two as the first one is closer to the ground truth than the second. Since this has a great impact on the overall user experience, it is desirable that models test and measure their performance scenarios like the one just described. We use the latent space of tracks to compute the average pairwise cosine distance between the embeddings of the predicted items and the ground truths.
- **Latent diversity**: Diversity is closely tied with the maximization of marginal relevance as a way to acknowledge uncertainty of user intent and to address user utility in terms of discovery [36]. Diversity is often considered a partial proxy for fairness and it is an important measure of the performance of recommender systems in real world scenarios [37]. We address diversity using the latent space of tracks testing for model density - where density is defined as the summation of the differences between each point in the prediction space and the mean of the prediction space. Additionally, in order to account also for the "correctness" of prediction vectors, we calculate a bias defined as the distance between the ground truth vector and the mean of the prediction vector and weight to penalize for high bias: the final score is computed as 0.3 * diversity - 0.7 * bias, where 0.3 and 0.7 are weights that we determined empirically to balance diversity and correctness.

Please note that since we aim at widening the community contribution to testing, the final code submission for `EvalRS` includes *as a requirement* that participants contribute at least one custom test, by extending the provided abstraction.

3.2.4. Final score

Since each of the tests above return a score from a potentially unique, non-normal distribution, we need a way to define a *macro-score* for the leaderboard. To define the formula we adopt an empirical approach in two phases:

1. *First phase*: scores of individual tests are simply averaged to get the leaderboard macro-score. The purpose of this phase is to gather data on the relative difficulty and utility of the different tests, and get participants comfortable, through harmless iterations, with the dataset and the multi-faceted nature of the challenge.
2. *Second phase*: after the organizers have evaluated the score distributions for individual tests, they will attach different weights to each test to produce a balanced *macro-score* - i.e. if a test turns out to be easy for most participants, its importance will be counter-biased in the calculation. At the beginning of this phase, participants are asked to update their evaluation script by cloning again the data challenge repository: the purpose for each team becomes now leveraging the insights from the previous phase to optimize their models as much as possible for the leaderboard. *Only scores obtained in this phase are considered for the final prizes.*

3.3. Methodology

Since the focus of the challenge is a popular public dataset, we implemented a robust evaluation procedure to avoid data leakage and ensure fairness[5]. Our protocol is split in two phases: *local* – when teams iterate on their solution *during the challenge* - and *remote* – when organizers verify the submissions at the end and proclaim the winners:

- *Local evaluation protocol*: For each fold, the provided script first samples 25% of the users in the dataset. It then partitions the dataset into training and testing sets using the leave-one-out protocol: the testing set comprises a list of unique users, where the target song for each of them has been picked randomly from their history. The training set is the listening history for these sampled users with their test song removed. Participants' models will be trained and tuned based on their custom logic on the training set, and then evaluated over the test suite (Section 3.2) to provide a final score for each run (Section 3.2.4); partitioning, training, testing, scoring will be done for a total of 4 repetitions: the average of the runs will constitute the leaderboard score.
- *Remote evaluation protocol*: the organizers will run the code submitted by participants, and repeat the random evaluation loop. The scores thus obtained on the `EvalRS` test suite will be compared with participants submissions as a sanity check (statistical comparison of means and 95% bootstrapped CI).

Thanks to the provided APIs, participants will be able to run the full evaluation loop locally, as well as update

[5]To help participants with the implementation, we provide a template script that can be modified with custom model code.

their leaderboard score automatically through the provided script. To ensure a fair and reproducible remote evaluation, final submission should contain a docker image that runs the local evaluation script and produces the desired output *within the maximum allotted time on the target cloud machine.* Please check `EvalRS` repository for the exact final requirements and up-to-date instructions.

4. Organization, Community, Impact

4.1. Structure and timeline

`EvalRS` unfolds in three main phases:

1. **CHALLENGE**: An open challenge phase, where participating teams register for the challenge and work on improving the scores on both standard and behavioral metrics across the two phases explained above (3.2.4).
2. **CFP**: A call for papers, where teams submit a written contribution, describing their system, custom testing, data insights.
3. **CONFERENCE**: At the conference, winners will be announced and special prizes for novel testings and oustanding student work will be awarded. During the workshop, we plan to discuss solicited papers and host a round-table with experts on RSs evaluation.

Our *CFP* takes a "design paper" perspective, where teams are invited to discuss both how they adapted their initial model to take into account the test suite, and how the tests strengthened their understanding of the target dataset and use case[6].

We emphasize the *CFP* and *CONFERENCE* steps as moments to share with the community *additional* tests, error analysis and data insights inspired by `EvalRS`. By leveraging *RecList*, we not only enable teams to quickly iterate starting from our ideas, but we promise to immediately circulate in the community their testing contribution through a popular open source package. Finally, we plan on using CEUR-WS to publish the accepted papers, as well as drafting a final public report as an additional, actionable artifacts from the challenge.

4.2. Organizers

Jacopo Tagliabue Jacopo Tagliabue was co-founder of Tooso, an Information Retrieval company acquired by Coveo in 2019. As Director of AI at Coveo, he divides his time between product, research, and evangelization: he is Adj. Professor of MLSys at NYU, publishes regularly in top-tier conferences (including NAACL, ACL, RecSys, SIGIR), and is co-organizer of SIGIR eCom. Jacopo was the lead organizer of the SIGIR Data Challenge 2021, spearheading the release of the largest session-based dataset for eCommerce research.

Federico Bianchi Federico Bianchi is a postdoctoral researcher at Stanford University. He obtained his Ph.D. in Computer Science at the University of Milano-Bicocca in 2020. His research, ranging from Natural Language Processing methods for textual analytics to recommender systems for the e-commerce has been accepted to major NLP and AI conferences (EACL, NAACL, EMNLP, ACL, AAAI, RecSys) and journals (Cognitive Science, Applied Intelligence, Semantic Web Journals). He co-organized the SIGIR Data Challenge 2021. He frequently releases his research as open-source tools that have collected almost a thousand GitHub stars and been downloaded over 100 thousand times.

Tobias Schnabel Tobias Schnabel is a senior researcher in the Productivity+Intelligence group at Microsoft Research. He is interested in improving human-facing machine learning systems in an integrated way, considering not only algorithmic but also human factors. To this end, his research draws from causal inference, reinforcement learning, machine learning, HCI, and decision-making under uncertainty. He was a co-organizer for a WSDM workshop this year and has served as (senior) PC member for a wide array of AI and data science conference (ICML, NeurIPS, WSDM, KDD). Before joining Microsoft, he obtained Ph.D. from the Computer Science Department at Cornell University under Thorsten Joachims.

Giuseppe Attanasio Giuseppe Attanasio is a postdoctoral researcher at Bocconi, where he works on large-scale neural architectures for Natural Language Processing. His research focuses on understanding and regularizing models for debiasing and fairness purposes. His research on the topic has been accepted to major NLP conferences (ACL). While working at Bocconi, he is concluding his Ph.D. at the Department of Control and Computer Engineering at Politecnico di Torino.

Ciro Greco Ciro Greco was the co-founder and CEO of Tooso, a San Francisco based startup specialized in Information Retrieval. Tooso was acquired in 2019 by Coveo, where he now works as VP or Artificial Intelligence.He holds a Ph.D. in Linguistics and Cognitive Neuroscience at Milano-Bicocca. He worked as visiting scholar at MIT and as a post-doctoral fellow at Ghent University. He published extensively in top-tier conferences (including

[6] As customary in these events, we will involve a small committee from top-tier practitioners and scholars to ensure the quality of the final submissions.

NAACL, ACL, RecSys, SIGIR) and scientific journals (The Linguistic Review, Cognitive Science, Nature Communications). He was also co-organizer of the SIGIR Data Challenge 2021.

Gabriel de Souza P. Moreira Gabriel Moreira is a Sr. Applied Research Scientist at NVIDIA, leading the research efforts of Merlin research team. He had his PhD degree from ITA university, Brazil, with a focus on Deep Learning for RecSys and Session-based recommendation. Before joining NVIDIA, he was lead Data Scientist at CI&T for 5 years, after working as software engineer for more than a decade. In 2019, he was recognized as a Google Developer Expert (GDE) for Machine Learning. He was part of the NVIDIA teams that won recent RecSys competitions: ACM RecSys Challenge 2020, WSDM WebTour Workshop Challenge 2021 by Booking.com and the SIGIR eCommerce Workshop Data Challenge 2021 by Coveo.

Patrick John Chia Patrick John Chia is an Applied Scientist at Coveo. Prior to this, he completed his Master's degree at Imperial College London and spent a year at Massachusetts Institute of Technology (MIT). He was co-organizer of the 2021 SIGIR Data Challenge and has been a speaker on topics at the intersection of Machine Learning and eCommerce (SIGIR eCom, ECNLP at ACL). His latest interests lie in developing AI that has the ability to learn like infants and applying it to creating solutions at Coveo.

5. Similar Events and Broader Outlook

The CIKM-related community has shown great interest in themes at the intersection of aligning machine learning with human judgment, rigorous evaluation settings, and fairness, as witnessed by popular Data Challenges and important workshops in top-tier venues. Among recent challenges, the 2021 SIGIR-Ecom Data Challenge, the 2021 Booking Data Challenge, and the 2020 RecSys Challenge are all events centered around the evaluation of RSs, yet still substantially different: for example, the SIGIR Challenge focused on MRR as a success metric [10], while the Booking Challenge [38] used top-k accuracy.

Moreover, the growing interest for rounded evaluation led to the creation of many interesting workshops in recent years, such as *IntRS: Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*, *ImpactRS: Workshop on the Impact of Recommender Systems* and *FAccTRec: Workshop on Responsible Recommendation*. For this reason, we expect *this* challenge to attract a diverse set of practitioners: first, researchers interested in the evaluation of RSs and fairness; second, researchers who proposed a new model and desire to test its generalization abilities on new metrics; third, industrial practitioners that started using `RecList` after its release in recent months, and already signaled strong support for behavioral testing in their real-world use cases.

`EvalRS` makes a novel and significant contribution to the community: first, we ask practitioners to "live and breath" the problem of evaluation, operationalizing principles and insights through sharable code; second, we embrace a "build in the open" approach, as all artifacts from the event will be available to the community as a permanent contribution, in the form of open source code, design papers, and public documentation – through prizes assigned based on scores, but also outstanding testing and paper contributions, and special awards for students, we hope to actively encourage more practitioners to join the evaluation debate and get a more diverse set of perspectives for our workshop.

As argued throughout *this* paper, when comparing `EvalRS` methodology to typical data challenges, we can summarize three important differentiating factors: *first*, we fight public leaderboard overfitting through our randomized evaluation loop; *second*, we discourage complex solutions that cannot be practically used, as our open source code competition provides a fixed (and reasonable) compute budget; *third* and most importantly, with a thorough evaluation with per-group and behavioral tests, we encourage participants to seek non-standard performance and discuss fairness implications.

We strongly believe these points will lay down the foundation for a first-of-its-kind automatic, shared, identifiable evaluation standard for RSs.

6. ACKNOWLEDGEMENTS

`RecList` is an open source library whose development is supported by forward looking companies in the machine learning community: the organizers wish to thank *Comet*, *Neptune*, *Gantry* for their generous support.[7]

References

[1] P. J. Chia, J. Tagliabue, F. Bianchi, C. He, B. Ko, Beyond NDCG: behavioral testing of recommender systems with reclist, CoRR abs/2111.09963 (2021). URL: https://arxiv.org/abs/2111.09963. arXiv:2111.09963.

[2] M. Schedl, The lfm-1b dataset for music retrieval and recommendation, in: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICMR '16, Association for

[7] Please check the project website for more details: https://reclist.io/.

Computing Machinery, New York, NY, USA, 2016, p. 103–110. URL: https://doi.org/10.1145/2911996.2912004. doi:10.1145/2911996.2912004.

[3] M. F. Dacrema, P. Cremonesi, D. Jannach, Are we really making much progress? a worrying analysis of recent neural recommendation approaches, in: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 101–109. URL: https://doi.org/10.1145/3298689.3347058. doi:10.1145/3298689.3347058.

[4] Z. Sun, D. Yu, H. Fang, J. Yang, X. Qu, J. Zhang, C. Geng, Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison, in: Fourteenth ACM conference on recommender systems, 2020, pp. 23–32.

[5] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, 2019, pp. 165–174.

[6] A. Rashed, S. Jawed, L. Schmidt-Thieme, A. Hintsches, Multirec: A multi-relational approach for unique item recommendation in auction systems, Fourteenth ACM Conference on Recommender Systems (2020).

[7] P. Kouki, I. Fountalis, N. Vasiloglou, X. Cui, E. Liberty, K. Al Jadda, From the lab to production: A case study of session-based recommendations in the home-improvement domain, in: Fourteenth ACM Conference on Recommender Systems, RecSys '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 140–149. URL: https://doi.org/10.1145/3383313.3412235. doi:10.1145/3383313.3412235.

[8] T. Moins, D. Aloise, S. J. Blanchard, Recseats: A hybrid convolutional neural network choice model for seat recommendations at reserved seating venues, in: Fourteenth ACM Conference on Recommender Systems, RecSys '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 309–317. URL: https://doi.org/10.1145/3383313.3412263. doi:10.1145/3383313.3412263.

[9] F. Bianchi, J. Tagliabue, B. Yu, Query2Prod2Vec: Grounded word embeddings for eCommerce, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, Association for Computational Linguistics, Online, 2021, pp. 154–162. URL: https://aclanthology.org/2021.naacl-industry.20. doi:10.18653/v1/2021.naacl-industry.20.

[10] J. Tagliabue, C. Greco, J.-F. Roy, F. Bianchi, G. Cassani, B. Yu, P. J. Chia, Sigir 2021 e-commerce workshop data challenge, in: SIGIR eCom 2021, 2021.

[11] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2015). URL: https://doi.org/10.1145/2827872. doi:10.1145/2827872.

[12] H. Zamani, M. Schedl, P. Lamere, C.-W. Chen, An analysis of approaches taken in the acm recsys challenge 2018 for automatic music playlist continuation, ACM Trans. Intell. Syst. Technol. 10 (2019). URL: https://doi.org/10.1145/3344257. doi:10.1145/3344257.

[13] D. Kotkov, J. Veijalainen, S. Wang, Challenges of serendipity in recommender systems, in: WEBIST, 2016.

[14] D. Jannach, M. Ludewig, When recurrent neural networks meet the neighborhood for session-based recommendation, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 306–310.

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

[16] M. T. Ribeiro, T. S. Wu, C. Guestrin, S. Singh, Beyond accuracy: Behavioral testing of nlp models with checklist, in: ACL, 2020.

[17] G. d. S. P. Moreira, S. Rabhi, J. M. Lee, R. Ak, E. Oldridge, Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation, in: Fifteenth ACM Conference on Recommender Systems, 2021, pp. 143–153.

[18] K. Ariu, N. Ryu, S. Yun, A. Proutière, Regret in online recommendation systems, ArXiv abs/2010.12363 (2020).

[19] J. Tagliabue, B. Yu, F. Bianchi, The Embeddings That Came in From the Cold: Improving Vectors for New and Rare Products with Content-Based Inference, Association for Computing Machinery, New York, NY, USA, 2020, p. 577–578. URL: https://doi.org/10.1145/3383313.3411477.

[20] L. Briand, G. Salha-Galvan, W. Bendada, M. Morlon, V.-A. Tran, A semi-personalized system for user cold start recommendation on music streaming apps, 2020. URL: arXiv:2106.03819.

[21] M. Hendriksen, E. Kuiper, P. Nauts, S. Schelter, M. de Rijke, Analyzing and predicting purchase intent in e-commerce: Anonymous vs. identified customers, 2020. URL: https://arxiv.org/abs/2012.08777.

[22] Krista Garcia, The impact of product recommendations, 2018. URL: https://www.emarketer.com/content/the-impact-of-product-recommendations.

[23] A. Flexer, D. Schnitzer, J. Schlueter, A mirex meta-analysis of hubness in audio music similarity, 2012.

[24] M. Twohey, G. J. Dance, Lawmakers press amazon on sales of chemical used in suicides, 2022. URL: https://www.nytimes.com/2022/02/04/technology/

amazon-suicide-poison-preservative.html.
[25] J. Tagliabue, You Do Not Need a Bigger Boat: Recommendations at Reasonable Scale in a (Mostly) Serverless and Open Stack, Association for Computing Machinery, New York, NY, USA, 2021, p. 598–600. URL: https://doi.org/10.1145/3460231.3474604.
[26] V. Batagelj, M. Zaveršnik, Generalized cores, Advances in Data Analysis and Classification 5 (2011) 129–145.
[27] M. Schedl, Investigating country-specific music preferences and music recommendation algorithms with the lfm-1b dataset, International Journal of Multimedia Information Retrieval 6 (2017) 71 – 84.
[28] J. S. Ke Yang, Measuring fairness in ranked outputs, in: SSDBM 2017: Proceedings of the 29th International Conference on Scientific and Statistical Database Management, 2017, pp. 1 – 6. URL: https://doi.org/10.1145/3085504.3085526.
[29] C. Castillo, Fairness and transparency in ranking, in: ACM SIGIR ForumVolume, volume Volume 52, 2019, pp. 64 – 71. URL: https://doi.org/10.1145/3308774.3308783.
[30] M. Zehlike, K. Yang, J. Stoyanovich, Fairness in ranking: A survey, in: TBD. ACM, 2020, pp. 1–58. URL: https://arxiv.org/pdf/2103.14000.pdf.
[31] M. O'Mahony, N. Hurley, N. Kushmerick, G. Silvestre, Collaborative recommendation: A robustness analysis, volume 4, 2004. URL: https://doi.org/10.1145/1031114.1031116.
[32] S. Saxena, S. Jain, Exploring and mitigating gender bias in recommender systems with explicit feedback (2021). URL: arXivpreprintarXiv:2112.02530.
[33] D. Kowald, M. Schedl, E. Lex, The unfairness of popularity bias in music recommendation: A reproducibility study, European conference on information retrieval (2020).
[34] Òscar Celma, P. Cano, From hits to niches? or how popular artists can bias music recommendation and discovery, in: Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, 2008. URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.168.5009&rep=rep1&type=pdf.
[35] P. Bello, D. Garcia, Cultural divergence in popular music: the increasing diversity of music consumption on spotify across countries, Humanities and Social Sciences Communications 8 (2021).
[36] M. Drosou, H. Jagadish, E. Pitoura, J. Stoyanovich, Diversity in big data: A review, Big data 5.2 (2017) 73–84.
[37] Diversity in recommender systems – a survey, Knowledge-Based Systems (2017) 154–162.
[38] M. Baigorria Alonso, Data augmentation using many-to-many rnns for session-aware recommender systems, in: ACM WSDM Workshop on Web Tourism (WSDM WebTour'21), 2021.