# ELLIOT: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation

Vito Walter Anelli*
vitowalter.anelli@poliba.it
Politecnico di Bari, Italy

Alejandro Bellogín
alejandro.bellogin@uam.es
Autónoma Madrid, Spain

Antonio Ferrara
antonio.ferrara@poliba.it
Politecnico di Bari, Italy

Daniele Malitesta
daniele.malitesta@poliba.it
Politecnico di Bari, Italy

Felice Antonio Merra
felice.merra@poliba.it
Politecnico di Bari, Italy

Claudio Pomo*
claudio.pomo@poliba.it
Politecnico di Bari, Italy

Francesco Maria Donini
donini@unitus.it
Università della Tuscia, Italy

Tommaso Di Noia
tommaso.dinoia@poliba.it
Politecnico di Bari, Italy



## ABSTRACT

Recommender Systems have shown to be an effective way to alleviate the over-choice problem and provide accurate and tailored recommendations. However, the impressive number of proposed recommendation algorithms, splitting strategies, evaluation protocols, metrics, and tasks, has made rigorous experimental evaluation particularly challenging. Puzzled and frustrated by the continuous recreation of appropriate evaluation benchmarks, experimental pipelines, hyperparameter optimization, and evaluation procedures, we have developed an exhaustive framework to address such needs. ELLIOT is a comprehensive recommendation framework that aims to run and reproduce an entire experimental pipeline by processing a simple configuration file. The framework loads, filters, and splits the data considering a vast set of strategies (13 splitting methods and 8 filtering approaches, from temporal training-test splitting to nested K-folds Cross-Validation). ELLIOT[1] optimizes hyperparameters (51 strategies) for several recommendation algorithms (50), selects the best models, compares them with the baselines providing intra-model statistics, computes metrics (36) spanning from accuracy to beyond-accuracy, bias, and fairness, and conducts statistical analysis (Wilcoxon and Paired t-test).[2]

## CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Collaborative filtering*; • **Computing methodologies** → *Learning from implicit feedback*; Neural networks; Factorization methods.

## KEYWORDS

Recommender Systems; Evaluation; Reproducibility; Bias; Fairness

**ACM Reference Format:**
Anelli et al. 2021. ELLIOT: A Comprehensive and Rigorous Framework for Reproducible Recommender Systems Evaluation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3463245

*Corresponding authors: vitowalter.anelli@poliba.it, claudio.pomo@poliba.it.
[1]https://github.com/sisinflab/elliot
[2]An extended version of this paper is available at https://arxiv.org/abs/2103.02590


*SIGIR '21, July 11–15, 2021, Virtual Event, Canada.*

ACM ISBN 978-1-4503-8037-9/21/07.
https://doi.org/10.1145/3404835.3463245

## 1 INTRODUCTION

In the last decade, Recommendation Systems (RSs) have gained momentum as the pivotal choice for personalized decision-support systems. Recommendation is essentially a retrieval task where a catalog of items is ranked and the top-scoring items are presented to the user [58]. Once it was demonstrated their ability to provide personalized items to clients, both Academia and Industry devoted their attention to RSs [4, 13, 66, 67]. This collective effort resulted in an impressive number of recommendation algorithms, ranging from memory-based [85] to latent factor-based [21, 29, 55, 76], and deep learning-based methods [63, 97]. At the same time, the RS research community became conscious that *accuracy* was not sufficient to guarantee user satisfaction [71]. *Novelty* and *diversity* [17, 46, 93] came into play as new dimensions to be analyzed when comparing algorithms. However, this was only the first step in the direction of a more comprehensive evaluation of RSs. Indeed, more recently, the presence of *biased* [9, 107] and *unfair* [23, 25, 26] recommendations towards user groups and item categories has been widely investigated. In fact, RSs have been widely studied and applied in various domains and tasks, with different (and often contradicting in their hypotheses) splitting preprocessing strategies [16] fitting the specific scenario. Moreover, machine learning (and recently also deep learning) techniques are prominent in algorithmic research and require their hyperparameter optimization strategies and procedures [6, 92].

The abundance of possible choices generated much confusion about choosing the correct baselines, conducting the hyperparameter optimization and the experimental evaluation [81, 82], and reporting the details of the adopted procedure. Consequently, two major concerns arose: unreproducible evaluation and unfair comparisons [88]. On the one hand, the negative effect of unfair comparisons is that various proposed recommendation models have been compared with suboptimal baselines [22, 79]. On the other hand, in a recent study [22], it has been shown that only one-third of the published experimental results are, in fact, reproducible. Progressively, the RS community has welcomed the emergence of recommendation, evaluation, and even hyperparameter tuning frameworks [15, 24, 31, 88, 93]. However,facilitating reproducibility or extending the provided functionality would typically depend on developing bash scripts or programming on whatever language each framework is written.

This work introduces ELLIOT, a novel kind of recommendation framework, to overcome these obstacles. The framework analyzes the recommendation problem from the researcher's perspective. Indeed, ELLIOT conducts a whole experiment, from dataset loading to results gathering. The core idea is to feed the system with a simple and straightforward configuration file that drives the framework through the experimental setting choices. ELLIOT natively provides for widespread research evaluation features, like the analysis of multiple cut-offs and several RSs (50). According to the recommendation model, the framework allows, to date, the choice among 27 similarities, the definition of multiple neural architectures, and 51 hyperparameter tuning combined approaches, unleashing the full potential of the HyperOpt library [15]. To enable the evaluation for the diverse tasks and domains, ELLIOT supplies 36 metrics (including Accuracy, Error-based, Coverage, Novelty, Diversity, Bias, and Fairness metrics), 13 splitting strategies, and 8 prefiltering policies. The framework can also measure to what extent the RS results are significantly different from each other, providing the paired t-test and Wilcoxon statistical hypothesis tests. Finally, ELLIOT lets the researcher quickly build their models and include them in the experiment.

## 2 PRIOR WORK

**Background.** RS evaluation is an active, ever-growing research topic related to reproducibility, which is a cornerstone of the scientific process as identified by Konstan and Adomavicius [53]. Recently researchers have taken a closer look at this problem, in particular because depending on how well we evaluate and assess the efficacy of a system, the significance and impact of such results will increase.

Some researchers argue that to enhance reproducibility, and to facilitate fair comparisons between different works (either frameworks, research papers, or published artifacts), at least the following four stages must be identified within the evaluation protocol [81]: *data splitting*, *item recommendations*, *candidate item generation*, and *performance measurement*. In a recent work [11], these stages have been completed with *dataset collection* and *statistical testing*. Some of these stages can be further categorized, such as performance measurement, depending on the performance dimension to be analyzed (e.g., ranking vs error, accuracy vs diversity, and so on).

In fact, the importance and relevance of the aforementioned stages have been validated in recent works; however, even though most of the RS literature has been focused on the impact of the item recommendation stage as an isolated component, this is far from being the only driver that affects RS performance or the only component impacting on its potential for reproducibility. In particular, Meng et al. [72] survey recent works in the area and conclude that no standard splitting strategy exists, in terms of random vs temporal splits; furthermore, the authors found that the selection of the splitting strategy can have a strong impact on the results. Previously, Campos et al. [16] categorized and experimented with several variations of random and temporal splitting strategies, evidencing the same inconsistency in the results. Regarding the candidate item generation, it was first shown [10] that different strategies selecting the candidate items to be ranked by the recommendation algorithm may produce results that are orders of magnitude away from each other; this was later confirmed [81] in the context of benchmarking recommendation frameworks. Recent works [58, 62] evidenced that some of these strategies selecting the candidate items may introduce inconsistent measurements which should, hence, not be trusted.

Finally, depending on the recommendation task and main goal of the RS, several performance dimensions, sometimes contradicting, can be assessed. For a classical overview of these dimensions, we refer the reader to Gunawardana and Shani [32], where metrics accounting for prediction accuracy, coverage, confidence, trust, novelty, diversity, serendipity, and so on are defined and compared. However, to the best of our knowledge, there is no public implementation providing more than one or two of these dimensions. Moreover, recently the community has considered additional dimensions such as bias (in particular, popularity bias [1]) and fairness [26]. These dimensions are gaining attention, and several metrics addressing different subtleties are being proposed, but no clear winner or standard definition emerged so far – as a consequence, the community lacks an established implementation of these novel evaluation dimensions.

**Related Frameworks.** Reproducibility is the keystone of modern RSs research. Dacrema et al. [22] and Rendle et al. [78] have recently raised the need of comprehensive and fair recommender model evaluation. Their argument on the outperforming recommendation accuracy of latent-factor models over deep-neural ones, when an extensive hyper-parameter tuning was performed, made it essential the development of novel recommendation frameworks. Starting from 2011, Mymedialite [31], LensKit [24, 27], LightFM [59], RankSys [93], and Surprise [45], have formed the basic software for rapid prototyping and testing of recommendation models, thanks to an easy-to-use model execution and the implementation of standard accuracy, and beyond-accuracy, evaluation measures and splitting techniques. However, the outstanding success and the community interests in deep learning (DL) recommendation models, raised need for novel instruments. LibRec [33], Spotlight [60], and OpenRec [100] are the first open-source projects that made DL-based recommenders available – with less than a dozen of available models without filtering, splitting, and hyper-optimization tuning strategies. An important step towards more exhaustive and up-to-date set of model implementations have been released with RecQ [102], DeepRec [35], and Cornac [83] frameworks. However, they do not provide a general tool for extensive experiments on the pre-elaboration and the evaluation of a dataset. Indeed, after the reproducibility hype [22, 78], DaisyRec [88] and RecBole [105] raised the bar of framework capabilities, making available both large set of models, data filtering/splitting operations and, above all, hyper-parameter tuning features. However, we found a significant gap in splitting and filtering capabilities, in addition to a complete lack of two nowadays popular (even critical) aspects of recommendation performance: biases and fairness. Reviewing these related frameworks, emerged a striking lack of an open-source recommendation framework able to perform *by design* an extensive set of pre-elaboration operations, to support several hyperparameters optimization strategies and multiple sets of evaluation measures, which include bias and fairness ones, supported by statistical significance tests – a feature absent in other frameworks (as of February 2021). ELLIOT meets all these needs. Table 1 gives an overview of the frameworks and to which extent they satisfy the mentioned requirements.

## 3 ELLIOT

ELLIOT is an extensible framework composed of eight functional modules, each of them responsible for a specific step of an experimental recommendation process. What happens under the hood

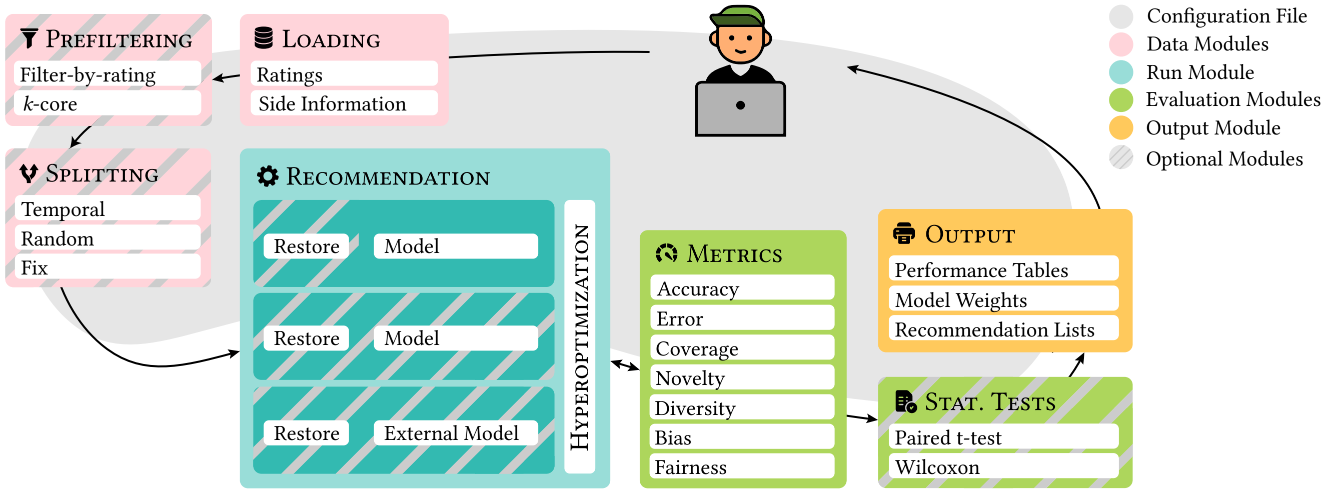


**Figure 1: Overview of Elliot.**

(Figure 1) is transparent to the user, who is only expected to provide human-level experimental flow details using a customizable configuration file. Accordingly, Elliot builds the overall pipeline. The following sections deepen into the details of the eight Elliot's modules and outline the preparation of a configuration file.

## 3.1 Data Preparation

The *Data* modules are responsible for handling and managing the experiment input, supporting various additional information, e.g., item features, visual embeddings, and images. After being loaded by the *Loading* module, the input data is taken over by *Prefiltering* and *Splitting* modules whose strategies are reported in Table 1.

***3.1.1 Loading.*** RSs experiments could require different data sources such as user-item feedback or additional side information, e.g., the visual features of an item images. To fulfill these requirements, Elliot comes with different implementations of the *Loading* module. Additionally, the user can design computationally expensive prefiltering and splitting procedures that can be stored and loaded to save future computation. Data-driven extensions can handle additional data like visual features [19, 51], and semantic features extracted from knowledge graphs [7]. Once a side-information-aware *Loading* module is chosen, it filters out the items devoiding the required information to grant a fair comparison.

***3.1.2 Prefiltering.*** After data loading, Elliot provides data filtering operations through two possible strategies. The first strategy implemented in the *Prefiltering* module is *Filter-by-rating*, which drops off a user-item interaction if the preference score is smaller than a given threshold. It can be (i) a *Numerical* value, e.g., 3.5, (ii) a *Distributional* detail, e.g., global rating average value, or (iii) a user-based distributional (*User Dist.*) value, e.g., user's average rating value. The second prefiltering strategy, *k-core*, filters out users, items, or both, with less than $k$ recorded interactions. The *k-core* strategy can proceed iteratively (*Iterative k-core*) on both users and items until the *k-core* filtering condition is met, i.e., all the users and items have at least $k$ recorded interaction. Since reaching such condition might be intractable, Elliot allows specifying the maximum number of iterations (*Iter-n-rounds*). Finally, the *Cold-Users* filtering feature allows retaining cold-users only.

***3.1.3 Splitting.*** If needed, the data is served to the *Splitting* module. In detail, Elliot provides (i) *Temporal*, (ii) *Random*, and (iii) *Fix* strategies. The *Temporal* strategy splits the user-item interactions based on the transaction timestamp, i.e., fixing the timestamp, finding the optimal one [8, 12], or adopting a hold-out (*HO*) mechanism. The *Random* strategy includes hold-out (*HO*), $K$-repeated hold-out (*K-HO*), and cross-validation (*CV*). Table 1 provides further configuration details. Finally, the *Fix* strategy exploits a precomputed splitting.

## 3.2 Recommendation Models

After data loading and pre-elaborations, *Recommendation* module (Figure 1) provides the functionalities to train (and restore) the Elliot recommendation models and the new ones integrated by users.

***3.2.1 Implemented Models.*** Elliot integrates, to date, 50 recommendation models (see Table 1) partitioned into two sets. The first set includes 38 *popular* models implemented in at least two of frameworks reviewed in this work (i.e., adopting a framework-wise popularity notion). Table 1 shows that Elliot is the framework covering the largest number of popular models, with 30 models out of 38, i.e., 79%. The second set comprises other well-known state-of-the-art recommendation models implemented in less than two frameworks, namely, BPRSLIM [73], ConvNCF [39], NPR [74], MultiDAE [63], and NAIS [41], graph-learning based, i.e., NGCF [96], and LightGCN [38], visual-based, i.e., VBPR [36], DeepStyle [65], DVBPR [51], ACF [19], and VNPR [74], adversarial-robust, i.e., APR [40] and AMR [89], generative adversarial network (GAN)-based, i.e., IRGAN [95] and CFGAN [18], content-aware, i.e., Attribute-I-$k$NN and -U-$k$NN [31], VSM [5, 75], Wide & Deep [20], and KaHFM [7] recommenders.

***3.2.2 Hyper-parameter Tuning.*** Hyperparameter tuning is an ingredient of the recommendation model training that definitely influences its performance [78]. Elliot provides *Grid Search*, *Simulated Annealing*, *Bayesian Optimization*, and *Random Search* strategies. Furthermore, Elliot allows performing four traversing strategies across the search space defined in each recommendation model configuration. When the user details the possible hyperparameters (as a list) without specifying a search strategy, Elliot automatically performs an exhaustive *Grid Search*. Elliot may exploit the full potential of

Table 1: Overview of the Elliot and related frameworks functionalities.

**DATA ELABORATION AND MODEL OPTIMIZATION STRATEGIES**

| | **Prefiltering** | | | | | | | | | **Splitting** | | | | | | | | | | | | | | | | | | | **Hyperparam. Tuning** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Filter-by-rating | | | *k*-core | | | | | | Temporal | | | | | | | Random | | | | | | | | | | Fix | | Search Strategy | | | | Search Space | | | | |
| | | | | | | | | | | *Fix* | | *HO* | | | | | *HO* | | | | | *K-HO* | | | *CV* | | — | | | | | | | | | | |
| | Numerical | Distributional | User Dist. | User | Item | Iterative | Iter-*n*-rounds | Cold-Users | **Tot.** | Best timestamp | Handcrafted | By-Ratio Sys. | By-Ratio User | Leave-1-out | Leave-*n*-in | Leave-*n*-out | By-Ratio Sys. | By-Ratio User | Leave-1-out | Leave-*n*-in | Leave-*n*-out | By-Ratio Sys. | Leave-1-out | Leave-*n*-out | *k*-folds Sys. | *k*-folds User | Computed | **Tot.** | Grid | Annealing | Bayesian | Random | Fix | Uniform | Normal | Log-Uniform | **Tot.** |
| LensKit Java [27] | | | | | | | | | 0 | | | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | ✓ | 10 | | | | | | | | | 0 |
| Mymedialite [31] | | | | | | | | | 0 | | | ✓ | ✓ | | | ✓ | ✓ | | | | | | | | ✓ | | ✓ | 6 | | | | | | | | | 0 |
| RankSys [93] | | | | | | | | | 0 | | | | | | | | | | | | | | | | | | ✓ | 1 | | | | | | | | | 0 |
| LibRec [33] | | | | | | | | | 0 | | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | ✓ | | ✓ | | ✓ | 11 | | | | | | | | | 0 |
| Implicit [28] | | | | | | | | | 0 | | | | | | | | | | | | | | | | | | ✓ | 1 | | | | | | | | | 0 |
| OpenRec [100] | | | | | | | | | 0 | | | | | | | | | | | | | | | | | | ✓ | 1 | | | | | | | | | 0 |
| RecQ [102] | | | | | | | | | 0 | | | | | | | | | ✓ | | | | ✓ | | | | | ✓ | 3 | | | | | | | | | 0 |
| DeepRec [35] | | | | | | | | | 0 | | | | | | | | ✓ | | | | | | | | | | ✓ | 2 | | | | | | | | | 0 |
| LensKit Python [24] | | | | | | | | | 0 | | | | | | | | | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | 5 | | | | | | | | | 0 |
| Surprise [45] | | | | | | | | | 0 | | | | | | | | | ✓ | ✓ | | | ✓ | ✓ | | ✓ | | ✓ | 6 | ✓ | | | ✓ | ✓ | | | | 3 |
| Cornac [83] | ✓ | | | ✓ | ✓ | | | | 3 | | | | | | | | ✓ | ✓ | | | | | | | ✓ | | ✓ | 4 | ✓ | | | ✓ | ✓ | ✓ | | | 4 |
| RecBole [105] | | | | | | | | | 0 | | | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | | | ✓ | 8 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | 7 |
| DaisyRec [88] | ✓ | | | ✓ | ✓ | | | | 3 | | | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | | | | | | | | ✓ | 7 | | | ✓ | | ✓ | ✓ | | ✓ | 4 |
| Elliot | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **8** | ✓ | ✓ | | ✓ | ✓ | | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | **13** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | **8** |

**RECOMMENDER MODELS**

| | **Popular Models** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MostPop | Random | I-*k*NN[64] | U-*k*NN[80] | PureSVD[55] | FunkSVD[29] | SVD++[54] | SLIM[73] | MF[56] | WRMF[104] | FM[76] | NeuMF[42] | BPRMF[77] | DMF[99] | FISM[50] | NNMF[68] | NFM[37] | SoREC[69] | DeepFM[34] | PMF[84] | AFM[98] | FFM[49] | WBPR[30] | DSSM[44] | VBPR[36] | ConvMF[52] | GMF[42] | Caser[90] | MultiVAE[63] | I-AutoR[87] | U-AutoR[87] | CDAE[97] | CML[43] | LogMF[48] | LDA[57] | SoMF[47] | SlopeOne[61] | SoReg[70] | **Others** | **Tot.** |
| LensKit Java [27] | | | ✓ | ✓ | | ✓ | | | ✓ | | | | | | | | | | | | | | | | | | | | | | | | | | | | ✓ | | 0 | 5 |
| Mymedialite [31] | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | | | ✓ | | | | | | | | | | | | | | | | | | | | | ✓ | | ✓ | ✓ | | 6 | 18 |
| RankSys [93] | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | | ✓ | | | | | | | | | ✓ | | | | | | | | | | | | | | | ✓ | | | | 0 | 8 |
| LibRec [33] | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | 30 | 55 |
| Implicit [28] | | | ✓ | ✓ | | | | | | | | | ✓ | | | | | | | | | | | | | | | | | | | | | ✓ | | | | | 0 | 4 |
| OpenRec [100] | | | | | | | | | | | | | ✓ | | | | | | | ✓ | | | | | ✓ | | | | | | | | ✓ | | | | | | 7 | 11 |
| RecQ [102] | | | | | ✓ | | ✓ | | | | | | | | | | | ✓ | | ✓ | | | | | | | | | | | | | | | | ✓ | ✓ | ✓ | 6 | 13 |
| DeepRec [35] | | | | | | | | | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | | | ✓ | | | ✓ | | | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | | | 5 | 20 |
| LensKit Python [24] | ✓ | | ✓ | ✓ | | ✓ | | | ✓ | | | | ✓ | | | | | | | | | | | | | | | | | | | | | | | | | | 0 | 6 |
| Surprise [45] | | | ✓ | ✓ | ✓ | | ✓ | | | | | | | | | ✓ | | | | | | | | | | | | | | | | | | | | | ✓ | | 1 | 7 |
| Cornac [83] | ✓ | | ✓ | ✓ | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | ✓ | | ✓ | | | ✓ | | ✓ | ✓ | ✓ | | ✓ | | | | | | | | | | 21 | 35 |
| RecBole [105] | ✓ | | ✓ | | | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | | | ✓ | | | | | | | 50 | **65** |
| DaisyRec [88] | ✓ | | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | | ✓ | | ✓ | | | | | | | | ✓ | | | ✓ | | | | | | | 0 | 14 |
| Elliot | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | ✓ | | | ✓ | | 20 | 50 |

**EVALUATION OF RECOMMENDATION PERFORMANCE**

| | **Metric Families** | | | | | | | | **Statistical Tests** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Error | Coverage | Novelty | Diversity | Bias | Fairness | **Tot.** | Paired t-test | Wilcoxon | **Tot.** |
| LensKit Java [27] | 2 | 2 | 1 | | 1 | | | 6 | | | 0 |
| Mymedialite [31] | 6 | 2 | | | | | | 8 | | | 0 |
| RankSys [93] | 7 | | **3** | **6** | **18** | | | 34 | | | 0 |
| LibRec [33] | 6 | **4** | | | | | | 10 | | | 0 |
| Implicit [28] | | | | | | | | 0 | | | 0 |
| OpenRec [100] | 4 | | | | | | | 4 | | | 0 |
| RecQ [102] | 6 | 2 | | | | | | 8 | | | 0 |
| DeepRec [35] | 6 | 2 | | | | | | 8 | | | 0 |
| LensKit Python [24] | 4 | 2 | | | | | | 6 | | | 0 |
| Surprise [45] | | **4** | | | | | | 4 | | | 0 |
| Cornac [83] | 8 | 3 | | | | | | 11 | | | 0 |
| RecBole [105] | 8 | 3 | | | | | | 11 | | | 0 |
| DaisyRec [88] | 8 | | | | | | | 8 | | | 0 |
| Elliot | **11** | 3 | **3** | 2 | 3 | **10** | **4** | **36** | ✓ | ✓ | **2** |

the *HyperOpt* [15] library by considering all its sampling strategies. Table 1 summarizes the available *Search Strategies* and *Search Spaces*.

## 3.3 Performance Evaluation

After the training phase, ELLIOT continues its operations, evaluating recommendations. Figure 1 indicates this phase with two distinct evaluation modules: Metrics and Statistical Tests.

***3.3.1 Metrics.*** ELLIOT provides 36 evaluation metrics (see Table 1), partitioned into seven families: *Accuracy* [86, 106], *Error*, *Coverage*, *Novelty* [94], *Diversity* [103], *Bias* [2, 3, 91, 101, 109], and *Fairness* [23, 108]. It is worth mentioning that ELLIOT is the framework that exposes both the largest number of metrics and the only one considering bias and fairness measures. Moreover, the user can choose any metric to drive the model selection and the tuning.

***3.3.2 Statistical Tests.*** Table 1 shows that the reviewed related frameworks miss statistical hypothesis tests. This is probably due to the need to compute fine-grained (e.g., per-user or per-partition) results and retain them for each recommendation model. It implies that the framework should be designed for multi-recommender evaluation and handling the fine-grained results. ELLIOT brings the opportunity to compute two statistical hypothesis tests, i.e., *Wilcoxon* and *Paired t-test*, activating a flag in the configuration file.

## 3.4 Framework Outcomes

When the experiment finishes, it is time for ELLIOT to collect the results through the *Output* module in Figure 1. ELLIOT gives the possibility to store three classes of output reports: (i) *Performance Tables*, (ii) *Model Weights*, and (iii) *Recommendation Lists*. The former consist of spreadsheets (in a *tab-separated-value* format) with all the metric values computed on the test set for every recommendation model specified in the configuration file. The tables comprise cut-off specific and model-specific tables (i.e., considering each combination of the explored parameters). The user can also choose to store tables with the triple format, i.e., *<Model, Metric, Value>*. Tables also include cut-off-specific statistical hypothesis tests and a JSON file that summarizes the best model parameters. Optionally, ELLIOT saves model weights to avoid future re-training of the recommender. Finally, ELLIOT stores the top-$k$ recommendation lists for each model adopting a tab-separated *<User, Item, Predicted Score>* triple-based format.

## 3.5 Preparation of the Experiment

The operation of ELLIOT is triggered by a single configuration file written in YAML. Configuration 1 shows a toy example of a configuration file. The first section details the data loading, filtering, and splitting information as defined in Section 3.1. The `models` section represents the recommendation models configuration, e.g., Item-$k$NN, described in Section 3.2.1. Here, the model-specific hyperparameter optimization strategies are specified (Section 3.2.2), e.g., the grid-search in Configuration 1. The `evaluation` section details the evaluation strategy with the desired metrics (Section 3.3), e.g., nDCG in the toy example. Finally, `save_recs` and `top_k` keys detail, for example, the *Output* module abilities described in Section 3.4. It is worth noticing that, to the best of our knowledge, ELLIOT is the only framework able to run an extensive set of reproducible experiments by merely preparing a single configuration file. Section 4 exemplifies two real experimental scenarios commenting on the salient parts of the configuration files.

**Configuration 1: `hello_world.yml`**

```yaml
experiment:
  dataset: movielens_1m
  data_config:
    strategy: dataset
    dataset_path: ../data/movielens_1m/dataset.tsv
  splitting:
    test_splitting:
        strategy: random_subsampling
        test_ratio: 0.2
  models:
    ItemKNN:
      meta:
        hyper_opt_alg: grid
        save_recs: True
      neighbors: [50, 100]
      similarity: cosine
  evaluation:
    simple_metrics: [nDCG]
  top_k: 10
```

# 4 EXPERIMENTAL SCENARIOS

We illustrate how to prepare, execute and evaluate a *basic* and a more *advanced* experimental scenario with ELLIOT.

## 4.1 Basic Configuration

**Experiment.** In the first scenario, the experiments require comparing a group of RSs whose parameters are optimized via a grid-search. Configuration 2 specifies the data loading information, i.e., semantic features source files, in addition to the filtering and splitting strategies. In particular, the latter supplies an entirely automated way of preprocessing the dataset, which is often a time-consuming and non-easily-reproducible phase. The `simple_metrics` field allows computing accuracy and beyond-accuracy metrics, with two top-$k$ cut-off values (5 and 10) by merely inserting the list of desired measures, e.g., `[Precision, nDCG, ...]`. The knowledge-aware recommendation model, `AttributeItemKNN`, is compared against two baselines: `Random` and `ItemKNN`, along with a user-implemented model that is `external.MostPop`. The configuration makes use of ELLIOT's feature of conducting a grid search-based hyperparameter optimization strategy by merely passing a list of possible hyperparameter values, e.g., `neighbors: [50, 70, 100]`. The reported models are selected according to *nDCG@10*.

**Results.** Table 2 displays a portion of experimental results generated by feeding ELLIOT with the configuration file. The table reports four metric values computed on recommendation lists at cutoffs 5 and 10 generated by the models selected after the hyperparameter tuning phase. For instance, Attribute-I-$k$NN model reports values for the configuration with `neighbors` set to `100` and `similarity` set to `braycurtis`. Table 2 confirms some common findings: the item coverage value (*ICov*@10) of an Attribute-I-$k$NN model is higher than the one measured on I-$k$NN, and I-$k$NN is the most accurate model.

## 4.2 Advanced Configuration

**Experiment.** The second scenario depicts a more complex experimental setting. In Configuration 3, the user specifies an elaborate data splitting strategy, i.e., `random_subsampling` (for test splitting) and `random_cross_validation` (for model selection), by setting few splitting configuration fields. Configuration 3 does not provide a cut-off value, and thus a top-$k$ field value of 50 is assumed as the cut-off. Moreover, the evaluation section includes the `UserMADrating`

**Configuration 2: `basic_configuration.yml`**

```yaml
experiment:
  dataset: cat_dbpedia_movielens_1m
  data_config:
    strategy: dataset
    dataloader: KnowledgeChainsLoader
    dataset_path: <...>/dataset.tsv
    side_information:
      <...>
  prefiltering:
    strategy: user_average
  splitting:
    test_splitting:
      strategy: temporal_hold_out
      test_ratio: 0.2
    <...>
  external_models_path: ../external/models/__init__.py
  models:
    Random:
      <...>
    external.MostPop:
      <...>
    AttributeItemKNN:
      neighbors: [50, 70, 100]
      similarity: [braycurtis, manhattan]
      <...>
  evaluation:
    cutoffs: [10, 5]
    evaluation: [nDCG, Precision, ItemCoverage, EPC, Gini]
    relevance_threshold: 1
  top_k: 50
```

https://github.com/sisinflab/elliot/blob/master/config_files/basic_configuration.yml

**Table 2: Experimental results for Configuration 2.**

| Model | *nDCG@5* | *ICov@5* | *nDCG@10* | *ICov@10* |
|---|---|---|---|---|
| Random | 0.0098 | 3197 | 0.0056 | 3197 |
| MostPop | 0.0699 | 68 | 0.0728 | 96 |
| I-*k*NN | 0.0791 | 448 | 0.0837 | 710 |
| Attribute-I-*k*NN | 0.0464 | 1575 | 0.0485 | 2102 |

metric. Elliot considers it as a complex metric since it requires additional arguments (as shown in Configuration 3). The user also wants to implement a more advanced hyperparameter tuning optimization. For instance, regarding NeuMF, Bayesian optimization using *Tree of Parzen Estimators* [14] is required (i.e., `hyper_opt_alg: tpe`) with a logarithmic uniform sampling for the learning rate search space. Moreover, Elliot allows considering complex neural architecture search spaces by inserting lists of tuples. For instance, `(32, 16, 8)` indicates that the neural network consists of three hidden layers with 32, 16, and 8 units, respectively.

**Results.** Table 3 provides a summary of the experimental results obtained feeding Elliot with Configuration 3. Even here, the columns report the values for all the considered metrics (simple and complex metrics). Configuration 3 also requires statistical hypothesis tests. Therefore, the table reports the *Wilcoxon-test* outcome (computed on pairs of models with their best configuration). MultiVAE, coherently with the literature, outperforms the other baselines.

## 5 CONCLUSION

Elliot is a framework that examines the recommendation process from an RS researcher's perspective. It requires the user just to compile a flexible configuration file to conduct a rigorous and reproducible experimental evaluation. The framework provides several loading, prefiltering, splitting, hyperparameter optimization strategies, recommendation models, and statistical hypothesis tests. Elliot reports can be directly analyzed and inserted into research papers. We reviewed the RS evaluation literature, positioning Elliot among the existing frameworks, and highlighting its advantages and limitations. Next, we explored the framework architecture and how to build a working (and reproducible) experimental benchmark. To the best of our knowledge, Elliot is the first recommendation framework that provides a full multi-recommender experimental pipeline based on a simple configuration file. We plan to extend soon Elliot in various directions to include: sequential recommendation scenarios, adversarial attacks, reinforcement learning-based recommendation systems, differential privacy facilities, sampled evaluation, and distributed recommendation.

**Configuration 3: `advanced_configuration.yml`**

```yaml
experiment:
  dataset: movielens_1m
  data_config:
    strategy: dataset
    dataset_path: <...>/dataset.tsv
  prefiltering:
    strategy: iterative_k_core
    core: 10
  splitting:
    test_splitting:
      strategy: random_subsampling
      test_ratio: 0.2
    validation_splitting:
      strategy: random_cross_validation
      folds: 5
  models:
    BPRMF:
      <...>
    NeuMF:
      meta:
        hyper_max_evals: 5
        hyper_opt_alg: tpe
      lr: [loguniform, -10, -1]
      mf_factors: [quniform, 8, 32, 1]
      mlp_hidden_size: [(32, 16, 8), (64, 32, 16)]
      <...>
    MultiVAE:
      <...>
  evaluation:
    simple_metrics: [nDCG, ARP, ACLT]
    wilcoxon_test: True
    complex_metrics:
    - metric: UserMADrating
      clustering_name: Happiness
      clustering_file: <...>/u_happy.tsv
    relevance_threshold: 1
  top_k: 50
```

https://github.com/sisinflab/elliot/blob/master/config_files/advanced_configuration.yml

**Table 3: Experimental results for Configuration 3.**

| Model | *nDCG@50* | *ARP@50* | *ACLT@50* | *UMAD$_H$@50* |
|---|---|---|---|---|
| BPRMF | 0.2390 | 1096 | 0.0420 | 0.0516 |
| NeuMF | 0.2585 | 919 | 0.8616 | 0.0032 |
| MultiVAE | 0.2922† | 755† | 3.2871† | 0.1588 |

†*p-value* $\leq$ 0.001 *using Wilcoxon-test*

## ACKNOWLEDGMENTS

The authors acknowledge partial support of the projects: Servizi Locali 2.0, PON ARS01_00876 Bio-D, PON ARS01_00821 FLET4.0, PON ARS01_00917 OK-INSAID, H2020 PASSPARTOUT, PID2019-108965GB-I00

## REFERENCES


[1] Himan Abdollahpouri. 2019. Popularity Bias in Ranking and Recommendation. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*, Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor (Eds.). ACM, 529–530.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*, Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin (Eds.). ACM, 42–46.

[3] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking. In *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019*, Roman Barták and Keith W. Brawner (Eds.). AAAI Press, 413–418.

[4] Vito Walter Anelli, Amra Delic, Gabriele Sottocornola, Jessie Smith, Nazareno Andrade, Luca Belli, Michael M. Bronstein, Akshay Gupta, Sofia Ira Ktena, Alexandre Lung-Yut-Fong, Frank Portman, Alykhan Tejani, Yuanpu Xie, Xiao Zhu, and Wenzhe Shi. 2020. RecSys 2020 Challenge Workshop: Engagement Prediction on Twitter's Home Timeline. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 623–627.

[5] V. W. Anelli, T. Di Noia, E. Di Sciascio, A. Ragone, and J. Trotta. 2020. Semantic Interpretation of Top-N Recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1.

[6] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Claudio Pomo, and Azzurra Ragone. 2019. On the discriminative power of hyper-parameters in cross-validation and how to choose them. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 447–451.

[7] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Azzurra Ragone, and Joseph Trotta. 2019. How to Make Latent Factors Interpretable by Feeding Factorization Machines with Knowledge Graphs. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11778)*, Chiara Ghidini, Olaf Hartig, Maria Maleshkova, Vojtech Svátek, Isabel F. Cruz, Aidan Hogan, Jie Song, Maxime Lefrançois, and Fabien Gandon (Eds.). Springer, 38–56.

[8] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, Azzurra Ragone, and Joseph Trotta. 2019. Local Popularity and Time in top-N Recommendation. In *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11437)*, Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra (Eds.). Springer, 861–868.

[9] Ricardo Baeza-Yates. 2020. Bias in Search and Recommender Systems. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 2.

[10] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2011. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 333–336.

[11] Alejandro Bellogín and Alan Said. 2021. Improving Accountability in Recommender Systems Research Through Reproducibility. *CoRR* abs/2102.00482 (2021).

[12] Alejandro Bellogín and Pablo Sánchez. 2017. Revisiting Neighbourhood-Based Recommenders For Temporal Scenarios. In *Proceedings of the 1st Workshop on Temporal Reasoning in Recommender Systems co-located with 11th International Conference on Recommender Systems (RecSys 2017), Como, Italy, August 27-31, 2017 (CEUR Workshop Proceedings, Vol. 1922)*, Mária Bieliková, Veronika Bogina, Tsvi Kuflik, and Roy Sasson (Eds.). CEUR-WS.org, 40–44.

[13] James Bennett and Stan Lanning. 2007. The netflix prize. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*. ACM.

[14] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger (Eds.). 2546–2554.

[15] James Bergstra, Daniel Yamins, and David D. Cox. 2013. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013 (JMLR Workshop and Conference Proceedings, Vol. 28)*. JMLR.org, 115–123.

[16] Pedro G. Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Model. User Adapt. Interact.* 24, 1-2 (2014), 67–119.

[17] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 881–918.

[18] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A Generic Collaborative Filtering Framework based on Generative Adversarial Networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 137–146.

[19] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 335–344.

[20] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016*, Alexandros Karatzoglou, Balázs Hidasi, Domonkos Tikk, Oren Sar Shalom, Haggai Roitman, Bracha Shapira, and Lior Rokach (Eds.). ACM, 7–10.

[21] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker (Eds.). ACM, 39–46.

[22] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. 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 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 101–109.

[23] Yashar Deldjoo, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogin, and Tommaso Di Noia. 2020. A flexible framework for evaluating user and item fairness in recommender systems. *User Modeling and User-Adapted Interaction* (2020), 1–47.

[24] Michael D. Ekstrand. 2020. LensKit for Python: Next-Generation Software for Recommender Systems Experiments. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, Mathieu d'Aquin, Stefan Dietze, Claudia Hauff, Edward Curry, and Philippe Cudré-Mauroux (Eds.). ACM, 2999–3006.

[25] Michael D. Ekstrand, Robin Burke, and Fernando Diaz. 2019. Fairness and discrimination in recommendation and retrieval. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 576–577.

[26] Michael D. Ekstrand, Robin Burke, and Fernando Diaz. 2019. Fairness and Discrimination in Retrieval and Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 1403–1404.

[27] Michael D. Ekstrand, Michael Ludwig, Joseph A. Konstan, and John Riedl. 2011. Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 133–140.

[28] Ben Frederickson. 2018. Fast python collaborative filtering for implicit datasets.

[29] Simon Funk. 2006. Netflix update: Try this at home.

[30] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Personalized Ranking for Non-Uniformly Sampled Items. In *Proceedings of KDD Cup 2011 competition, San Diego, CA, USA, 2011 (JMLR Proceedings, Vol. 18)*, Gideon Dror, Yehuda Koren, and Markus Weimer (Eds.). JMLR.org, 231–247.

[31] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. MyMediaLite: a free recommender system library. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 305–308.

[32] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 265–308.

[33] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems. In *Posters, Demos, Late-breaking Results*

*and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015), Dublin, Ireland, June 29 - July 3, 2015 (CEUR Workshop Proceedings, Vol. 1388)*, Alexandra I. Cristea, Judith Masthoff, Alan Said, and Nava Tintarev (Eds.). CEUR-WS.org.

[34] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 1725–1731.

[35] Udit Gupta, Samuel Hsia, Vikram Saraph, Xiaodong Wang, Brandon Reagen, Gu-Yeon Wei, Hsien-Hsin S. Lee, David Brooks, and Carole-Jean Wu. 2020. DeepRecSys: A System for Optimizing End-To-End At-Scale Neural Recommendation Inference. In *47th ACM/IEEE Annual International Symposium on Computer Architecture, ISCA 2020, Valencia, Spain, May 30 - June 3, 2020*. IEEE, 982–995.

[36] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 144–150.

[37] Xiangnan He and Tat-Seng Chua. 2017. Neural Factorization Machines for Sparse Predictive Analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 355–364.

[38] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648.

[39] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer Product-based Neural Collaborative Filtering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 2227–2233.

[40] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 355–364.

[41] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Trans. Knowl. Data Eng.* 30, 12 (2018), 2354–2366.

[42] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 173–182.

[43] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge J. Belongie, and Deborah Estrin. 2017. Collaborative Metric Learning. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 193–201.

[44] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 2333–2338.

[45] Nicolas Hug. 2020. Surprise: A Python library for recommender systems. *J. Open Source Softw.* 5, 52 (2020), 2174.

[46] Neil Hurley and Mi Zhang. 2011. Novelty and Diversity in Top-N Recommendation - Analysis and Evaluation. *ACM Trans. Internet Techn.* 10, 4 (2011), 14:1–14:30.

[47] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker (Eds.). ACM, 135–142.

[48] Christopher C Johnson. 2014. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems* 27, 78 (2014), 1–9.

[49] Yu-Chin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware Factorization Machines for CTR Prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells (Eds.). ACM, 43–50.

[50] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: factored item similarity models for top-N recommender systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy (Eds.). ACM, 659–667.

[51] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian J. McAuley. 2017. Visually-Aware Fashion Recommendation and Design with Generative Image Models. In *2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017*, Vijay Raghavan, Srinivas Aluru, George Karypis, Lucio Miele, and Xindong Wu (Eds.). IEEE Computer Society, 207–216.

[52] Dong Hyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. 2016. Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells (Eds.). ACM, 233–240.

[53] Joseph A. Konstan and Gediminas Adomavicius. 2013. Toward identification and adoption of best practices in algorithmic recommender systems research. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, RepSys 2013, Hong Kong, China, October 12, 2013*, Alejandro Bellogín, Pablo Castells, Alan Said, and Domonkos Tikk (Eds.). ACM, 23–28.

[54] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, Ying Li, Bing Liu, and Sunita Sarawagi (Eds.). ACM, 426–434.

[55] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 77–118.

[56] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[57] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent dirichlet allocation for tag recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*, Lawrence D. Bergman, Alexander Tuzhilin, Robin D. Burke, Alexander Felfernig, and Lars Schmidt-Thieme (Eds.). ACM, 61–68.

[58] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1748–1757.

[59] Maciej Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015. (CEUR Workshop Proceedings, Vol. 1448)*, Toine Bogers and Marijn Koolen (Eds.). CEUR-WS.org, 14–21.

[60] Maciej Kula. 2017. Spotlight. https://github.com/maciejkula/spotlight.

[61] Daniel Lemire and Anna Maclachlan. 2005. Slope One Predictors for Online Rating-Based Collaborative Filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, Newport Beach, CA, USA, April 21-23, 2005*, Hillol Kargupta, Jaideep Srivastava, Chandrika Kamath, and Arnold Goodman (Eds.). SIAM, 471–475.

[62] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. 2020. On Sampling Top-K Recommendation Evaluation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 2114–2124.

[63] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 689–698.

[64] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Comput.* 7, 1 (2003), 76–80.

[65] Qiang Liu, Shu Wu, and Liang Wang. 2017. DeepStyle: Learning User Preferences for Visual Recommendation. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 841–844.

[66] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 462–466.

[67] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2021. Empirical analysis of session-based recommendation algorithms. *User Model. User Adapt. Interact.* 31, 1 (2021), 149–181.

[68] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. 2014. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. *IEEE Trans. Ind. Informatics* 10, 2 (2014), 1273–1284.

[69] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. SoRec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz,

Key-Sun Choi, and Abdur Chowdhury (Eds.). ACM, 931–940.

[70] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, Irwin King, Wolfgang Nejdl, and Hang Li (Eds.). ACM, 287–296.

[71] Sean M. McNee, John Riedl, and Joseph A. Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, Gary M. Olson and Robin Jeffries (Eds.). ACM, 1097–1101.

[72] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2020. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 681–686.

[73] Xia Ning and George Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, Diane J. Cook, Jian Pei, Wei Wang, Osmar R. Zaïane, and Xindong Wu (Eds.). IEEE Computer Society, 497–506.

[74] Wei Niu, James Caverlee, and Haokai Lu. 2018. Neural Personalized Ranking for Image Recommendation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 423–431.

[75] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito, and Markus Zanker. 2012. Linked open data to support content-based recommender systems. In *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*, Valentina Presutti and Helena Sofia Pinto (Eds.). ACM, 1–8.

[76] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu (Eds.). IEEE Computer Society, 995–1000.

[77] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461.

[78] Steffen Rendle, Walid Krichene, Li Zhang, and John R. Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 240–248.

[79] Steffen Rendle, Li Zhang, and Yehuda Koren. 2019. On the Difficulty of Evaluating Baselines: A Study on Recommender Systems. *CoRR* abs/1905.01395 (2019).

[80] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *CSCW '94, Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, NC, USA, October 22-26, 1994*, John B. Smith, F. Donelson Smith, and Thomas W. Malone (Eds.). ACM, 175–186.

[81] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, Alfred Kobsa, Michelle X. Zhou, Martin Ester, and Yehuda Koren (Eds.). ACM, 129–136.

[82] Alan Said and Alejandro Bellogín. 2014. Rival: a toolkit to foster reproducibility in recommender system evaluation. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*, Alfred Kobsa, Michelle X. Zhou, Martin Ester, and Yehuda Koren (Eds.). ACM, 371–372.

[83] Aghiles Salah, Quoc-Tuan Truong, and Hady W. Lauw. 2020. Cornac: A Comparative Framework for Multimodal Recommender Systems. *J. Mach. Learn. Res.* 21 (2020), 95:1–95:5.

[84] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis (Eds.). Curran Associates, Inc., 1257–1264.

[85] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko (Eds.). ACM, 285–295.

[86] G. Schröder, M. Thiele, and W. Lehner. 2011. Setting goals and choosing metrics for recommender system evaluations. *CEUR Workshop Proceedings* 811 (2011), 78–85.

[87] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi (Eds.). ACM, 111–112.

[88] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 23–32.

[89] Jinhui Tang, Xiaoyu Du, Xiangnan He, Fajie Yuan, Qi Tian, and Tat-Seng Chua. 2020. Adversarial Training Towards Robust Multimedia Recommender System. *IEEE Trans. Knowl. Data Eng.* 32, 5 (2020), 855–867.

[90] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 565–573.

[91] Virginia Tsintzou, Evaggelia Pitoura, and Panayiotis Tsaparas. 2019. Bias Disparity in Recommendation Systems. In *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019 (CEUR Workshop Proceedings, Vol. 2440)*, Robin Burke, Himan Abdollahpouri, Edward C. Malthouse, K. P. Thai, and Yongfeng Zhang (Eds.). CEUR-WS.org.

[92] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. 2018. On the robustness and discriminative power of information retrieval metrics for top-N recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan (Eds.). ACM, 260–268.

[93] Saúl Vargas. 2014. Novelty and diversity enhancement and evaluation in recommender systems and information retrieval. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, Shlomo Geva, Andrew Trotman, Peter Bruza, Charles L. A. Clarke, and Kalervo Järvelin (Eds.). ACM, 1281.

[94] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius (Eds.). ACM, 109–116.

[95] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 515–524.

[96] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 165–174.

[97] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, Paul N. Bennett, Vanja Josifovski, Jennifer Neville, and Filip Radlinski (Eds.). ACM, 153–162.

[98] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 3119–3125.

[99] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 3203–3209.

[100] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, and Deborah Estrin. 2018. OpenRec: A Modular Framework for Extensible and Adaptable Recommendation Algorithms. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, Yi Chang, Chengxiang Zhai, Yan Liu, and Yoelle Maarek (Eds.). ACM, 664–672.

[101] Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. 2012. Challenging the Long Tail Recommendation. *Proc. VLDB Endow.* 5, 9 (2012), 896–907.

[102] Junliang Yu, Min Gao, Hongzhi Yin, Jundong Li, Chongming Gao, and Qinyong Wang. 2019. Generating Reliable Friends via Adversarial Training to Improve Social Recommendation. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, Jianyong Wang, Kyuseok Shim, and Xindong Wu (Eds.). IEEE, 768–777.

[103] ChengXiang Zhai, William W. Cohen, and John D. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, Charles L. A. Clarke, Gordon V. Cormack, Jamie Callan, David Hawking, and Alan F. Smeaton (Eds.). ACM, 10–17.

[104] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. 2006. Learning from Incomplete Ratings Using Non-negative Matrix Factorization. In *Proceedings of the Sixth SIAM International Conference on Data Mining, April 20-22, 2006, Bethesda, MD, USA*, Joydeep Ghosh, Diane Lambert, David B. Skillicorn, and Jaideep Srivastava (Eds.). SIAM, 549–553.

[105] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. *CoRR* abs/2011.01731 (2020).

[106] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 1059–1068.

[107] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel*. ACM.

[108] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-Aware Tensor-Based Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 1153–1162.

[109] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 449–458.