Explaining Recommender Systems Fairness and Accuracy through the Lens of Data Characteristics

Yashar Deldjoo

Polytechnic University of Bari, Italy

Alejandro Bellogin

Universidad Autónoma de Madrid, Spain

Tommaso Di Noia

Polytechnic University of Bari, Italy

Abstract

The impact of data characteristics on the performance of classical recommender systems has been recently investigated and produced fruitful results about the relationship they have with recommendation accuracy. This work provides a systematic study on the impact of broadly chosen data characteristics (DCs) of recommender systems. This is applied to the accuracy and fairness of several variations of CF recommendation models. We focus on a suite of DCs that capture properties about the structure of the user-item interaction matrix, the rating frequency, item properties, or the distribution of rating values. Experimental validation of the proposed system involved large-scale experiments by performing 23,400 recommendation simulations on three real-world datasets in the movie (`ML-100K` and `ML-1M`) and book domains (`BookCrossing`). The validation results show that the investigated DCs in some cases can have up to 90% of explanatory power – on several variations of classical CF algorithms –, while they can explain — in the best case — about 40% of fairness results (measured according to user gender and age sensitive attributes). Therefore, this work evidences that it is more difficult to explain variations in performance when dealing with fairness dimension than accuracy.

Keywords: Explanatory power, Fairness, Accuracy, Collaborative filtering, Data Characteristics

1. Introduction

Recommender Systems (RSs) are widely used nowadays, after a dramatic expansion over the last decade. Companies either from the entertainment domain (such as Netflix or YouTube), human resources (LinkedIn), tourism

(Yelp, Trivago [1]), or finances (BBVA[1] or MPS [2] banks) exploit these techniques to increase their revenues by engaging with users in a more dynamic and personalized way. The key assumption of these approaches is that users who shared similar preferences in the past will likely agree in the future as well. Then, from an algorithmic point of view, these models keep track of users' historical behavior data (users' interactions and stated preferences) and find similar behavioral patterns to offer personalized new suggestions. However, even though these techniques tend to work *well* in most domains (as long as enough data is collected from users, to avoid the so-called cold-start problem), it is still not well-understood why some methods seem to be more suitable in some situations than others.

On top of that, there is a recent trend in the community towards shifting from the classical view where performance is equated to accuracy, to acknowledge (and aiming at improving) other dimensions. One key dimension, especially for some of the aforementioned domains, is fairness (understood as the capability of providing *comparable* recommendations to multiple groups of users, in particular, defined based on sensitive attributes such as gender or race) – for example, in [3] there is an example in the finance domain, for a broad survey on this concept we refer the reader to [4]. Defining when a recommendation is fair is not a trivial task, it depends on the context, the goal of the system, and the types of users. In the literature, it is possible to find different notions for this (ranging from the equal utility for users in the different groups, as in [5] to other approaches where it is based on merits and needs defined by the system developer [6]).

In this context, we propose herein a framework that helps to understand how recommendation algorithms behave as the underlying data characteristics on which they are trained change. For this, we focus on two competing evaluation dimensions: accuracy and fairness. As we shall show, we use a statistical model to identify the dataset characteristics that impact the most in the performance of different families of recommendation models; such impact will be referred to as *explanatory power* as it reflects the capability of such characteristic to influence a given definition of performance.

Our work was inspired by the work in [7], which studies the influence of rating data characteristics on the recommendation performance of popular collaborative RS, and by [8] where the authors use an explanatory framework to mainly focus on the robustness of CF models. Their work differs from ours because we utilize the explanatory model to explain more than one evaluation dimension with respect to many more data characteristics. Moreover, in [7] the authors only focus on error-based metrics, which have a very limited correlation with user satisfaction, as acknowledged by the community in recent years [9].

In particular, in this work we aim to address the following research questions:

RQ1 Which data characteristics impact the most in the performance of different families of recommendation algorithms when optimizing for accuracy? In particular, is it possible to capture (or predict) such performance with a

[1] `https://www.bbvadata.com/recsys/`, retrieved in December 2020.

minimal subset of these characteristics? How general are these characteristics for different datasets?

RQ2 How do these characteristics change when the goal of the system is shifted towards fairness? In comparison with the previous scenario, is it easier to predict the impact of these characteristics for fairness or for accuracy?

RQ3 Is it possible to augment the set of characteristics so that the inherent biases in the data are also considered?

The main contributions of this work are the following:

- We present a systematic, in-depth exploratory analysis of the impact of data characteristics on the performance of popular recommendation models, targeted at accuracy and fairness evaluation dimensions. To investigate the relationship between data characteristics and the performance of these models, we use regression-based explanatory modeling.
- We extend prior works on the definition and exploration of data characteristics, either based on the standard user-rating matrix, or from additional information regarding sensitive attributes related to the expected definition of the fairness dimension.
- We conduct extensive empirical analysis against a wide range of recommendation models across real-world datasets (where sensitive attributes are available). We rely on a statistical significance test with informed p-values to validate the hypotheses regarding the impact on the final model output according to the explanatory regression framework of the considered data characteristics; moreover, we exploit further statistical techniques to perform a selection of these characteristics and derive a minimum set with maximum explanatory power.

2. Background and related works

This section introduces the basic concepts of recommender systems (Section 2.1) and their evaluation (Section 2.2). At the end of this section, we present research works that we consider related to the research presented herein since their main goal is also understanding (or explaining) why and on which scenarios a recommender system reaches some performance level (Section 2.3).

2.1. Recommender systems

The recommendation problem is typically defined as finding a utility function to automatically predict how much a user will like an item that is unknown to her. More specifically, let $\mathcal{U}$ and $\mathcal{I}$ denote a set of users and items in a system, respectively. Given a utility function $g : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$, this problem is reduced to optimize the following function:

$$\forall u \in \mathcal{U}, i_u^* = \arg\max_{i \in \mathcal{I}} g(u, i) \tag{1}$$

as long as (as it is typically assumed) the item i_u^* was not enjoyed by user u before. Moreover, the classical scenario also requires a user-item rating matrix (URM) $R \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, where each entry $r_{ui} \in R$ represents a rating assigned by user $u \in \mathcal{U}$ to item $i \in \mathcal{I}$.

Most of the efforts on the RS community are devoted to finding and learning better utility functions g. Depending on the knowledge used to derive these functions, several categories have been proposed: based on preference patterns between users and items (collaborative filtering, or CF), based on similar items liked in the past by the users where the similarities could be based on textual data [10] or multimedia content [11, 12, 13] (content-based filtering, or CBF), based on the preferences of friends or social connections (social filtering), based on user demographics, and so on [14]. Among these alternatives, CF techniques are the most popular and effective ones, since they work well when enough user preferences are known [15], and do not need additional metadata or item information like other techniques.

2.2. Evaluating accuracy and fairness

RS evaluation has been traditionally linked to the analysis of the relevance of the recommendations using Information Retrieval (IR) metrics such as Precision, MAP, or nDCG [16] normally in a cross-validation (random) evaluation methodology [17]. Nonetheless, some researchers alerted about the use of more realistic evaluation methodologies by taking the interaction time into account when creating the splits [18]. The use of such methodologies is not straightforward, and there are several options worth of exploration, impacting the results of the algorithms and how realistic (or transferable to the real world) these results could be [18].

Moreover, despite the importance of relevance in recommendations, there has been a growing awareness on measuring other evaluation dimensions like novelty and diversity, as sometimes producing only accurate recommendations may not surprise or discover new items to the target user [19]. The document recently released by the European Union on guidelines on ethics in AI[2], shed light on the ethical rules that are now recommended when designing, developing, deploying, implementing, or using AI products. The key EU requirements for achieving trustworthy RS include robustness of RS [20, 21], privacy and data governance [22, 23], transparency [24], nondiscrimination and fairness [25, 26], societal and environmental well-being, and accountability [27, 28]. We focus our attention mainly on the fairness dimension. From an algorithmic point-of-view, blindly optimizing for accuracy-oriented metrics (or consumer relevance) may have adverse or unfavorable impacts on the fairness aspect of recommendations [29] or even other algorithmic biases may appear [30], e.g., in the employment recommendation context, certain genders or users from certain areas might be more likely to be recommended a job due to their behavioral differences and past information collected from users with the same characteristics [6].

[2] `https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai`, last accessed June 2021.

Regarding the fairness aspect, from a recommender systems perspective, where users are first-class citizens, there are reciprocal [31] and multiple stakeholders [32] which raise even more fairness issues. That means RS models should be designed to take into account the utility of recommendations relative to *(i)* target users or customers' preference, and *(ii)* the vendors and businesses – e.g., in terms of profitability [33]. Burke et al. [34] suggest that in multi-stakeholder recommender systems (MRS), the fairness of RS should be studied relative to (i) consumers (C-fairness), (ii) providers (P-fairness), and (iii) both (CP-fairness). From an algorithmic point-of-view, one can classify the prior literature on fairness-aware recommendation according to user-fairness or item-fairness (often directly related to businesses behind them), according to which sensitive attributes fairness is computed. For instance, for users, age, gender [35], and nationality are good examples of user-sensitive attributes, while for items, their category is commonly studied (e.g., gender of artists in music recommendation [36]).

Even though research on fairness has been a very active topic recently in several ML/IR/RS, there are few evaluation metrics designed that are capable of measuring fairness in RS. Tsintzou et al. [37] propose the metric "bias disparity" to quantify the relative deviation between the biases produced by RS and those inherently found in the data. Zhu et al. [38] propose the metric MAD (mean absolute deviation) to capture fairness in the average ratings between two groups. Yao et al. [39] define several unfairness quantities (non-parity, value, absolute, underestimation, overestimation, and balance unfairness) that can be applied to two groups of users and based on prediction errors. The main shortcoming of these evaluation metrics is that they are only valid for 2 groups and are focused on ratings or towards a rating prediction task, which has been displaced by the community because it does not correlate with the user satisfaction [17, 9].

To address these shortcomings, recently Deldjoo et al. [25] proposed a framework based on generalized cross-entropy (GCE) to evaluate the fairness of recommender systems for both users and items. Compared with those fairness evaluation metrics described above, GCE improves them in several dimensions: first, it can be used to define and measure fairness for both users and items; second, as it uses the probability distribution of recommendation outcomes over different (protected) groups, it inherently does not assume any predefined number of groups to define fairness upon and compared them in probabilistic sense; finally, it can incorporate different accuracy-related metrics to measure fairness upon, according to error metrics (e.g., RMSE, MAE), decision-support metrics (e.g., precision, recall), or ranking metrics (e.g., NDCG, MAP).

2.3. Understanding behavior of recommender systems

The definition of recommendation algorithms is, as presented before, at the core of the RS research. However, most of those proposals are based on intuitions or toy examples on how such methods should work for the general user. Moreover, these approaches are, usually, not completely deterministic and, in any case, with high levels of subjective behavior (in contrast to other domains like Machine Learning or Information Retrieval), due to the lack – by definition

– of a general notion of relevance. Hence, a general assessment of whether an algorithm is working as expected is usually achieved by an (objective) evaluation measurement, assuming that a well-performing algorithm (according to some definition of performance) is correlated with the lack of undesired behavior.

However, it is clear that these assumptions are not sufficient to actually know and understand how recommendation algorithms behave. With this goal in mind, researchers have analyzed specific components of the algorithms, such as similarity metrics and their effect on neighbor-based algorithms [40], or how the different hyper-parameters of the models affect the final performance [41]. In a research line closer to what we investigate in this work, the authors of [7] explicitly analyze the impact of data characteristics on the performance of classical recommendation algorithms. As discussed before, that paper was the original inspiration for this work, and in fact we follow the same experimental procedure (different random samples are extracted from the original datasets, see Section 4.2), although we extend the pool of data characteristics and use a more up-to-date definition of performance (i.e., a ranking-aware metric like nDCG instead of error-based metrics like Root Mean Squared Error). An extension of that work, but focused on the robustness of CF methods, is presented in [8]. That paper also uses a limited number of data characteristics and the discussion is restricted to accuracy as target performance, whereas in the current work we also analyze the impact on a fairness-aware metric.

In summary, the RS community is actively addressing and paying attention to the biases existing in items (novelty vs popularity), users (fairness), and other general recommendation aspects (such as temporal vs random evaluation, cold vs warm profiles, etc.), however, a clear understanding of the effect of these characteristics *inherently present* in the data is not available – this work aims to shed some light on this important issue.

3. Explanatory framework

In this section, we describe the foundations of the explanatory framework that aims to investigate the impact of data characteristics (DCs) on the performance of different families of collaborative filtering (CF) recommendation models, either measured as *(i)* accuracy or *(ii)* fairness.

The central question in the explanatory modeling research is the choice of explanatory variables (EVs) or data characteristics that can enable the researchers to apply an ever-widening range of models to data for explanatory analysis. Two main approaches exist for choosing the DCs, based on (i) confirmatory, or (ii) exploratory research [42, 7]. These two methods can be regarded as two complementary components of the same goal, that is to find relevant variables in the most efficient, reliable, and replicable manner. Their difference is that in confirmatory research, the potential impact of different variables are hypothesized a-priori, based on existing theories. This would in turn allow to focus on a small set of explanatory variables, from a larger set of alternatives. The confirmatory research approach is useful when researchers have a pretty good idea of the problem, or more precisely they have a theory (or theories) supported by facts.

The second approach is exploration-driven, which is used when there exists a lack of sufficient theory foundations. Exploratory research could likewise produce new hypotheses that could formally be evaluated later.

Similar to [7] our study belongs to the second category, where we design a general framework based on the explanatory modeling paradigm to study the impact of data characteristics on RS performance, measured in term of accuracy and fairness metrics. By treating the hypothesized impactful parameters, which vary in terms of the information they capture, as DCs, we can allow the explanatory framework to explain various DC factors in RS.

3.1. Theoretical modeling of the explanation framework

Given a dataset d, a recommendation model g (e.g., neighborhood-based or latent-factor CF model), the goal is to test the hypothesis whether some EVs — capturing DC information — can explain the variations on the dependent variable (DV) — related to RS performance. A regression model is used to model the relationship according to

$$y^g = \epsilon + \theta_0 + \sum_{c=1}^{C} \theta_c x_c \tag{2}$$

in which C is the number of DCs, θ_c is the regression coefficient of the c-th explanatory EV (cf. Section 3.2), $x_c \in \mathbb{R}$ represents the value of the c-th EV for the i-th training example, and $y^g \in \mathbb{R}$ is the measurement corresponding to a training sample according to recommendation model g, the measured DV (cf. Section 3.3).

3.2. Explanatory variables

The explanatory variables (EVs) considered in this work describe the DCs from a wide range of perspectives. The definition of these variables have been obtained by reviewing the most impactful studied parameters in the literature of RS over the last two decades – e.g., consider [43, 44, 45, 46, 47, 48, 49].

The EVs describe different aspects of data and can be categorized according to the following groups:

- Based on the structure of the URM
- Based on the rating frequency of the URM
- Based on item properties (popularity, long-tailness) of user profiles
- Based on the distribution of rating values

We formally describe the main features measured in each category in the next sections. In what follows, we assume we are dealing with a given URM, with a number of real users $|\mathcal{U}|$, real items $|\mathcal{I}|$, and ratings $|\mathcal{R}|$.

3.2.1. EVs based on the structure of the URM:

The EVs in this section measure properties that are directly impacted by the structure of the URM, specified by its dimension as well as number of known entries (ratings).

Definition 1 (*SpaceSize*)**.** *Given a URM, SpaceSize is defined as:*

$$x_1 = SpaceSize(URM) = |\mathcal{U}| \cdot |\mathcal{I}| \tag{3}$$

□

This EV directly measures the capacity of the URM without considering its entries. It is a simple, but useful, metric that allows to compare different datasets in terms of the maximum number of preferences that can be collected from users.

Definition 2 (*Shape*)**.** *Given a URM, we define Shape as follows:*

$$x_2 = Shape(URM) = \frac{|\mathcal{U}|}{|\mathcal{I}|}. \tag{4}$$

□

Note that when $Shape(URM) \ll 1$ then $|\mathcal{U}| \ll |\mathcal{I}|$, i.e., there are more candidate neighbor users than candidate neighbor items. On other hand, when $Shape(URM) \gg 1$ then $|\mathcal{U}| \gg |\mathcal{I}|$, i.e., there are more candidate neighbor items than candidate neighbor users. For instance, it is natural to foresee that this situation might work in the advantage of user-based CF compared with item-based CF or vice-versa, depending on whether the URM has more number of candidate users or items [50].

Definition 3 (*Density*)**.** *Given a URM, we define Density as follows:*

$$x_3 = Density(URM) = \frac{|\mathcal{R}|}{|\mathcal{U}| \times |\mathcal{I}|} \tag{5}$$

□

Data density is inversely related to data sparsity via Density $= 1 -$ Sparsity. Sparse information is a well-known phenomena in RS [45], it refers to settings where the fraction of known interactions is significantly lower than the potential number of possible ones, making it too difficult for CF recommendation models to make correct predictions. It is very common in the area to find experimental settings where this DC has been explicitly analyzed, such as [43] and [44].

Definition 4 (Rp_u, Rp_i)**.** *Given a URM, rating per user (Rp_u) and per item (Rp_i) are defined as follows:*

$$x_4 = Rp_u(URM) = \frac{|\mathcal{R}|}{|\mathcal{U}|} \tag{6}$$

$$x_5 = Rp_i(URM) = \frac{|\mathcal{R}|}{|\mathcal{I}|} \tag{7}$$

□

Note that Rp_u and Rp_i are two of the most widely used DCs in the literature, since they are often reported as statistics of tested URMs, side-by-side density. We provide intuition behind the reasons why these DCs are of interest for RS. For instance, Rp_u can directly impact the performance of any CF method since, at the end, CF models provide a personalized recommendation to each user based on their interaction (rating) profile. Also, quite often, research works prefer to apply a different threshold on the minimum number of ratings in the user profile to consider that user for evaluation [46, 47]. For these reasons, we deem these newly introduced DCs of high interest for this study.

Moreover, it should be noted another area worth of investigation for the last three features (*Density*, Rp_u, and Rp_i): simulating cold-start situations such as *sparse preferences*, *cold users*, and *cold items* [45], or even the transition from a cold-start to warm-start setting [51]; dealing with these issues is a quite common task in the community of RS and an active area of research in the field.

3.2.2. EVs based on the rating frequency of the URM:

Another important characteristic of a URM is the rating frequency distribution. The idea is that in many real applications, a small number of items receive a large number of ratings (short head or popular items), while a large number receive low or few feedbacks (long tail), causing the rating distribution to be skewed.[3] It turns out that the commercial profit from recommending long-tail items is more significant than short-head items [52]. However, these long-tail items have less chance to be recommended since they have less historical feedback [50]. We examine this characteristic because it could help on understanding how biased towards popular items the algorithms could be.

Definition 5 ($Gini_i$, $Gini_u$)**.** *Given a URM, let $|\mathcal{R}_i|$ and $|\mathcal{R}_u|$ be the number of ratings associated with item i and user u; then $Gini_i$ and $Gini_u$ are defined respectively in the following manner:*

$$x_6 = Gini_i(URM) = 1 - 2\sum_{i=1}^{|\mathcal{I}|} \frac{|\mathcal{I}| + 1 - i}{|\mathcal{I}| + 1} \times \frac{|\mathcal{R}_i|}{|\mathcal{R}|} \tag{8}$$

$$x_7 = Gini_u(URM) = 1 - 2\sum_{u=1}^{|\mathcal{U}|} \frac{|\mathcal{U}| + 1 - u}{|\mathcal{U}| + 1} \times \frac{|\mathcal{R}_u|}{|\mathcal{R}|} \tag{9}$$

□

More specifically, the Gini coefficient measures the concentration of items, or users, ratings to capture the rating frequency distribution. A uniform popularity distribution (e.g., all users or items give the same number of ratings) is represented with the value of the Gini coefficients to 0, while the total inequality (e.g., only one user or item has given all ratings) is represented with a value of 1. Note

[3] It should be noted that, although this discussion is centered around ratings, a similar argument can be made based on other types of interactions, such as clicks or listenings.

that Equations 8 and 9 assume items and users are sorted according to $\mathcal{R}_i$ and $\mathcal{R}_u$ respectively.

3.2.3. EVs based on item properties of user profiles:

The EVs defined in this section have never been investigated – to the best of our knowledge – in a similar explanatory framework before. They are however, widely used in the evaluation of recent RS, as they are related to the inherent biases that can be found in the data exploited by a recommender system [53, 54, 49, 48].

Definition 6 (Popularity Bias)**.** *The popularity profile of the user is measured as the average popularity of items consumed by a user. Once averaged over users, the computed score provides an evaluation of the popularity bias [48] of a given dataset. A general formulation over popularity bias assessment is defined as:*

$$x_{8:11} = f\left(\left\{\frac{\sum_{i\in\mathcal{R}_u}\phi(i)}{|\mathcal{R}_u|}\right\}_u\right) \tag{10}$$

where $\phi(i)$ is the popularity score of item i defined as the number of users who consumed item i over the entire number of users, and $|\mathcal{R}_u|$ is the size of the rating profile of user u, as in the previous definition. f is an aggregation operator over users, to capture inter-user differences in popularity profiles of users. They include average popularity bias (x_8, APB), standard deviation of popularity bias scores (x_9, StPB), skewness popularity bias (x_10, SkPB), and kurtosis popularity bias (x_11, KuPB).

□

Definition 7 (Long tail items)**.** *The goal of this EV is to understand how many less-known (unpopular) items are consumed by and exist in the profile of each user. It is defined as follows:*

$$x_{12:15} = h\left(\left\{\frac{|i, i\in(\mathcal{R}_u\cap\Gamma)|}{|\mathcal{R}_u|}\right\}_u\right) \tag{11}$$

where $\mathcal{R}_u$ is the rating profile of user u and Γ represents long-tail items, and it is determined after the dataset is splitted into two categories (short head v.s. long-tail) in such a way that long-tail items correspond to 20% of ratings, while short-head items provide the remaining 80%. h is an statistical aggregating operator applied over this user distribution. For instance, once averaged over users, the computed EV would correspond to average percentage of long-tail items (x_{12}, $LTail_{avg}$) [53], and would tell us the fraction of items in the entire users' profiles that belong to the long-tail set. We further accommodate other statistical aggregation operators applied over users, namely standard deviation of long-tail items (x_{13}, $LTail_{std}$), skewness of long-tail items (x_{14}, $LTail_{skew}$), and kurtosis of long-tail items (x_{15}, $LTail_{ku}$).

□

3.2.4. EVs based on the distribution of rating values:

Rating values, when available, provide a different, alternative viewpoint of the user behavior with the system, in comparison against the rating frequency or item properties. On the one hand, some systems – either because of their interface or the nature of the items – might be biased towards more spread or extreme rating values;[4] on the other hand, recommendation algorithms might not perform equally on the entire rating scale [43]. Because of these reasons, we consider this dimension might be valuable to better understand how the data impacts the performance of RSs.

Definition 8 (Distribution of rating values)**.** *The goal of this EV is to measure the statistical distribution of rating values, which is a different measurement to the ones introduced in previous sections, based on the rating entries. The distribution of rating values can be described based on*

$$x_{16:19} = m\left(\{r_{u,i}\}_r\right) \tag{12}$$

where $r_{u,i}$ is the rating given by user u to item i and m is an statistical aggregation operator over known rating entries, as in previous definitions. For instance, we explore the possible influence of its standard deviation (x_{17}, Std_{rating}), since its negative impact on the rating prediction task measured by RMSE was previously reported in [7]. Similar to previous EVs, we compute the average of this distribution (x_{16}, $Mean_{rating}$), its skewness (x_{18}, Sk_{rating}), and kurtosis (x_{19}, Ku_{rating}) aggregation operators on the rating values.

□

3.3. Dependent variables

The dependent variables (DV) represent the performance of the recommender system; in this work we propose to measure performance in two different, complementary ways: accuracy and fairness.

Definition 9 (Recommendation accuracy)**.** *Normalized discounted cumulative gain is a highly popular rank-aware metric in RS, that measures the utility of an item based on its position in the result list. However, as recommendation results may vary in length depending on the user, to allow comparisons between users, the ideal cumulative gain computed over the entire test set of a user is used to normalize this metric. Normalized discounted cumulative gain, or nDCG, is defined as*

$$y_1 = nDCG@N = \sum_{u \in \mathcal{U}} \frac{1}{IDCG_u@N} \sum_{k=1}^{N} \frac{2^{r_{uk}} - 1}{\log_2(1+k)} \tag{13}$$

[4] A famous example was the redesign of the YouTube interface, explained in `https://www.cnet.com/news/youtubes-big-redesign-goes-live-to-everyone/` (retrieved in December 2020).

where k is the position of an item in the recommendation list and $IDCG@N$ measures the score obtained by an ideal ranking of the recommendation list Rec_u^N that contains solely relevant items, up to a cutoff N.

□

Definition 10 (Recommendation fairness)**.** *For the purpose of fairness evaluation, we use MAD-ranking [6], which measures differences between the groups, interpreted as unfairness. MAD is then defined formally by*

$$y_2 = MAD(i,j) = \left| rank^{(i)} - rank^{(j)} \right| \tag{14}$$

where $rank^{(i)}$ denotes the average ranking performance restricted to those users in group i, and $rank^{(j)}$ captures the same metric score for group j. Larger values for MAD imply differentiation between groups interpreted as unfairness. To make the results comparable with recommendation accuracy, we used nDCG@N as ranking metric when calculating MAD.

□

4. Experimental setting

In this section, we present in detail the experimental settings adopted to validate the research questions introduced in the beginning of the paper, whose final goal is a better understanding of how recommendation algorithms behave as the underlying data characteristics (on which they are trained) change.

We first show the datasets used (Section 4.1) and the sampling procedure that generates several instances for training (Section 4.2), then, the recommendation algorithms we compared (Section 4.3) and the parameters and other settings considered in the experiments (Section 4.4) that will be presented and discussed in the following section.

4.1. Datasets

The fairness dimension of RS is typically evaluated based on the definition of a number of sensitive attributes associated with users and/or items. In this work, we have focused on user fairness, defined according to *user gender* and *user age*. Nonetheless, the evaluation setup can be easily extended to incorporate various other user and item (sensitive) attributes. To choose the right dataset, we needed to use the ones that *(i)* both contain the intended attributes, *(ii)* they contain continuous preference scores (ratings). For these reasons, we used two different versions of the MovieLens[5] (ML) dataset [55], namely `ML-100K` and `ML-1M` where both datasets contain user gender information, and the `BookCrossing`

[5] Available at `https://grouplens.org/datasets/movielens/`

Table 1: Characteristics of the user-rating matrix associated with `ML-100K` and `ML-1M`: $|\mathcal{U}|$ — number of users, $|\mathcal{I}|$ — number of items, $|\mathcal{R}|$ — number of ratings. $|\mathcal{R}|$ represents the *density* of that dataset. The last column (USA ratio) represent dataset composition in terms of user-sensitive attributes, utilized for the fairness study. These attributes include *gender* (for `ML-100K` and `ML-1M`), and *age* (for `BookCrossing`), where for the latter we considered two age groups: Children & Young (0-24 years old), Adult & Senior (25-99 years old).

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{R}\rvert$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{U}\rvert}$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{I}\rvert}$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{I}\rvert\times\lvert\mathcal{U}\rvert}$	USA ratio
`ML-100K`	943	1682	100,000	106.04	59.45	0.0630	(71.05%, 28.95%)
`ML-1M`	6,040	3,667	1,000,209	165.59	272.75	0.0451	(71.61%, 28.29%)
`BookCrossing`	105,283	340,556	1,149,780	10.92	3.37	32e-5	(26.18%, 73.81%)
`BookCrossing` (sampled)	53,423	157,914	344,934	6.46	2.18	408e-5	(13.81%, 86.19%)

dataset which includes ratings to book items and, as user metadata, age categories and locations.[6]

The `ML-100K` dataset contains 100K movie ratings given by 1K unique users to 1.7K unique items (movies). The `ML-1M` dataset includes 1M movie ratings given by 6K users to 4K items. Each item is rated on a 1-5 Likert scale in both datasets. The `BookCrossing` dataset, on the other hand, contains 278K users (anonymized but with demographic information) providing 1.1M ratings about 340K books. Note that one interesting aspect about these datasets is their contrasting number of users and items, where $|\mathcal{U}| < |\mathcal{I}|$ in `ML-100K` and `BookCrossing`, while $|\mathcal{U}| > |\mathcal{I}|$ in ML-1M. These differences in original URM characteristics for MovieLens datasets would encourage samples generated having significant diverse DCs, effectively improving the results/insights obtained from the explanatory research in this study, even though these datasets come from the same domain, movies.

As for the `BookCrossing` dataset, we noticed it has a number of items which is dramatically larger than `ML-1M` – namely about 110 times higher. This created huge computational issues for typical recommendation models, as there were simply too many possibilities to form the candidate items. To address this shortcoming, we randomly took 30% of the interactions in `BookCrossing` to serve as the original matrix. Then, as we shall explain in the next section, we created sub-samples out of it similarly as with `ML-100K` and `ML-1M`.

Table 1 summarizes the global characteristics of these datasets.

4.2. Sampling procedure

Based on the regression-based explanatory model formalized by Eq. 2, the goal is to compute the regression model coefficients, based on DCs generated

[6] We want to emphasize the difficulty on finding datasets with enough personal information of good quality to perform the described experiments. Among the well-known datasets used in the community [56], Yelp, Epinions, and Amazon datasets do not include user attributes, while Last.fm does not contain explicit ratings. In fact, the datasets included do not share the same sensitive attributes regarding users: whereas MovieLens includes the user gender, `BookCrossing` provides age and location.

Algorithm 1 Sample generation procedure

Input: URM
$n_u \leftarrow$ number of users of the URM
$n_i \leftarrow$ number of users of the URM
$n_r \leftarrow$ number of ratings of the URM
$\tau_u \leftarrow$ constraint on average number of ratings for users
$\tau_i \leftarrow$ constraint on maximum number of items
Results: N sub-datasets (urm_n)
$n \leftarrow 1$
while $n \leq N$ **do**
    Random shuffle the row of the URM
    $n_u \leftarrow rnd([100, n_u])$
    $n_i \leftarrow rnd([100, n_i])$
    $urm_n \leftarrow$ Selection of n_u, n_i from URM
    if $\frac{n_r}{n_u} < \tau_u$ or $n_i > \tau_i$ **then**
        $n \leftarrow n + 1$

from a given dataset (URM). In order to obtain reliable and replicable regression solutions, many training samples of type (x, y) should be generated. It is desired that the training samples are generated from a wide range of perspectives, e.g., via different scales and sizes.

The sampling procedure is specified in Algorithm 1. To this end, we adopt the sampling generation strategy presented in [7, 8], where for a given URM, we generated $n = 600$ different samples. These samples (sub-datasets) are denoted with urm_n in Algorithm 1 and represent smaller URMs with a wide diverse range of DCs, as we outlined in Section 3.2.2, for instance with different sizes, levels of sparsities, and so forth. When creating these samples, we impose a number of constraints to ensure that the generated samples are useful to build a model based upon, they include: (i) each sample should have minimum 100 users and items, (ii) the average number of ratings in the user profiles should be over a threshold (e.g., we set $\tau_u = 10$ in the case of `ML-100K` and `ML-1M`), and (iii) the number of items should not go beyond a maximum value as it may cause computational issues ($\tau_i = 70,000$ for `BookCrossing`).

We want to highlight that cold-start scenarios are not considered in this work (and left as a potential research avenue that might be addressed in the future) for the sake of clarity and conciseness. As we have described, to obtain reliable recommendations we impose constraints on the number of interactions each user has when creating these samples. This is because cold-start situations should be evaluated carefully, as done in the area [57, 45], and we believe they deserve a proper analysis on different profile sizes to explore whether the same data characteristics are as explainable in standard scenarios as in cold-start ones.

4.3. Compared CF recommendation models

In this work we study the impact of data characteristics for various collaborative filtering (CF) recommendation models. They can be classified into two main classes of *(i)* neighborhood-based model (a.k.a. memory-based), and *(ii)* latent-factor models.

4.3.1. Neighborhood-based models

For the choice of neighborhood-based CF models, we relied on two popular models: `UserKNN` and `ItemKNN`, together with several variations of these models that by and large differ from each other based on the core similarity metric, or the weighting/amplification of ratings when calculating similarities.

- `UserKNN-Cosine` [58]: A user-based neighborhood-based method that computes user-user similarities based on the cosine similarity of their interaction (here, rating) profiles. The closest neighbor users to a given target user are chosen according to the computed similarities.
- `UserKNN-Pearson` [59]: It uses Pearson correlation coefficients as similarity function to find user-user similarities.
- `UserKNN-Amplified`: This method introduces a weight factor whose role is to amplify the importance of more similar users relative to less similar ones. The effectiveness of amplification on improving the accuracy of recommendation has been shown on other CF tasks, such as playlist recommendation [60].
- `UserKNN-IDF` [61]: A variant of `UserKNN` that weights ratings with the inverse document (item) frequency (IDF). In this way, it allows to account for the popularity (in fact, for the novelty) of the items.
- `UserKNN-BM25` [61]: Another variant of `UserKNN` that weights ratings via BM25 algorithm. This algorithm is widely used in text retrieval [16] and has demonstrated good modeling capabilities in several tasks, from tag to item recommendation [62].

- `ItemKNN-Cosine` [63, 64]: An item-based implementation of the K-nearest neighbor algorithm, that finds nearest item neighbors based on the cosine function computed on item ratings.
- `ItemKNN-Pearson` [63, 64]: It uses Pearson correlation coefficient similarity function to compute item-item similarities.
- `ItemKNN-Adjusted` [64]: It uses a variation of the Cosine similarity, where the user's average rating is considered to *adjust* the similarity computation and personalize it to each particular user.

All these methods are instantiations of the following formulations, for instance, by considering specific similarity functions:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in \mathcal{U}_i^k(u)} \text{sim}(u,v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in \mathcal{U}_i^k(u)} \text{sim}(u,v)} \tag{15}$$

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in \mathcal{I}_u^k(i)} \text{sim}(i,j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in \mathcal{I}_u^k(i)} \text{sim}(i,j)} \tag{16}$$

where $\text{sim}(\cdot,\cdot)$ is a similarity function between two elements, and $\mathcal{U}_i^k(u)$ and $\mathcal{I}_u^k(i)$ are the neighborhoods of a given user or item, that is, those k users or items closest to that user according to the similarity function.

4.3.2. Latent-factor based models

We also considered a wide range of latent factors models, used in the past and current research works of RS achieving very good performance in rating and ranking tasks [65, 66].

- `MF` [65]: A classical Matrix Factorization approach, in this case, the user and item factor are learned through Stochastic Gradient Descent, even though other techniques are available in the area [67]. The predicted rating in MF is computed as $\hat{r}_{ui} = \mathbf{q}_i^T \mathbf{p}_u$, where $\mathbf{q}_i \in \mathbb{R}^H$ and $\mathbf{p}_u \in \mathbb{R}^H$ are the item and user latent vectors learned by the model.

- `SVD` [65]: An extension of the previous MF approach described where user and item biases are considered when learning the user and item factors. The predictor in SVD has the form $\hat{r}_{ui} = b_{ui} + \mathbf{q}_i^T \mathbf{p}_u$, where $b_{ui} = \mu + b_u + b_i$, and μ, b_u, b_i represent the overall average rating and the observed biases of user u and item i.

- `BPR-MF` [68, 65]: BPR is the state-of-the-art method for personalized ranking, particularly on implicit feedback datasets. BPR-MF uses MF as the predictor and a pairwise ranking loss whose goal is to optimize personalized ranking obtained by the MF model. BPR is widely known for achieving competitive performance in item recommendation tasks.

- `PMF` [69]: A Probabilistic Matrix Factorization algorithm, where the matrix is factorized based on a probabilistic lineal model with Gaussian noise and a Maximum A Posteriori method.

- `NMF` [70]: A Non-negative Matrix Factorization method based on adding a constraint on classical MF techniques, taking into account that ratings in recommendation are always positive integers; in particular, it enforces user and item factors are kept positive.

It should be emphasized that in this paper our goal is not to obtain the best performance with any of these methods, but to understand under which situations any of them may improve their performance, or more precisely their performance changes. That is why we selected a wide range of methods but left other representative (and more recent) approaches out of the study. Thus, we aim to include other families as future work.

4.4. Evaluation metrics and settings

For each considered recommendation model, we ran them at their default hyper-parameter values according to their implementation in the Cornac recommender framework [71]. The results of the recommendation were generated based on a hold-out setting (80%-20% training-test split).

As for the choice of evaluation metric, we chose MAP to compare the results on the datasets chosen in this work (`ML-100K`, `ML-1M`, `BookCrossing`). At the same time, we reported the result on NDCG@100, though in the appendix (and only for `ML-100K` and `ML-1M`), given that MAP produced more reliable recommendations to use the explanatory study upon. We discuss this aspect better in Section 5.1.

5. Results and Discussion

In this section, we present the results of the large-scale performed experiments, which involve 1,800 generated samples (600 for each of three datasets) and running 13 CF recommendation models. Hence, a total of 23,400 recommendation simulations were performed to report the results shown in the current section.

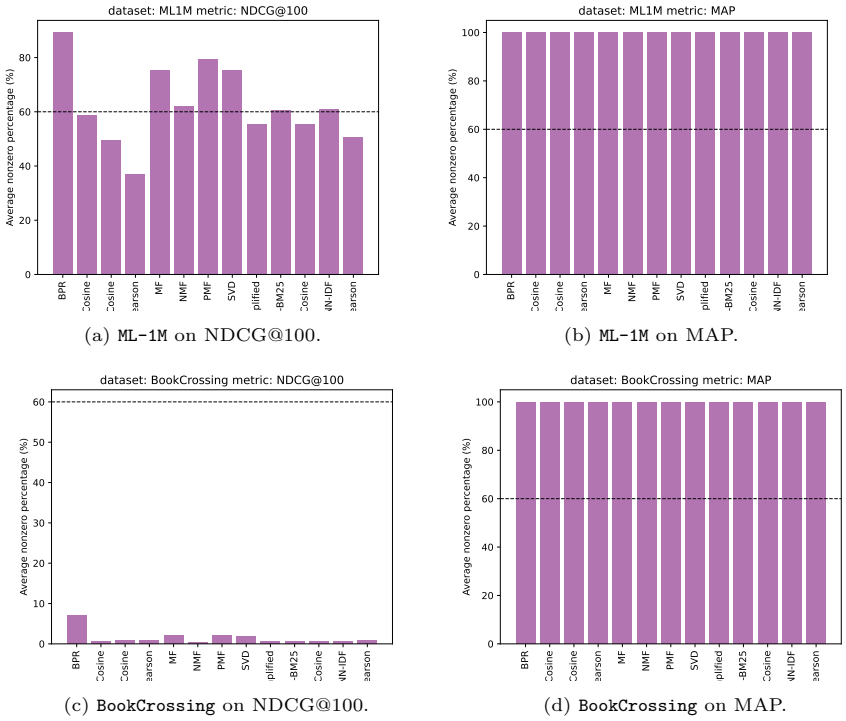
5.1. Quality control and sanity check

In the present work at hand, we deal with two large pools, *(i)* a pool of CF models, and *(ii)* a pool of DCs. These two sets need to undergo a quality or sanity check before applying the regression model on them. The main motivation is to ensure the reliability and precision of regression modeling, which would serve as the main tool for the explanatory study. We made the following observations in this regard.

First, we noticed that some of the neighborhood methods produce (in a user-basis evaluation) several zero values for a considerable fraction of users contained in the training sample (the smaller urm). The reason for this phenomena can be directly linked with their lower total number of ratings. The data scarcity can in fact harm the quality of some specific recommendation models more than the others, the reason we could not allow these methods to enter the final pool of methods considered for the explanatory study. The motivation is as follows: even though we wish to evaluate recommendation performance based on DCs, this would be meaningless if the recommendations do not achieve a minimum quality level. In other terms, explaining recommendation performance that achieves very poor performance is of no value.

We show in Figure 1, the average number of user-based evaluations not equal to zero across all training samples (small urms). In essence, what each bar in these plots represents is, for each recommendation model and over $N = 600$ samples, on average what is the percentage of users, which do NOT have a zero user-based evaluation. Obviously, the higher this value, the more reliable the recommendation result is from an explanatory study point of view. One observation is that the results dramatically change based on evaluation metrics: while MAP produces perfect user-based evaluation with almost all recommenders producing non zero user-based evaluation, NDCG produces worse performance. Thus, for NDCG@100, we made sure each recommendation model receives on average at least 60% non-zero user-based evaluations. Because of this, in the final pool, `ItemKNN-Cosine`, `ItemKNN-Pearson`, `UserKNN-Pearson` were excluded.[7] To have a consistent set of models for both metrics and given the space limitation, the final pool of CF models thus consists of 10 models: `UserKNN-Amplified`, `UserKNN-BM25`, `UserKNN-Cosine`, `UserKNN-IDF`, `ItemKNN-Adjusted`, `BPR`, `MF`, `SVD`, `PMF`, and `NMF`.

[7] For `ML-100K`, all methods except `UserKNN-Pearson` achieved the desired performance. For `ML-1M`, 6 methods fall below the threshold, in which `ItemKNN-Adjusted`, `UserKNN-Amplified`, `UserKNN-Cosine` had a narrow gap with the desired threshold. Thus, we kept them in the final pool of CF models, to have comparable methods in both considered datasets.



(a) `ML-1M` on NDCG@100. (b) `ML-1M` on MAP.

(c) `BookCrossing` on NDCG@100. (d) `BookCrossing` on MAP.

Figure 1: Analysis of accuracy performance (by measuring the percentage of users with non-zero performance) of the recommendation algorithms tested in `ML-1M` vs `BookCrossing`.

As an additional quality check, we also control for the co-linearity of DCs using variance inflation factor (VIF). This measures the impact of multi-colinearity among the DCs in a regression model on the precision of the estimation. VIF produces a score for each EV that indicates the degree to which multi-colinearity amongst the DCs degrades the precision of an estimate. Unfortunately, there is no well-defined critical value on what can be considered as a large/bad VIF, although some research works suggest $VIF = 10$ can indicate a problem [72, 73]. To remain far from this threshold, in this work, we chose the threshold of $VIF = 5$ and removed a handful of variables. We have provided the detailed results of the filtering process in Table 2 and Figure 2.

One of the first points we observe in top part of Figure 2 is the high correlation degree between some sets of variables, either positive (popularity kurtosis vs popularity skewness) or negative (rating skewness vs rating kurtosis). This is a strong indication that those DCs are not independent to each other, an aspect that violates the assumption of the statistical regression model. To remove undesired variables we performed two steps:

- **Feature normalization**: for which we use min-max normalization;
- **Feature removal**: which involved discarding highly correlated features

Table 2: Sanity check for choosing the most suitable set of features that are not co-linear. $VIF > 10$ indicates high degree of colinearity between explanatory variables that can degrade the precision of the estimation in the regression model. We ensured after sanity check all $VIF < 5$. See section 5.1 for further information.

feat.	dataset	ML-100K		BookCrossing	
	DC/Sanity	VIF before	VIF after	VIF before	VIF after
f_1	$SpaceSize$	38.6	3.0	30.1	1.3
f_2	$Shape$	16.7	2.4	9799.0	dropped
f_3	$Density$	1621.7	4.3	3329.2	dropped
f_4	Rp_u	85.6	3.9	6122.2	1.7
f_5	Rp_i	77.8	dropped	24847.6	dropped
f_6	$Gini_u$	1.2	1.1	12198.4	dropped
f_7	$Gini_i$	1895.6	4.5	925.6	dropped
f_8	Pop_{avg}	5313.3	2.5	5336.8	1.4
f_9	Pop_{std}	1220.1	dropped	533.6	dropped
f_{10}	Pop_{skew}	173.8	2.6	3076.6	1.1
f_{11}	Pop_{ku}	25.0	dropped	1076.8	dropped
f_{12}	$LTail_{avg}$	2390.8	1.2	78738.7	1.3
f_{13}	$LTail_{std}$	1507.1	dropped	291407.0	dropped
f_{14}	$LTail_{skew}$	1415.0	2.6	789034.6	1.6
f_{15}	$LTail_{ku}$	145.7	dropped	70667.8	dropped
f_{16}	$Mean_{rating}$	29387.1	dropped	355711.4	dropped
f_{17}	Std_{rating}	50698.7	1.2	400634.8	dropped
f_{18}	Sk_{rating}	4373.8	dropped	125043.0	dropped
f_{19}	Ku_{rating}	1246.7	dropped	22471.4	dropped

according to their pairwise correlation score.

We show in Table 2 the result of VIF before and after the data sanity step outlined above, whereas Figure 2 shows pairwise DCs correlation values. We notice that using different feature aggregators to statistically aggregate user-based DCs over users, tend to produce more correlated features. For instance, from the popularity bias category, standard deviation of popularity bias scores (f_9) and kurtosis popularity bias (f_{11}) were discarded. Similarly standard deviation of long-tail items (f_{13}) and kurtosis of long-tail items (f_{15}) were excluded. On the distribution of rating values, skewness of rating values (f_{18}) and their kurtosis (f_{19}) were also discarded. Note that these correlations may change depending on the datasets; as shown in Table 2, features such as shape (f_2) or density (f_3) are removed for BookCrossing. Finally, as reported in [72, 73], since VIF is sensitive to mean centralization, we mean-centered all the variables and obtained the final pool of DCs and corresponding VIF values, that can be found in Table 2 (second column in each dataset) and bottom part of Figure 2. Note that the same pattern of outcomes was obtained for the other MovieLens dataset.

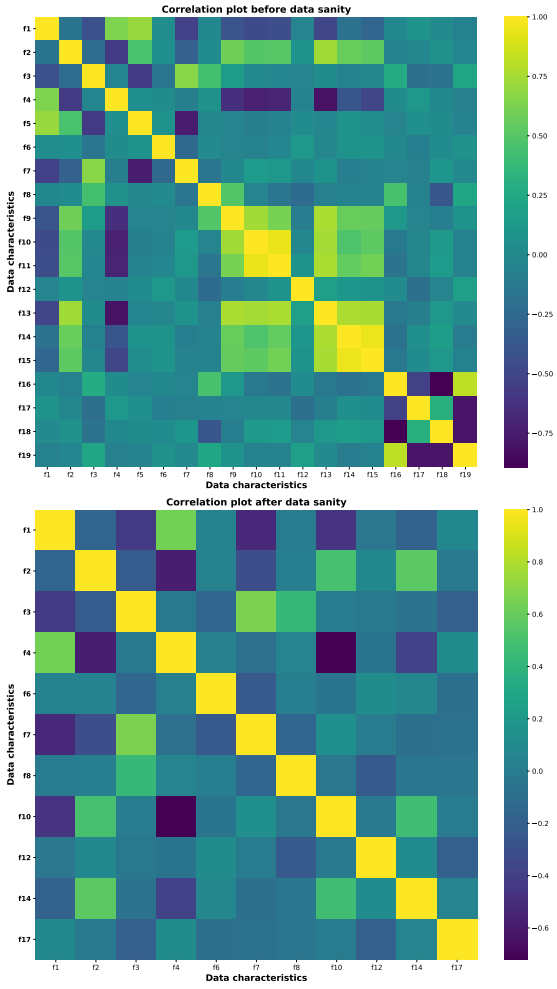


Figure 2: Correlation plot of the data characteristics in `ML-1M` before (top) and after (bottom) data sanity check. Note that higher correlation values are harmful for the regression model estimation. Note also that the square clusters on the top figure occur when aggregating a certain feature over users via different statistical aggregation functions (e.g., mean, std, skewness, kurtosis).

Table 3: Regression results for the within dataset analysis (target metric: MAP).

	Target MAP	**Memory-based**					**Model-based**				
		UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
ML-100K	R^2 (adj.R)	0.825 (0.822)	0.826 (0.823)	0.824 (0.821)	0.826 (0.822)	0.881 (0.879)	0.868 (0.865)	0.727 (0.722)	0.725 (0.719)	0.75 (0.745)	0.827 (0.824)
	Constant	0.02322***	0.02364***	0.02503***	0.02391***	0.02215***	0.09795***	0.0522***	0.04895***	0.0395***	0.03873***
	SpaceSize	0.00149***	0.00146***	0.00152***	0.00146***	0.00213***	-0.00059	0.00396***	0.00332***	0.00175***	0.00042
	Shape	0.00352***	0.00357***	0.00352***	0.00356***	0.00404***	0.0088***	0.00537***	0.00533***	0.00226***	0.00483***
	Density	0.00368***	0.0037***	0.00371***	0.00371***	0.0013***	-0.00236***	0.00625***	0.00515***	0.00327***	0.0032***
	Rp_u	-0.00433***	-0.00433***	-0.00437***	-0.00434***	-0.00339***	-0.0126***	-0.00968***	-0.00878***	-0.00593***	-0.00634***
	$Gini_u$	-0.00081***	-0.00081***	-0.00081***	-0.00081***	-0.00066***	-0.00138***	-0.00157***	-0.00163***	-0.00085***	-0.00062*
	$Gini_i$	0.0001	9e-05	9e-05	8e-05	3e-05	0.00384***	0.00024	0.00024	0.00022	0.00023
	Pop_{avg}	-0.00012	-0.00012	-0.00013	-0.00013	0.00104***	0.00802***	-0.00037	-0.00021	0.00056	0.00057
	Pop_{skew}	0.00114***	0.00116***	0.00115***	0.00116***	0.00038***	0.0059***	0.00272***	0.00255***	0.00096***	0.00162***
	$LTail_{avg}$	0.00025	0.00024	0.00024	0.00024	0.00036***	0.00176***	-1e-05	0.00021	0.00051*	5e-05
	$LTail_{skew}$	0.00198***	0.00198***	0.00199***	0.00199***	0.00087***	0.00444***	0.00343***	0.00307***	0.00119***	0.00209***
	Std_{rating}	0.00017	0.00018	0.00019	0.00018	-0.0	-0.00021	0.0008	0.00081	0.00011	0.00059*
	Accuracy	0.025 ± 0.009	0.025 ± 0.0091	0.025 ± 0.0091	0.025 ± 0.0091	0.0247 ± 0.0071	0.102 ± 0.0281	0.0535 ± 0.0179	0.0524 ± 0.017	0.0415 ± 0.0095	0.0398 ± 0.013
ML-1M	R^2 (adj.R)	0.863 (0.861)	0.845 (0.842)	0.864 (0.861)	0.861 (0.858)	0.866 (0.863)	0.931 (0.93)	0.709 (0.704)	0.746 (0.742)	0.849 (0.847)	0.878 (0.875)
	Constant	0.01625***	0.01896***	0.01623***	0.01934***	0.02073***	0.08667***	0.04403***	0.04454***	0.05167***	0.02261***
	SpaceSize	0.0007***	0.00051*	0.0007***	0.00056*	0.00234***	0.00087	0.00459***	0.0031***	0.00255***	0.00125***
	Shape	0.00258***	0.00301***	0.00258***	0.00448***	0.00198***	0.00644***	0.00517***	0.00515***	0.00278***	0.00363***
	Density	0.00063***	0.00041	0.00063***	0.00051	0.00033*	-0.00503***	0.00126	0.00073	0.00037	0.00137***
	Rp_u	-0.00208***	-0.00288***	-0.00208***	-0.00258***	-0.00251***	-0.01121***	-0.00682***	-0.00622***	-0.00855***	-0.00399***
	$Gini_u$	-0.00033***	-0.00019	-0.00033***	-0.00027	-0.00023***	-0.00062*	-0.00051	-0.00056	-0.00073***	-0.0005***
	$Gini_i$	0.00075***	0.00062*	0.00074***	0.00035	-0.00078***	0.00692***	0.00129	0.00037	0.00262***	0.00159***
	Pop_{avg}	0.0005***	0.00054*	0.0005***	0.00053*	0.00067***	0.00937***	0.00249***	0.00239***	0.0034***	0.00113***
	Pop_{skew}	0.0008***	0.00147***	0.00079***	0.00146***	0.00032*	0.006***	0.00145***	0.00139***	0.00213***	0.0016***
	$LTail_{avg}$	-0.00038***	-0.00057***	-0.00038***	-0.00048***	-0.00033***	-0.00093***	-0.00077*	-0.001***	-0.00139***	-0.00046***
	$LTail_{skew}$	0.00168***	0.0026***	0.00168***	0.00188***	0.00154***	0.00745***	0.0031***	0.00292***	0.00432***	0.00307***
	Std_{rating}	-0.00024*	-0.00018	-0.00024*	-8e-05	-9e-05	-0.00082***	0.00069*	0.00068*	-0.00036	-0.00019
	Accuracy	0.0162 ± 0.0059	0.019 ± 0.0085	0.0162 ± 0.0059	0.0193 ± 0.0089	0.0207 ± 0.0053	0.0867 ± 0.0266	0.044 ± 0.0141	0.0445 ± 0.0137	0.0517 ± 0.0149	0.0226 ± 0.0101
BookCrossing	R^2 (adj.R)	0.337 (0.33)	0.337 (0.33)	0.337 (0.33)	0.337 (0.33)	0.634 (0.63)	0.143 (0.134)	0.104 (0.095)	0.111 (0.102)	0.049 (0.04)	0.717 (0.715)
	Constant	0.00032***	0.00032***	0.00032***	0.00032***	0.00022***	0.00575***	0.00148***	0.00135***	0.00146***	0.00023***
	SpaceSize	3e-05***	3e-05***	3e-05***	3e-05***	0.0	-4e-05	-0.00029***	-0.00027***	-8e-05*	-2e-05***
	Rp_u	-3e-05*	-3e-05*	-3e-05*	-3e-05*	-2e-05*	-0.00038***	-3e-05	-1e-05	-0.00017***	-2e-05***
	Pop_{avg}	0.00016***	0.00016***	0.00016***	0.00016***	0.00017***	-0.00024***	7e-05	9e-05*	3e-05	0.0002***
	Pop_{skew}	1e-05	1e-05	1e-05	1e-05	-1e-05	-0.00028***	-5e-05	-7e-05	-4e-05	-1e-05*
	$LTail_{avg}$	1e-05	1e-05	1e-05	1e-05	1e-05*	0.00014*	6e-05	5e-05	7e-05*	1e-05
	$LTail_{skew}$	1e-05	1e-05	1e-05	1e-05	1e-05	0.00054***	7e-05	6e-05	0.00015***	1e-05*
	Accuracy	0.0003 ± 0.0003	0.0003 ± 0.0003	0.0003 ± 0.0003	0.0003 ± 0.0003	0.0002 ± 0.0002	0.0058 ± 0.0016	0.0015 ± 0.0011	0.0014 ± 0.001	0.0015 ± 0.0007	0.0002 ± 0.0002

5.2. Explanatory framework on accuracy target metric

We begin our experimental analysis by presenting the result of the explanatory study on accuracy as target metric, which we present in Table 3 (and in the Appendix, in Table A.7). The results obtained for the coefficient of determination (R) indicate that the 11 DCs can explain (on average) more than 90% of the variation in MAP and NDCG, respectively, in MovieLens datasets. It should be noted that for `BookCrossing`, NDCG is not reported because results were not reliable; this is due to this dataset being more sparse (see Figure 1) which produces the recommenders to generate relevant suggestions for less than 10% of the users, for this reason we decided to ignore this metric for this dataset and focus on MAP. In this situation (`BookCrossing`), these DCs can explain up to 70% of the metric variation. More specifically, by focusing on three random choices, `UserKNN-Cosine`, `ItemKNN-Adjusted`, and `BPR`, we can note that their corresponding values for adj.R in `ML-1M` are 0.861, 0.863, and 0.930, respectively. For `BookCrossing`, these values correspond to 0.330, 0.630, and 0.134.

However, when the significance of the DCs is considered, we observe a few surprising observations:

- The first observation is related to `BPR`, the state-of-the-art method for personalized ranking. We observe that this method does not get impacted by *SpaceSize*, while the rest of methods do in all the datasets. This is important because, as presented in Figure 1 and the accuracy rows in these tables, `BPR` is the best performing technique, hence, the fact that a DC is not helpful for this method might be particularly revealing to understand optimal requirements or constraints on input data of well-performing approaches.

- Most of the features significantly contribute to the explanation of the target metric (denoted with ***). Even in this case, a reasonable question we would like to answer is: can we tell them apart and understand which EV provides more impactful effects on the target metric? More specifically, would we be able to explain the variation in the target metric by using a smaller set of DCs?

- Whereas for MovieLens, the explanatory power remains quite high for all recommendation methods, in `BookCrossing` this is only true for ItemKNN and NMF.

To answer the first question regarding BPR, we carefully checked the regression coefficient results, and hypothesized that, due to the introduction of the newly introduced DCs in this work compared with previous works [7, 8] – namely Rp_u (ratings per user) – the latter captures all the necessary information in other DCs such as *Density*. Thus, when *Density* and Rp_u are used together in BPR, Rp_u becomes more impactful. However, when we removed the feature Rp_u, an additional set of experiments showed that *Density* became at that moment an important factor (hence, receiving ***, i.e., its p-value is significant). These results show the inter-dependence that exist between the different DCs and their effect on the regression model.

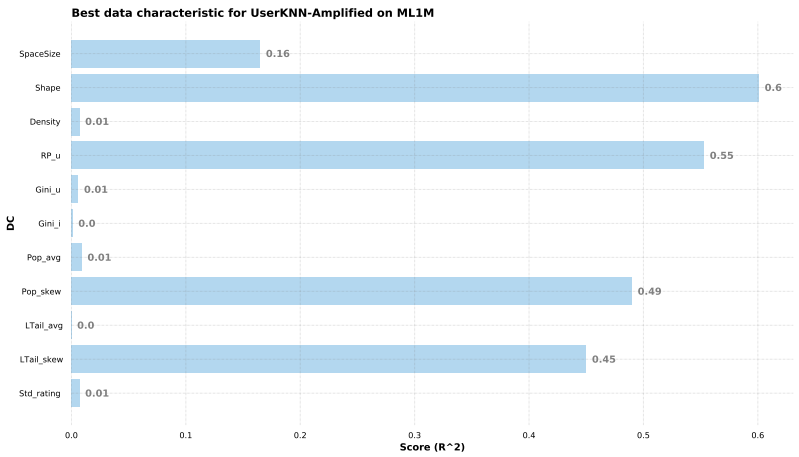
This first observation was a very good move forward to better understand the role and impact of DCs. This moved us to the second related question: if some of the DCs contain similar information as others, how can we distinguish between them? In particular, because the p-value just tells us which DCs are important but it does not say anything about how important they are.

To address this concern, we report the regression coefficient of determination R^2, for each of the DCs in isolation in Figure 3 and for all possible pairs of DCs as shown in visualization heatmaps provided in Figure 4 and Figure 5.

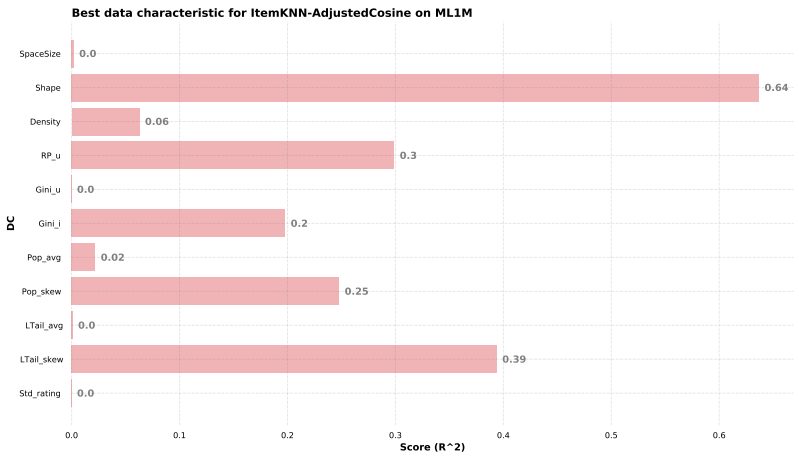
The results are quite insightful and suggest the following:

- We observe that *Shape* is the most impactful EV in all the reported recommender models together with Rp_u.
- `ItemKNN-Adjusted` has characteristics that are *(i)* **in common** with `UserKNN`, and also *(ii)* **different** from `UserKNN` and/or the rest of recommenders; for instance, only *Shape* explains more than 60% of memory-based models' accuracy-metric variations; Rp_u is the second important DCs for user-based models but it is less important for `ItemKNN-Adjusted`.
- Rp_u impacts almost both families of recommenders, memory-based and model-based quite significantly; for instance on `ML-1M`, `BPR-MF` can explain 65% of the accuracy variations and when combined with $LTail_{skew}$ about 75% of variations, which is substantially high. The only exception is `ItemKNN-Adjusted`, which is less impacted by Rp_u. In particular, this may suggest that this recommender could be useful in scenarios with cold-start users.
- A very insightful observation by this exploratory research is the impact of **skewness-based DCs**, namely Pop_{skew} and $LTail_{skew}$ on the overall performances; essentially what Pop_{skew} measures is the asymmetry of the probability distribution on user popularity profiles (or popularity biases); a high value of this asymmetry value in the absolute sense, it means the distribution has a longer tail, which can impact CF recommenders compared to the case where users have an average popularity profile.

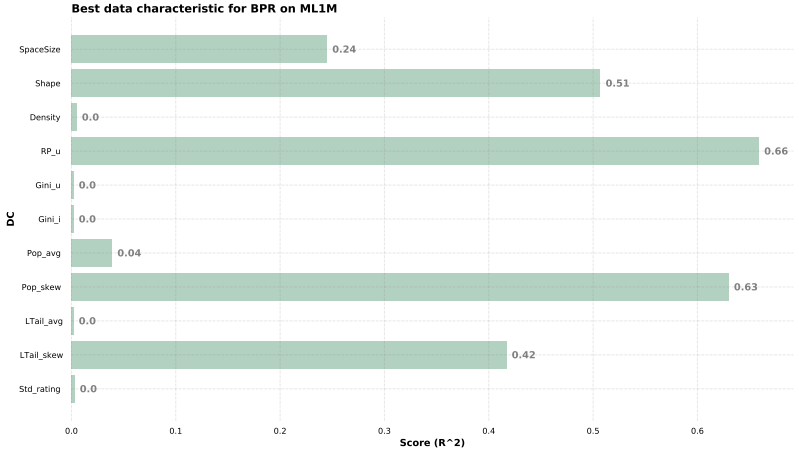
We now show the pairwise impact of DCs on two recommender models (due to space constraints, but results hold for the others). Thus, based on the results presented in Figures 4 and 5 we find the two best features among all. Hence, we can record *Shape* as the second most impactful EV (the first one after Rp_u). By following this process (inspired by the feature selection literature from Machine Learning), we obtain a ranking list based on the importance of these variables regarding their explanatory power with respect to each recommendation algorithm.



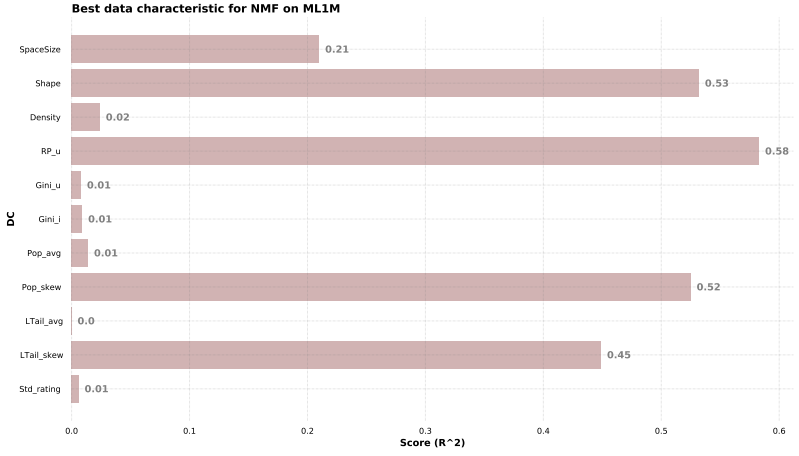
(a) UserKNN-Amplified



(b) ItemKNN-AdjustedCosine

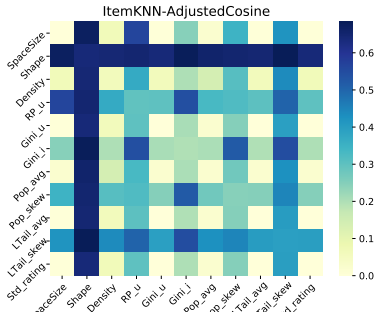


(c) BPR

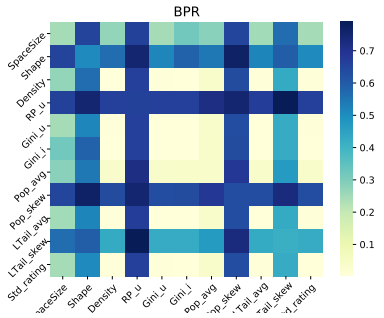


(d) NMF

Figure 3: Best features (data characteristics) according to R^2 for `ML-1M` using metric MAP

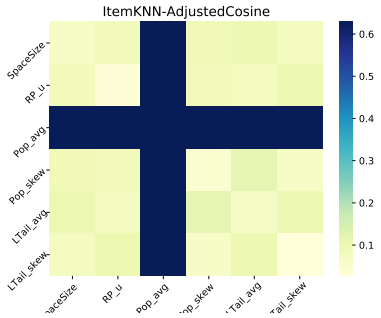


(a) ItemKNN with AdjustedCosine recommender.

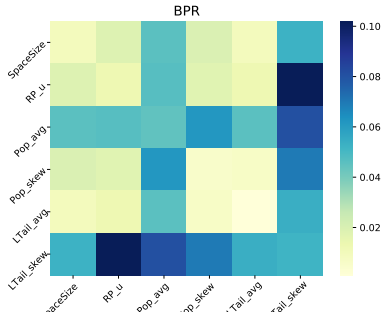


(b) BPR recommender.

Figure 4: Choosing the two best DCs on `ML-1M` dataset for two RSs: ItemKNN (left) and BPR (right).



(a) ItemKNN with AdjustedCosine recommender.



(b) BPR recommender.

Figure 5: Choosing the two best DCs on `BookCrossing` dataset for two RSs: ItemKNN (left) and BPR (right).

Table 4: Regression results for the within dataset analysis (target metric: fairness as MAD (MAP)).

Fairness MAD (MAP)		**Memory-based**					**Model-based**				
		UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
ML-100K	R^2 (adj.R)	0.265 (0.251)	0.263 (0.249)	0.265 (0.251)	0.264 (0.25)	0.277 (0.263)	0.327 (0.314)	0.233 (0.218)	0.255 (0.241)	0.234 (0.219)	0.241 (0.227)
	Constant	0.00166***	0.00174***	0.00181***	0.00178***	0.00237***	0.0113***	0.00583***	0.00628***	0.00456***	0.00386***
	SpaceSize	0.00037*	0.00036*	0.00037*	0.00036*	0.00013	0.00183*	0.00076	0.00093	0.00015	-0.00033
	Shape	-0.00018	-0.00024	-0.0002	-0.00024	-9e-05	-0.00125	-0.00049	-0.00076	-0.00057*	-0.00014
	Density	0.00084***	0.00081***	0.00081***	0.00081***	1e-05	0.00392***	0.00202***	0.00206***	0.00141***	0.00105***
	Rp_u	-0.00086***	-0.00088***	-0.00088***	-0.00089***	-0.00078***	-0.00555***	-0.00257***	-0.00292***	-0.00164***	-0.00081***
	$Gini_u$	4e-05	2e-05	2e-05	3e-05	-0.00025*	0.00029	0.0001	1e-05	-0.00011	-0.00019
	$Gini_i$	0.0003*	0.00029*	0.00031*	0.00029*	0.00059***	0.00215***	0.00118***	0.00114***	0.00028	9e-05
	Pop_{avg}	-0.0	0.0	2e-05	1e-05	0.00048***	7e-05	-0.00067	-0.00102*	-0.00044	-0.00057***
	Pop_{skew}	0.00028***	0.00031***	0.00028***	0.00031***	0.0003*	0.00118*	0.00095***	0.00095***	0.00022	0.00043*
	$LTail_{avg}$	-5e-05	-4e-05	-3e-05	-4e-05	-8e-05	4e-05	-0.0009***	-0.00095***	8e-05	-0.00029
	$LTail_{skew}$	0.00028***	0.00029***	0.00029***	0.00028***	0.00056***	0.00191***	0.00066*	0.00077*	6e-05	0.00074***
	Std_{rating}	0.0003***	0.0003***	0.0003***	0.0003***	5e-05	0.00085	0.00031	0.00025	0.0001	7e-05
	Accuracy	0.0019 ± 0.0023	0.002 ± 0.0023	0.002 ± 0.0023	0.002 ± 0.0023	0.0025 ± 0.0027	0.0116 ± 0.012	0.0065 ± 0.0075	0.0065 ± 0.0073	0.0047 ± 0.0044	0.0041 ± 0.0039
ML-1M	R^2 (adj.R)	0.431 (0.421)	0.436 (0.426)	0.431 (0.421)	0.415 (0.404)	0.255 (0.241)	0.392 (0.381)	0.307 (0.294)	0.338 (0.325)	0.302 (0.289)	0.39 (0.378)
	Constant	0.00061***	0.00084***	0.00061***	0.0008***	0.0008***	0.00483***	0.00235***	0.00263***	0.00287***	0.00106***
	SpaceSize	0.00033***	0.00027***	0.00033***	0.00032***	0.00013*	0.00081*	0.00061***	0.00055***	-5e-05	0.00021*
	Shape	-0.00013*	-9e-05	-0.00013*	3e-05	-3e-05	-0.0003	-0.00062***	-0.0007***	-0.00081***	-0.00018*
	Density	0.00023***	4e-05	0.00023***	0.00017*	2e-05	9e-05	0.00063***	0.00062***	0.00064*	0.00028***
	Rp_u	-0.00039***	-0.00043***	-0.00038***	-0.00046***	-0.00026***	-0.00176***	-0.00106***	-0.00109***	-0.00067*	-0.00047***
	$Gini_u$	-0.00018***	-0.0002***	-0.00018***	-0.00016***	-0.0	-9e-05	-1e-05	6e-05	0.00036*	-0.0001
	$Gini_i$	0.00045***	0.00058***	0.00044***	0.00054***	0.00029***	0.00297***	0.00088***	0.00089***	0.00058*	0.00036***
	Pop_{avg}	-3e-05	0.00011	-3e-05	-1e-05	0.00016***	0.00045	-8e-05	3e-05	-2e-05	0.00016*
	Pop_{skew}	0.00012*	0.00014*	0.00013*	6e-05	3e-05	0.00064*	0.00047*	0.00043*	0.00075***	0.00035***
	$LTail_{avg}$	-4e-05	-5e-05	-3e-05	2e-05	-3e-05	-0.00025	-0.0003*	-0.00041***	-0.00037*	-5e-05
	$LTail_{skew}$	0.00029***	0.00039***	0.00029***	0.00037***	0.00027***	0.00082***	0.00074***	0.00077***	0.0009***	0.00027***
	Std_{rating}	-9e-05*	-0.00012***	-8e-05*	-4e-05	-6e-05	-0.00012	-0.00013	-0.00022	-0.00056***	-1e-05
	Accuracy	0.0007 ± 0.0011	0.0008 ± 0.0013	0.0007 ± 0.0011	0.0008 ± 0.0013	0.0009 ± 0.0009	0.0048 ± 0.0057	0.0026 ± 0.0033	0.0027 ± 0.0033	0.0031 ± 0.004	0.0011 ± 0.0015
BookCrossing	R^2 (adj.R)	0.017 (0.003)	0.018 (0.004)	0.015 (0.001)	0.005 (-0.01)	0.023 (0.009)	0.069 (0.055)	0.077 (0.064)	0.062 (0.048)	0.064 (0.05)	0.04 (0.026)
	Constant	-1e-05	-2e-05	-1e-05	-3e-05	-0.0	0.0002	-0.00014	-0.00013	-2e-05	1e-05
	SpaceSize	-0.0	-0.0	-0.0	-0.0	-0.0*	-0.0***	-0.0***	-0.0***	-0.0***	-0.0*
	Rp_u	-0.00014	-0.0002*	-0.0001	-4e-05	-2e-05	-0.00031	7e-05	0.00011	-0.00032	2e-05
	Pop_{avg}	-0.01364	0.05176	-0.00586	-0.01904	0.00156	0.04031	-0.03635	-0.13582	-0.16387	0.01028
	Pop_{skew}	3e-05	3e-05	2e-05	1e-05	-1e-05	-8e-05	-0.0	-1e-05	-9e-05*	-1e-05
	$LTail_{avg}$	0.00391	0.00271	-9e-05	0.0014	-0.00146	0.01073	0.01864	-0.00123	-0.00024	0.00129
	$LTail_{skew}$	0.00014	0.00038	0.00038	0.00012	0.00032*	0.00189	-0.00085	0.00071	0.00211***	-4e-05
	Accuracy	0.0003 ± 0.0005	0.0003 ± 0.0005	0.0003 ± 0.0005	0.0003 ± 0.0006	0.0002 ± 0.0003	0.0025 ± 0.0029	0.0012 ± 0.0018	0.001 ± 0.0015	0.0011 ± 0.0014	0.0001 ± 0.0003

5.3. Explanatory framework on fairness-aware target metric

The second dimension of the explanatory study in this work, that differentiates it from the previous works [7, 8], is the attempt to explain the impact of DCs on the fairness of recommendation models. The results of the regression modeling based on the complete pool of DCs is presented in Table 4 (and extended in the Appendix in Table A.8), depending on how MAD is computed. These tables have been generated in a similar manner to Table 3 with the difference that they use the MAD metric as the dependent variable (cf. Section 3.3). When comparing these new results with those obtained for the accuracy dimension, two immediate observations are observed:

- The amount of explainability, derived from the R^2 statistics, is much smaller in the fairness dimension that in accuracy dimension.
- The variables that significantly contribute to the explainability of fairness, compared with accuracy, are much lower in terms of number, and different (at least partially) in terms of type.

As for the first observation, it can be seen that in `ML-100K` the proposed DCs can explain 33% of the target metric (fairness using NDCG as base metric) in the best case (`UserKNN-BM25`), and 15% in the lowest case (`ItemKNN-Adjusted`), while for `ML-1M`, these values are relatively higher for neighborhood-based models, e.g., 43% for `UserKNN-Cosine`, while 18% for `ItemKNN-Adjusted`. The corresponding values when the fairness metric uses MAP as base metric are similar, although the explainability is lower in `ML-100K`, but the results are more stable in `ML-1M`.

These results can be justified considering several viewpoints (i) the proposed set of DCs reflect the global characteristics of datasets, and not their biases; to answer unfairness based on data, it might be more relevant to capture data biases instead of global DCs, as influential factors in unfairness; (ii) the metric used as the DV for fairness may inherently entail lower explainability. Consider that, since MAD essentially computes the difference between the average MAP (or NDCG) of two groups, it could be that in some cases, two DCs have a neutralizing impact; for instance, one EV may impact the *Male* group significantly, while the other EV may impact the *Female* group also significantly, and due to the subtraction sign in the MAD metric, their impacts could be evened out. This means some DCs, despite being important, may not be identified as significantly impactful by the regression analysis.

In the direction of measuring and using biases instead of global DCs, we now provide a novel explorative procedure. Table 5 shows the results when using the DCs as in previous experiments, together with (at the bottom of the table) the results when the corresponding DCs of Male and Female are multiplied and then used as the explanatory variable. The explanation for multiplying the DCs corresponding to each sensitive group (in this case, males and females because we are reporting `ML-1M`) is that, given a fixed value for a global EV, such as $EV_M + EV_F = cte$, their multiplication $EV_M \times EV_F$ is maximal, when $EV_M = EV_F$ and different otherwise; thus the multiplication/interaction of two features from the constituting fairness groups (Male, Female) is effectively

measuring differences in DCs interpreted as data bias in this work. We can note that, interestingly, by defining DCs as the relation between Male and Female (or as we call them, biases) we get slightly more explainability. For instance, we can note that the R^2 statistics for `UserKNN-Cosine` increases from 0.431 to 0.471, an improvement on explanatory power of about 9.3%.

As for the second observation, the general trend is that a lower number of DCs contribute significantly to the fairness explainability, for instance *Shape*, *Density*, and Rp_u. We can find a number of nuances captured by these results, they include: first $Gini_i$, which was never significant in the accuracy study; in addition, we observe that BPR does not get impacted significantly by DCs, perhaps indicating robustness of this method against DC variations.

Table 5: Comparison in regression results (for target metric: fairness as MAD (MAP)) between using directly the DCs or their corresponding values normalized for males and females and multiplied.

Fairness MAD (MAP)		Memory-based					Model-based				
		UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
ML-1M	R^2 (adj.R)	0.431 (0.421)	0.436 (0.426)	0.431 (0.421)	0.415 (0.404)	0.255 (0.241)	0.392 (0.381)	0.307 (0.294)	0.338 (0.325)	0.302 (0.289)	0.39 (0.378)
	Constant	0.00061***	0.00084***	0.00061***	0.0008***	0.0008***	0.00483***	0.00235***	0.00263***	0.00287***	0.00106***
	SpaceSize	0.00033***	0.00027***	0.00033***	0.00032***	0.00013*	0.00081*	0.00061***	0.00055***	-5e-05	0.00021*
	Shape	-0.00013*	-9e-05	-0.00013*	3e-05	-3e-05	-0.0003	-0.00062***	-0.0007***	-0.00081***	-0.00018*
	Density	0.00023***	4e-05	0.00023***	0.00017*	2e-05	9e-05	0.00063***	0.00062***	0.00064*	0.00028***
	Rp_u	-0.00039***	-0.00043***	-0.00038***	-0.00046***	-0.00026***	-0.00176***	-0.00106***	-0.00109***	-0.00067*	-0.00047***
	$Gini_u$	-0.00018***	-0.0002***	-0.00018***	-0.00016***	-0.0	-9e-05	-1e-05	6e-05	0.00036*	-0.0001
	$Gini_i$	0.00045***	0.00058***	0.00044***	0.00054***	0.00029***	0.00297***	0.00088***	0.00089***	0.00058*	0.00036***
	Pop_{avg}	-3e-05	0.00011	-3e-05	-1e-05	0.00016***	0.00045	-8e-05	3e-05	-2e-05	0.00016*
	Pop_{skew}	0.00012*	0.00014*	0.00013*	6e-05	3e-05	0.00064*	0.00047*	0.00043*	0.00075***	0.00035***
	$LTail_{avg}$	-4e-05	-5e-05	-3e-05	2e-05	-3e-05	-0.00025	-0.0003*	-0.00041***	-0.00037*	-5e-05
	$LTail_{skew}$	0.00029***	0.00039***	0.00029***	0.00037***	0.00027***	0.00082***	0.00074***	0.00077***	0.0009***	0.00027***
	Std_{rating}	-9e-05*	-0.00012***	-8e-05*	-4e-05	-6e-05	-0.00012	-0.00013	-0.00022	-0.00056***	-1e-05
	Accuracy	0.0007 ± 0.0011	0.0008 ± 0.0013	0.0007 ± 0.0011	0.0008 ± 0.0013	0.0009 ± 0.0009	0.0048 ± 0.0057	0.0026 ± 0.0033	0.0027 ± 0.0033	0.0031 ± 0.004	0.0011 ± 0.0015
ML-1M (male, female multiplied)	R^2 (adj.R)	0.472 (0.462)	0.452 (0.442)	0.471 (0.462)	0.431 (0.42)	0.231 (0.216)	0.413 (0.402)	0.327 (0.314)	0.353 (0.341)	0.318 (0.305)	0.407 (0.396)
	Constant	0.00069***	0.00084***	0.00069***	0.00084***	0.00088***	0.00484***	0.00259***	0.00266***	0.00306***	0.00111***
	SpaceSize	0.00012*	0.00013*	0.00012*	0.00012	5e-05	0.0008***	0.00026	0.0002	2e-05	0.00016*
	Shape	-7e-05	0.0	-7e-05	7e-05	1e-05	-8e-05	-0.00019	-0.00027*	-0.0003	-7e-05
	Density	0.00077***	0.00061***	0.00076***	0.00079***	0.00018*	0.00123***	0.00166***	0.00158***	0.0008*	0.00043***
	Rp_u	-0.00016***	-0.00021***	-0.00015***	-0.00022***	-0.00013***	-0.00129***	-0.00061***	-0.00062***	-0.00056***	-0.00039***
	$Gini_u$	0.00023***	0.00026***	0.00023***	0.00024***	-0.0	0.00025	0.00017	0.00015	-9e-05	0.00011*
	$Gini_i$	6e-05	-7e-05	6e-05	8e-05	-0.0001	-0.00212***	-0.0001	-0.00013	-0.00087***	-0.00031*
	Pop_{avg}	-0.00022***	-0.00011	-0.00021***	-0.00024***	0.0001*	5e-05	-0.00039*	-0.00028	7e-05	0.00015
	Pop_{skew}	0.00022***	0.00031***	0.00023***	0.00026***	0.00012*	0.00104***	0.00053***	0.0005***	0.00044*	0.00023***
	$LTail_{avg}$	-8e-05*	-0.00011*	-8e-05*	-2e-05	2e-05	-0.0001	-0.00046***	-0.00054***	-0.0005***	-7e-05
	$LTail_{skew}$	0.00024***	0.00027***	0.00024***	0.00027***	0.00021***	0.0008***	0.00047***	0.00048***	0.00095***	0.00039***
	Std_{rating}	-7e-05	-9e-05*	-7e-05	-3e-05	-3e-05	-0.00023	0.0	-0.0001	-0.00064***	-3e-05
	Accuracy	0.0007 ± 0.0011	0.0008 ± 0.0013	0.0007 ± 0.0011	0.0008 ± 0.0013	0.0009 ± 0.0009	0.0048 ± 0.0057	0.0026 ± 0.0033	0.0027 ± 0.0033	0.0031 ± 0.004	0.0011 ± 0.0015

5.4. Discussion and Limitations

Based on the analysis of the results related to the presented approach, we are now ready to answer the three research questions posed in the introduction.

[**RQ1.**] *Which DCs contribute the most to the explainability of the performance of different families of recommendation algorithms when optimizing for accuracy? Is it possible to use a smaller set of DCs?*

We have presented a suite of DCs based on (i) the structure of the URM, (ii) the rating frequency of the URM, (iii) the item properties of user profiles, and (iv) distribution of rating values. After the data sanity phase, a number of col-linearly related DCs were discarded, leaving us with the pool of 11 DCs for `ML-100K` and `ML-1M`, and 6 DCs for `BookCrossing` categorized in the above four dimensions. Results of the explanatory study based on regression analysis indicate that:

- In general, $\underline{Rp_u}$ and $\underline{Shape}$ impact both types of recommendation models (memory and model-based) quite consistently. While *Shape* – which is a structural DC – impacts memory-based models more, Rp_u impacts model-based such as BPR more. What is very interesting in these results is the emergence of the DCs Pop_{skew} and $LTail_{skew}$. These are DCs that measure the skewness of the average popularity and average long tail items of user profiles. The way we can interpret this is that the higher the skewness, the more this distribution looks like a belly *short-head long-tail* distribution. For Pop_{skew} for instance, this means that few users are popular item consumers, while many are not (or vice versa). In this situation, `BPR` quality is impacted strongly.

- More specifically, between `ML-100K` and `ML-1M` dataset: Two out of three most important DCs are similar, they include: $\underline{Rp_u}$ and $\underline{Shape}$. For the third feature, they are *Density* (in `ML-100K`) and $\underline{LTail_{skew}}$ (in `ML-1M`), the latter one in a consistent way for all the recommendation models. The main exception is `BPR`, which gets impacted by the popularity bias, i.e., $\underline{Pop_{avg}}$, by a significant degree. See Table 6 for a summary. In `BookCrossing`, on the other hand, the number of significant DCs gets reduced, where $\underline{Pop_{avg}}$ and $\underline{SpaceSize}$ are the most important ones.

- Between recommendation algorithms: the general trend is that the latent-factor models `PMF` and `NMF` have a correlated behavior with `BPR`. `BPR`, `PMF`, and `NMF` are well-known to have a strong popularity bias in their recommendations, so it is not surprising that Pop_{avg} and Pop_{skew} are such an important characteristic for these approaches. However, `BPR` is the only method in `ML-100K` to have this characteristic in its top 3 (in detriment of *Density*); our hypothesis is that for `BPR`, Pop_{avg} is more influential than *Density* because it affects to how this technique works internally: since `BPR` samples items according to their observed interactions, those situations where items are clearly different with respect to their popularity would allow to capture better which items are relevant for a user, on the other hand, when all items are equally popular (the distribution is

flatter) this sampling procedure would work in a random way, impacting dramatically in the performance of BPR.

Furthermore, while neighborhood-based methods also show similar behavior, we notice that the behavior of Item-KNNAdjusted is considerably different from user-based methods. For instance, it is interesting to note *Density* is not impacting Item-KNNAdjusted in ML-1M, while *SpaceSize* does, a trend which is opposite of user-based methods. Finally, we can note that for UB-kNN, the similarity metric is not affected by the DCs.

The above results are reported based on individual and pairwise feature importance approach, shown in Figures 3, 4, and 5. We noticed that via the proposed feature selection approach we are able to explain more than 60% on accuracy variation (in some cases, up to 75%). This would allow to identify a small set of DCs to control for during training of recommendation algorithms.

[**RQ2.**] *How do these DCs change when the goal of the system is shifted towards fairness? is it easier to predict the impact of DCs for fairness or for accuracy?*

In general, our observation was that it is much harder to explain variation in fairness based on global DCs than that for accuracy. The amount of explainability for accuracy v.s. fairness in full case is about 80% on average, while for fairness is less than 40%. We tried to provide two explanations for this new finding: (i) data biases (as identified based on interaction of Male and Female features) can explain better fairness, (ii) the target fairness-evaluation metric may influence the generality of the obtained results.

Finally, we see slight differences between the target set of most important features: while for accuracy, Rp_u, *Shape*, and Pop_{skew} are very important features, for the fairness dimension Rp_u, *Shape*, and $Gini_i$ are the ones showing more explanatory power. Our hypothesis is that $Gini_i$, which measures how uniform the item distribution is, emphasizes internal biases in the data, such as a very concentrated or spread distribution, which seems to impact more strongly user groups (based on some predefined sensitive attributes) rather than the recommendation algorithms.

[**RQ3.**] *Is it possible to augment the set of DCs so that the inherent biases are also considered?*

To answer this question, for each DC, we computed the relative DC value with respect to its constituting groups, e.g., relative male to female DC value in ML-100K and ML-1M datasets. The goal is to measure the *proportionality biases* in the underlying data. Specifically, we used for a given EV, the relative EV according to $EV = EV_M \times EV_F$ (where M and F denote male and female users). By transforming DCs according to this procedure, following the data sanity check, and running regression analysis, we obtained improvement in explanation of fairness dimension, as summarized in Table 5. For instance for UserKNN-Amplified, ItemKNN-Adjusted, BPR-MF, and NMF, the changes in R^2 respectively are: 0.432 v.s 0.472, 0.255 v.s. 0.231, 0.392 v.s. 0.413 and 0.39 v.s. 0.407. Thus, we can note by performing this data transformation, we

Table 6: The best three impactful data characteristics (variables) based on their explanatory power measured via R^2 statistics. ✓ is used to note that such characteristic is among the 3 most impactful ones in `ML-100K`, and ✗ for `ML-1M`.

	UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
$SpaceSize$					✓		✗			
$Shape$	✓✗	✓✗	✓✗	✓✗	✓✗	✓	✓	✓✗	✓	✓
$Density$	✓	✓	✓	✓			✓	✓	✓	✓
Rp_u	✓✗	✓✗	✓✗	✓✗	✓✗	✓✗	✓✗	✓✗	✓✗	✓✗
$Gini_u$										
$Gini_i$										
Pop_{avg}						✓✗			✗	✗
Pop_{skew}										
$LTail_{avg}$										
$LTail_{skew}$	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Std_{rating}										

obtain about 10% improvement in the explanation of fairness metric. This transformation however did not improve the accuracy dimension as much.

These improvements open up the possibility to explore other features that might be relevant when injected in a multiplicative way; hence, we believe there is room for improvement and that further gains might be obtained based on our approach to incorporate biases for fairness explanation.

Limitations. It should be emphasized that, because of the experimental settings considered (where the recommendation models are not tuned to achieve an optimal performance), the proposed procedure would identify the variables that explain the performance of any given recommender, which we argue is closer to what any practitioner could find in the real world, especially for cases where fine-tuning is not feasible or proper groundtruth is not available. Because of this, we incorporate the quality control step presented in Section 5.1. An alternative possibility would be to test this framework on algorithms tuned with respect to some evaluation metric. In that case, the presented framework would identify which data characteristics explain the performance of optimized recommenders. However, how to define these 'optimized recommenders' is usually unclear; one needs to select the range of the parameters, the metric to be optimized, the cutoff of such metric, etc. Therefore, even though the experimental setting used in the paper is not the only possible one, we believe it might be useful on several scenarios, and, in any case, it might be applicable even when recommenders are tuned appropriately.

Moreover, at the moment the presented analysis produces different results depending on the dataset under consideration. This is somewhat expected as the data characteristics should change based on the dataset. However, it is important to emphasize that it is possible to extend the regression analysis to explore variability over datasets, as done in [7, 8]. We aim to incorporate this variation in the future, granted more datasets with sensitive attributes are available so we could also explore the explanatory power of the fairness dimension, as presented before.

Finally, it is worth mentioning another limitation of the current work. The regression analysis depends on the number of sensitive attributes and their granularity. While some sensitive attributes are binary or discrete in nature, such as gender or nationality, others are continuous (like age). Hence, the result

of the analysis could change dramatically depending on the number of sensitive classes the fairness metric aims to measure. Nonetheless, we want to emphasize that this limitation does not come from the framework *per se*, but from the high variability of state-of-the-art fairness metrics that depend on that; at the moment, the presented results could change if a different granularity of the attributes is considered, but the interesting aspect to take into account is that the same framework could be applied independently of the number of classes of such attributes.

6. Conclusions and future work

In this work, we have applied an explanatory framework based on regression models to better understand how data characteristics impact on the fairness and accuracy of recommender systems. We considered a suite of data characteristics, which can be classified according to: (i) the structure of the user-rating matrix, (ii) the rating frequency distribution, (iii) the item properties of the user profiles, and (iv) the distribution of rating values. We conducted extensive experiments based on sampling the original datasets to generate a large number of different training instances. Then, we compared and analyzed the significance of a wide array of data characteristics to verify the impact on the performance of classical recommendation approaches.

Our results show that the top three data characteristics may explain up to 80-90% of the performance (depending on the target metric) of recommendation algorithms when aiming for accuracy. These results, however, are not so positive when fairness is considered as target metric. This evidences the delicate entanglement between accuracy, fairness, and input information that should be managed in a recommendation system.

We have shown the potential of the explanatory framework presented in this work. We consider this framework is far from being fully explored; in particular, we envision many other useful variables could be incorporated into the analysis to derive fruitful conclusions. For instance, as done in [8] a between-dataset analysis could be incorporated into the model. Furthermore, we believe the regression model could be extended to also incorporate the hyper-parameters of the algorithms as explanatory variables, in such a way that richer outputs and explanations about the behavior of the models could be obtained. Additionally, and considering the current trend in the area where realistic training-test splits are preferred, the presented framework would benefit if the sampling could be done by satisfying temporal constraints. In this way, the derived conclusions could be extended more easily to production systems. This, however, is probably too difficult to achieve if timestamps are not realistic or the datasets do not cover a large period of time. In the future, we would like to explore the proposed framework to study the impact of data characteristics collected from implicit-feedback datasets on RSs performance. This is important because implicit datasets are more frequently used at recommendation engines in industrial applications. We also think explanatory frameworks like the one proposed in this work could be easily adapted in more specialized tasks such as based on

hybrid models employing content [13, 10] and multi-modal data [74, 11], or in session-based recommendation tasks [75, 76]. We would also like to extend the explanatory framework proposed here to counterfactual analysis, as done in the Machine Learning area for classifiers [77]. It should be noted that adapting such analysis to personalized data is probably not a straightforward task, as the data becomes sparser and the counterfactual conclusions would be less reliable.

Acknowledgements

This work was supported in part by the Ministerio de Ciencia, Innovación y Universidades (reference: PID2019-108965GB-I00), and in part by Servizi Locali 2.0, PON ARS01_00876 Bio-D, PON ARS01_00821 FLET4.0, PON ARS01_00917 OK-INSAID, H2020 PASSPARTOUT. The authors thank the reviewers for their thoughtful comments and suggestions.

References

[1] P. Knees, Y. Deldjoo, F. B. Moghaddam, J. Adamczak, G.-P. Leyson, P. Monreal, Recsys challenge 2019: session-based hotel recommendations, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 570–571.

[2] A. Gigli, F. Lillo, D. Regoli, Recommender systems for banking and financial services, in: D. Tikk, P. Pu (Eds.), Proceedings of the Poster Track of the 11th ACM Conference on Recommender Systems (RecSys 2017), Como, Italy, August 28, 2017, Vol. 1905 of CEUR Workshop Proceedings, CEUR-WS.org, 2017.

[3] R. Maestre, J. R. Duque, A. Rubio, J. Arévalo, Reinforcement learning for fair dynamic pricing, in: K. Arai, S. Kapoor, R. Bhatia (Eds.), Intelligent Systems and Applications - Proceedings of the 2018 Intelligent Systems Conference, IntelliSys 2018, London, UK, September 6-7, 2018, Volume 1, Vol. 868 of Advances in Intelligent Systems and Computing, Springer, 2018, pp. 120–135.

[4] M. D. Ekstrand, A. Das, R. Burke, F. Diaz, Fairness in recommender systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook (3rd edition), Springer, 2021.

[5] M. D. Ekstrand, M. Tian, I. M. Azpiazu, J. D. Ekstrand, O. Anuyah, D. McNeill, M. S. Pera, All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness, in: Conference on Fairness, Accountability and Transparency, 2018, pp. 172–186.

[6] Y. Deldjoo, V. W. Anelli, H. Zamani, A. B. Kouki, T. D. Noia, Recommender systems fairness evaluation via generalized cross entropy, 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, Vol. 2440 of CEUR Workshop Proceedings, CEUR-WS.org, 2019.

[7] G. Adomavicius, J. Zhang, Impact of data characteristics on recommender systems performance, ACM Trans. Manag. Inf. Syst. 3 (1) (2012) 3:1–3:17.

[8] Y. Deldjoo, T. D. Noia, E. D. Sciascio, F. A. Merra, How dataset characteristics affect the robustness of collaborative recommendation models, 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, ACM, 2020, pp. 951–960.

[9] S. M. McNee, J. Riedl, J. A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in: G. M. Olson, R. Jeffries (Eds.), Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006, ACM, 2006, pp. 1097–1101. doi:10.1145/1125451.1125659.
URL https://doi.org/10.1145/1125451.1125659

[10] C. Musto, M. d. Gemmis, P. Lops, F. Narducci, G. Semeraro, Semantics and content-based recommendations, in: Recommender Systems Handbook, 3rd Edition, Springer, 2021.

[11] Y. Deldjoo, M. Schedl, B. Hidasi, Y. Wei, X. He, Multimedia recommender systems: Algorithms and challenges, in: Recommender Systems Handbook, 3rd Edition, Springer, 2021.

[12] Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, Recommender systems leveraging multimedia content, ACM Computing Surveys (CSUR) 53 (5) (2020) 1–38.

[13] Y. Deldjoo, M. Schedl, P. Knees, Content-driven music recommendation: Evolution, state of the art, and challenges, arXiv preprint arXiv (2021).

[14] F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, 2015.

[15] I. Pilászy, D. Tikk, Recommending new movies: even a few ratings are more valuable than metadata, in: L. D. Bergman, A. Tuzhilin, R. D. Burke, A. Felfernig, L. Schmidt-Thieme (Eds.), Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009, ACM, 2009, pp. 93–100.

[16] R. Baeza-Yates, B. A. Ribeiro-Neto, Modern Information Retrieval - the concepts and technology behind search, Second edition, Pearson Education Ltd., Harlow, England, 2011.

[17] A. Gunawardana, G. Shani, Evaluating recommender systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, 2015, pp. 265–308. doi:10.1007/978-1-4899-7637-6_8.
URL https://doi.org/10.1007/978-1-4899-7637-6_8

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

[19] P. Castells, N. J. Hurley, S. Vargas, Novelty and diversity in recommender systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, 2015, pp. 881–918.

[20] Y. Deldjoo, T. D. Noia, F. A. Merra, A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks, ACM Computing Surveys (CSUR) 54 (2) (2021) 1–38.

[21] V. W. Anelli, Y. Deldjoo, T. Di Noia, F. Antonio, Adversarial recommender systems: Attack, defense, and advances, in: Recommender Systems Handbook, 3rd Edition, Springer, 2021.

[22] V. W. Anelli, Y. Deldjoo, T. D. Noia, A. Ferrara, F. Narducci, Federank: User controlled feedback with federated recommender systems, in: Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I, Vol. 12656 of Lecture Notes in Computer Science, Springer, 2021, pp. 32–47. doi:10.1007/978-3-030-72113-8_3.
URL https://doi.org/10.1007/978-3-030-72113-8_3

[23] B. P. Knijnenburg, S. Berkovsky, Privacy for recommender systems: tutorial abstract, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 394–395.

[24] K. Balog, F. Radlinski, Measuring recommendation explanation quality: The conflicting goals of explanations, 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, ACM, 2020, pp. 329–338. doi:10.1145/3397271.3401032.
URL https://doi.org/10.1145/3397271.3401032

[25] Y. Deldjoo, V. W. Anelli, H. Zamani, A. Bellogin, T. Di Noia, A flexible framework for evaluating user and item fairness in recommender systems, User Modeling and User-Adapted Interaction (2021) 1–47.

[26] M. D. Ekstrand, A. Das, R. Burke, F. Diaz, Fairness in recommender systems., in: Recommender Systems Handbook, 3rd Edition, Springer, 2021.

[27] L. M. Krebs, O. L. A. Rodriguez, P. Dewitte, J. Ausloos, D. Geerts, L. Naudts, K. Verbert, Tell me what you know: GDPR implications on designing transparency and accountability for news recommender systems, in: R. L. Mandryk, S. A. Brewster, M. Hancock, G. Fitzpatrick, A. L. Cox, V. Kostakos, M. Perry (Eds.), Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019, ACM, 2019. doi:10.1145/3290607.3312808.
URL https://doi.org/10.1145/3290607.3312808

[28] E. Gómez, L. Boratto, M. Salamó, Disparate impact in item recommendation: A case of geographic imbalance, in: D. Hiemstra, M. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I, Vol. 12656 of Lecture Notes in Computer Science, Springer, 2021, pp. 190–206. doi:10.1007/978-3-030-72113-8_13.
URL https://doi.org/10.1007/978-3-030-72113-8_13

[29] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, F. Diaz, Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems, in: A. Cuzzocrea, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, H. Wang (Eds.), Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, ACM, 2018, pp. 2243–2251.

[30] L. Boratto, G. Fenu, M. Marras, The effect of algorithmic bias on recommender systems for massive open online courses, in: L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, D. Hiemstra (Eds.), Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I, Vol. 11437 of Lecture Notes in Computer Science, Springer, 2019, pp. 457–472.

[31] I. Palomares, C. Porcel, L. Pizzato, I. Guy, E. Herrera-Viedma, Reciprocal recommender systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation, Information Fusion 69 (2021) 103–127.

[32] H. Abdollahpouri, R. Burke, Multistakeholder recommender systems., in: Recommender Systems Handbook, 3rd Edition, Springer, 2021.

[33] L. Akoglu, C. Faloutsos, Valuepick: Towards a value-oriented dual-goal recommender system, in: W. Fan, W. Hsu, G. I. Webb, B. Liu, C. Zhang, D. Gunopulos, X. Wu (Eds.), ICDMW 2010, The 10th IEEE International Conference on Data Mining Workshops, Sydney, Australia, 13 December 2010, IEEE Computer Society, 2010, pp. 1151–1158.

[34] R. Burke, N. Sonboli, A. Ordonez-Gauger, Balanced neighborhoods for multi-sided fairness in recommendation, in: S. A. Friedler, C. Wilson (Eds.), Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA, Vol. 81 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 202–214.

[35] R. Melchiorre, A., N. Parada-Cabaleiro, E. Brandl, S. Lesota, M. Schedl, Investigating gender fairness of recommendation algorithms in the music domain., Elsevier, 2021.

[36] A. Ferraro, X. Serra, C. Bauer, Break the loop: Gender imbalance in music recommenders, in: F. Scholer, P. Thomas, D. Elsweiler, H. Joho, N. Kando, C. Smith (Eds.), CHIIR '21: ACM SIGIR Conference on Human Information Interaction and Retrieval, Canberra, ACT, Australia, March 14-19, 2021, ACM, 2021, pp. 249–254. doi:10.1145/3406522.3446033.
URL https://doi.org/10.1145/3406522.3446033

[37] V. Tsintzou, E. Pitoura, P. Tsaparas, 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, Vol. 2440 of CEUR Workshop Proceedings, CEUR-WS.org, 2019.

[38] Z. Zhu, X. Hu, J. Caverlee, Fairness-aware tensor-based recommendation, in: A. Cuzzocrea, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, H. Wang (Eds.), Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, ACM, 2018, pp. 1153–1162.

[39] S. Yao, B. Huang, Beyond parity: Fairness objectives for collaborative filtering, in: I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 2921–2930.

[40] A. Bellogín, A. P. de Vries, Understanding similarity metrics in neighbour-based recommender systems, in: O. Kurland, D. Metzler, C. Lioma, B. Larsen, P. Ingwersen (Eds.), International Conference on the Theory of Information Retrieval, ICTIR '13, Copenhagen, Denmark, September 29 - October 02, 2013, ACM, 2013, p. 13.

[41] V. W. Anelli, T. D. Noia, E. D. Sciascio, C. Pomo, A. Ragone, On the discriminative power of hyper-parameters in cross-validation and how to choose them, in: T. Bogers, A. Said, P. Brusilovsky, D. Tikk (Eds.), Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019, ACM, 2019, pp. 447–451.

[42] E. B. Nilsen, D. E. Bowler, J. D. Linnell, Exploratory and confirmatory research in the open science era, Journal of Applied Ecology 57 (4) (2020) 842–847.

[43] X. Ning, G. Karypis, SLIM: sparse linear methods for top-n recommender systems, in: D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, X. Wu (Eds.), 11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011, IEEE Computer Society, 2011, pp. 497–506.

[44] Y. Liu, T. Pham, G. Cong, Q. Yuan, An experimental evaluation of point-of-interest recommendation in location-based social networks, Proc. VLDB Endow. 10 (10) (2017) 1010–1021.

[45] M. Elahi, M. Braunhofer, T. Gurbanov, F. Ricci, User preference elicitation, rating sparsity and cold start, in: S. Berkovsky, I. Cantador, D. Tikk (Eds.), Collaborative Recommendations - Algorithms, Practical Challenges and Applications, WorldScientific, 2018, pp. 253–294.

[46] Y. Shi, M. A. Larson, A. Hanjalic, List-wise learning to rank with matrix factorization for collaborative filtering, in: X. Amatriain, M. Torrens, P. Resnick, M. Zanker (Eds.), Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM, 2010, pp. 269–272.

[47] J. Wang, A. P. de Vries, M. J. T. Reinders, Unifying user-based and item-based collaborative filtering approaches by similarity fusion, in: E. N. Efthimiadis, S. T. Dumais, D. Hawking, K. Järvelin (Eds.), SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, ACM, 2006, pp. 501–508.

[48] H. Abdollahpouri, R. Burke, B. Mobasher, Managing popularity bias in recommender systems with personalized re-ranking, in: R. Barták, K. W. Brawner (Eds.), Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019, AAAI Press, 2019, pp. 413–418.

[49] A. Bellogín, P. Castells, I. Cantador, Statistical biases in information retrieval metrics for recommender systems, Inf. Retr. Journal 20 (6) (2017) 606–634.

[50] X. Ning, C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, Springer, 2015, pp. 37–76.

[51] Y. Deldjoo, M. F. Dacrema, M. G. Constantin, H. Eghbal-zadeh, S. Cereda, M. Schedl, B. Ionescu, P. Cremonesi, Movie genome: alleviating new item cold start in movie recommendation, User Model. User-Adapt. Interact. 29 (2) (2019) 291–343.

[52] C. Anderson, The long tail: Why the future of business is selling less of more, Hachette Books, 2006.

[53] M. Q. Pham, T. T. S. Nguyen, P. M. T. Do, A. Kozierkiewicz, Incremental svd-based collaborative filtering enhanced with diversity for personalized recommendation, in: Advances in Computational Collective Intelligence - 12th International Conference, ICCCI 2020, Da Nang, Vietnam, November 30 - December 3, 2020, Proceedings, Vol. 1287 of Communications in Computer and Information Science, Springer, 2020, pp. 212–223.

[54] Ò. Celma, P. Herrera, A new approach to evaluating novel recommendations, in: P. Pu, D. G. Bridge, B. Mobasher, F. Ricci (Eds.), Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008, ACM, 2008, pp. 179–186.

[55] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, TiiS 5 (4) (2016) 19:1–19:19.

[56] 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: R. L. T. Santos, L. B. Marinho, E. M. Daly, L. Chen, K. Falk, N. Koenigstein, E. S. de Moura (Eds.), RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020, ACM, 2020, pp. 23–32. doi:10.1145/3383313.3412489.
URL https://doi.org/10.1145/3383313.3412489

[57] D. Kluver, J. A. Konstan, Evaluating recommender behavior for new users, in: A. Kobsa, M. X. Zhou, M. Ester, Y. Koren (Eds.), Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014, ACM, 2014, pp. 121–128. doi:10.1145/2645710.2645742.
URL https://doi.org/10.1145/2645710.2645742

[58] J. S. Breese, D. Heckerman, C. M. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: G. F. Cooper, S. Moral (Eds.), UAI '98: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, University of Wisconsin Business School, Madison, Wisconsin, USA, July 24-26, 1998, Morgan Kaufmann, 1998, pp. 43–52.

[59] J. L. Herlocker, J. A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: F. C. Gey, M. A. Hearst, R. M. Tong (Eds.), SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA, ACM, 1999, pp. 230–237.

[60] D. M. Kelen, D. Berecz, F. Béres, A. A. Benczúr, Efficient K-NN for playlist continuation, in: Proceedings of the ACM Recommender Systems Challenge, RecSys Challenge 2018, Vancouver, BC, Canada, October 2, 2018, ACM, 2018, pp. 6:1–6:4.

[61] Y. Wu, C. DuBois, A. X. Zheng, M. Ester, Collaborative denoising auto-encoders for top-n recommender systems, in: P. N. Bennett, V. Josifovski, J. Neville, F. Radlinski (Eds.), Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016, ACM, 2016, pp. 153–162.

[62] I. Cantador, A. Bellogín, D. Vallet, Content-based recommendation in social tagging systems, in: X. Amatriain, M. Torrens, P. Resnick, M. Zanker (Eds.), Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM, 2010, pp. 237–240.

[63] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Analysis of recommendation algorithms for e-commerce, in: Proceedings of the 2nd ACM conference on Electronic commerce, ACM, 2000, pp. 158–167.

[64] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: V. Y. Shen, N. Saito, M. R. Lyu, M. E. Zurko (Eds.), Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001, ACM, 2001, pp. 285–295.

[65] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[66] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: X. Amatriain, M. Torrens, P. Resnick, M. Zanker (Eds.), Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM, 2010, pp. 39–46.

[67] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, IEEE Computer Society, 2008, pp. 263–272.

[68] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: bayesian personalized ranking from implicit feedback, in: J. A. Bilmes, A. Y. Ng (Eds.), UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009, AUAI Press, 2009, pp. 452–461.

[69] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: J. C. Platt, D. Koller, Y. Singer, S. T. Roweis (Eds.), 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, Curran Associates, Inc., 2007, pp. 1257–1264.

[70] D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, in: T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, MIT Press, 2000, pp. 556–562.

[71] A. Salah, Q.-T. Truong, H. W. Lauw, Cornac: A comparative framework for multimodal recommender systems, Journal of Machine Learning Research 21 (95) (2020) 1–5. URL http://jmlr.org/papers/v21/19-805.html

[72] R. A. Stine, Graphical interpretation of variance inflation factors, The American Statistician 49 (1) (1995) 53–56.

[73] C. Robinson, R. E. Schumacker, Interaction effects: centering, variance inflation factor, and interpretation issues, Multiple linear regression viewpoints 35 (1) (2009) 6–11.

[74] Y. Deldjoo, J. R. Trippas, H. Zamani, Towards multi-modal conversational information seeking, in: Proceedings of the 44th International ACM SIGIR conference on research and development in Information Retrieval, 2021.

[75] M. Quadrana, P. Cremonesi, D. Jannach, Sequence-aware recommender systems, ACM Computing Surveys (CSUR) 51 (4) (2018) 1–36.

[76] J. Adamczak, Y. Deldjoo, F. B. Moghaddam, P. Knees, G.-P. Leyson, P. Monreal, Session-based hotel recommendations dataset: As part of the acm recommender system challenge 2019, ACM Transactions on Intelligent Systems and Technology (TIST) 12 (1) (2020) 1–20.

[77] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: M. Hildebrandt, C. Castillo, E. Celis, S. Ruggieri, L. Taylor, G. Zanfir-Fortuna (Eds.), FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, ACM, 2020, pp. 607–617.

Appendix A. Extended results

In this appendix we show some results that complement those included in the rest of the paper. First, Table A.7 shows the same results as in Table 3 but for target metric NDCG (recall that this metric did not produce reliable performance results in `BookCrossing`). Similarly, Table A.8 shows the equivalent results to those presented in Table 4 but for NDCG, for the same reasons as before.

Additionally, in Table A.9 we include an analysis on how the explainability changes depending on the number of samples considered. It is important to observe that the results, in general, hold, independently of the number of samples, however, we decided to use 600 samples as in previous works [7, 8].

Table A.7: Regression results for the within dataset analysis (target metric: NDCG@100).

Accuracy nDCG@100		**Memory-based**					**Model-based**				
		UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
ML-100K	R^2 (adj.R)	0.9 (0.898)	0.9 (0.898)	0.9 (0.898)	0.9 (0.898)	0.918 (0.916)	0.927 (0.925)	0.862 (0.86)	0.863 (0.86)	0.894 (0.892)	0.915 (0.913)
	Constant	0.09129***	0.08548***	0.08963***	0.08562***	0.0882***	0.23965***	0.1446***	0.14888***	0.12805***	0.12924***
	SpaceSize	0.00774***	0.00753***	0.00775***	0.00753***	0.01347***	0.00096	0.00837***	0.00814***	0.00481***	0.00241*
	Shape	0.01504***	0.01507***	0.01498***	0.01506***	0.01308***	0.01062***	0.01184***	0.01225***	0.0089***	0.01367***
	Density	0.01284***	0.01285***	0.01284***	0.01286***	0.00574***	-0.00166	0.01306***	0.0117***	0.00905***	0.01021***
	Rp_u	-0.03233***	-0.03229***	-0.03237***	-0.03229***	-0.0282***	-0.0321***	-0.03001***	-0.0289***	-0.02483***	-0.03204***
	$Gini_u$	-0.00304***	-0.00301***	-0.00302***	-0.00301***	-0.003***	-0.00171***	-0.00298***	-0.00309***	-0.00248***	-0.00238***
	$Gini_i$	0.00213*	0.00207*	0.00206*	0.00207*	1e-05	0.00358***	0.00111	0.00113	0.00126	0.00135
	Pop_{avg}	0.00099	0.00097	0.00096	0.00096	0.00342***	0.01137***	0.001	0.0014	0.00214***	0.00226*
	Pop_{skew}	0.00539***	0.00544***	0.00546***	0.00543***	0.00214***	0.00555***	0.00542***	0.00515***	0.00316***	0.00481***
	$LTail_{avg}$	-8e-05	-0.00013	-7e-05	-0.00013	0.00108*	0.00198***	0.00028	0.00061	0.00078	0.00045
	$LTail_{skew}$	0.00693***	0.007***	0.00698***	0.007***	0.00323***	0.00501***	0.00651***	0.00631***	0.0035***	0.00542***
	Std_{rating}	0.00154*	0.00156*	0.0016*	0.00156*	0.0002	0.00012	0.00222***	0.00195*	0.00024	0.00226***
	Accuracy	0.0925 ± 0.0484	0.0923 ± 0.0485	0.0926 ± 0.0484	0.0923 ± 0.0485	0.0903 ± 0.0365	0.2589 ± 0.047	0.1587 ± 0.0438	0.1566 ± 0.0429	0.1411 ± 0.0349	0.1336 ± 0.0484
ML-1M	R^2 (adj.R)	0.903 (0.901)	0.905 (0.904)	0.903 (0.901)	0.918 (0.916)	0.891 (0.889)	0.954 (0.953)	0.851 (0.848)	0.865 (0.863)	0.907 (0.905)	0.925 (0.924)
	Constant	0.04839***	0.05745***	0.04976***	0.05759***	0.07027***	0.20276***	0.11985***	0.12334***	0.13607***	0.06403***
	SpaceSize	0.00696***	0.00576***	0.00698***	0.00744***	0.00769***	0.0026***	0.01375***	0.01047***	0.0061***	0.00656***
	Shape	0.00945***	0.00859***	0.00948***	0.01196***	0.00837***	0.00805***	0.00959***	0.00945***	0.00447***	0.00988***
	Density	0.00325***	0.00221*	0.00322***	0.00269***	0.00116	-0.00458***	0.00341***	0.00271*	0.00226*	0.00428***
	Rp_u	-0.01947***	-0.02316***	-0.01948***	-0.02328***	-0.01509***	-0.02909***	-0.02343***	-0.02205***	-0.02443***	-0.02392***
	$Gini_u$	-0.00173***	-0.00133*	-0.00173***	-0.00166***	-0.00138***	-0.00114***	-0.00116	-0.00126*	-0.00118*	-0.00154***
	$Gini_i$	0.00303***	0.0011	0.00304***	0.00026	-0.0027***	0.00768***	0.00431***	0.00206	0.00392***	0.00529***
	Pop_{avg}	0.00212***	0.00186*	0.00213***	0.00163*	0.00231***	0.01286***	0.00646***	0.00578***	0.00679***	0.00404***
	Pop_{skew}	0.00416***	0.00525***	0.00415***	0.00498***	0.00236***	0.00672***	0.00398***	0.00369***	0.0039***	0.00543***
	$LTail_{avg}$	-0.00199***	-0.00257***	-0.00196***	-0.00242***	-0.00184***	-0.00126***	-0.00173***	-0.00235***	-0.00224***	-0.00163***
	$LTail_{skew}$	0.01101***	0.01283***	0.01098***	0.01074***	0.00912***	0.01135***	0.00875***	0.00848***	0.00923***	0.01106***
	Std_{rating}	-0.00199***	-0.00162***	-0.002***	-0.00146***	-0.00069	-0.00128***	0.00015	0.00022	-0.0007	-0.00113*
	Accuracy	0.0516 ± 0.0346	0.063 ± 0.0399	0.0516 ± 0.0346	0.064 ± 0.0402	0.0711 ± 0.0278	0.2207 ± 0.0466	0.1288 ± 0.035	0.1297 ± 0.0342	0.1461 ± 0.0345	0.0711 ± 0.04

Table A.8: Regression results for the within dataset analysis (target metric: fairness as MAD (NDCG@100)).

Fairnesss MAD (NDCG@100)		**Memory-based**					**Model-based**				
		UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
ML-100K	R^2 (adj.R)	0.322 (0.309)	0.339 (0.327)	0.323 (0.31)	0.339 (0.327)	0.155 (0.139)	0.236 (0.222)	0.251 (0.237)	0.251 (0.237)	0.197 (0.182)	0.233 (0.219)
	Constant	0.00646***	0.00698***	0.007***	0.00684***	0.00735***	0.01433***	0.01173***	0.01164***	0.00949***	0.00905***
	SpaceSize	0.00102*	0.00107*	0.00105*	0.00105*	0.00049	0.00066	1e-05	-6e-05	-0.00076	-0.00012
	Shape	-0.00158***	-0.00161***	-0.00158***	-0.00162***	-0.00039	-0.00255***	-0.0021***	-0.00227***	-0.00112*	-0.00173***
	Density	0.00163***	0.00157***	0.00159***	0.00155***	0.00054	0.00372***	0.00224*	0.00211*	0.00167*	0.00256***
	Rp_u	-0.00372***	-0.00367***	-0.00365***	-0.00367***	-0.00197***	-0.00481***	-0.00356***	-0.00356***	-0.00181***	-0.00224***
	$Gini_u$	0.00013	5e-05	0.00011	5e-05	-0.00045	0.00033	0.00023	-0.00018	0.00041	-0.00048
	$Gini_i$	0.00126***	0.00142***	0.00135***	0.00142***	0.00131***	0.00179*	0.00241***	0.00229***	0.00123*	0.00067
	Pop_{avg}	0.00055	0.00074*	0.0007*	0.00076*	0.00072	4e-05	0.00053	0.00011	-0.00057	-0.00022
	Pop_{skew}	0.00029	0.00032	0.0003	0.00031	0.00054	0.00026	0.00012	2e-05	0.00025	2e-05
	$LTail_{avg}$	-0.00047	-0.00047	-0.00048	-0.00047	-0.00012	0.00023	-0.00104*	-0.00141***	0.00047	-0.00039
	$LTail_{skew}$	0.00054*	0.00057*	0.00059*	0.00056*	0.00031	0.0018***	0.00087	0.00095*	-0.00031	0.00053
	Std_{rating}	0.00054*	0.00046	0.00045	0.00047	0.0002	0.0012*	0.0001	-0.0001	0.00083*	0.00054
	Accuracy	0.0071 ± 0.0067	0.007 ± 0.0067	0.0071 ± 0.0067	0.007 ± 0.0067	0.0076 ± 0.0072	0.0156 ± 0.0135	0.0127 ± 0.0116	0.0124 ± 0.0114	0.0098 ± 0.0089	0.0096 ± 0.0083
ML-1M	R^2 (adj.R)	0.434 (0.424)	0.419 (0.408)	0.434 (0.423)	0.402 (0.391)	0.185 (0.17)	0.323 (0.31)	0.293 (0.28)	0.309 (0.296)	0.248 (0.234)	0.398 (0.386)
	Constant	0.00215***	0.00242***	0.00213***	0.0026***	0.0028***	0.00637***	0.00477***	0.00463***	0.00521***	0.00278***
	SpaceSize	0.00088***	0.00078***	0.00087***	0.00071***	-0.00013	0.00025	0.00086*	0.00067	3e-05	0.00036*
	Shape	-0.00089***	-0.00068***	-0.00087***	-0.00079***	-0.0004***	-0.00053	-0.00122***	-0.00138***	-0.0012***	-0.00059***
	Density	0.00016	-0.00033	0.00011	-4e-05	0.00032	0.00119*	0.00231***	0.00217***	0.00106*	0.00012
	Rp_u	-0.00168***	-0.00171***	-0.00166***	-0.0017***	-0.00084***	-0.00121*	-0.00155***	-0.00146***	-0.00129***	-0.00109***
	$Gini_u$	-0.00057***	-0.00058***	-0.00056***	-0.00056***	0.00034***	0.00024	0.0001	-1e-05	0.0004	8e-05
	$Gini_i$	0.00146***	0.00182***	0.00148***	0.00142***	0.00021	0.00298***	0.0008	0.00079	0.00124***	0.00127***
	Pop_{avg}	0.00014	0.00031	0.00015	0.00013	0.00033*	-0.0	-0.00092***	-0.0007*	-0.00014	0.00027
	Pop_{skew}	0.00025	-4e-05	0.00024	8e-05	-0.00014	0.00057	0.00049	0.00046	0.00051	0.0006***
	$LTail_{avg}$	0.00013	0.00026*	0.00014	0.00011	-0.00013	-0.00015	-0.00039	-0.00058***	-0.00072***	-0.00014
	$LTail_{skew}$	0.00074***	0.0008***	0.00071***	0.00084***	0.00046***	0.00034	0.00115***	0.00131***	0.00095***	0.00047***
	Std_{rating}	9e-05	2e-05	9e-05	-4e-05	5e-05	0.00013	8e-05	2e-05	-0.00047*	0.00017
	Accuracy	0.0024 ± 0.0033	0.0026 ± 0.0034	0.0024 ± 0.0033	0.0029 ± 0.0034	0.0029 ± 0.0027	0.0071 ± 0.0074	0.0051 ± 0.0059	0.0052 ± 0.0058	0.0054 ± 0.0062	0.003 ± 0.0031

Table A.9: Regression results for the within dataset analysis when changing the number of samples (target metric: MAP).

	Target MAP	Memory-based					Model-based				
		UserKNN-Amplified	UserKNN-BM25	UserKNN-Cosine	UserKNN-IDF	ItemKNN-Adjusted	BPR	MF	SVD	PMF	NMF
ML-1M 10% of samples	R^2 (adj.R)	0.861 (0.858)	0.848 (0.844)	0.872 (0.869)	0.859 (0.856)	0.858 (0.855)	0.929 (0.927)	0.711 (0.705)	0.745 (0.739)	0.849 (0.846)	0.876 (0.874)
	$SpaceSize$	0.0***	0.0*	0.0***	0.0*	0.0***	0.0	0.0***	0.0***	0.0***	0.0***
	$Shape$	0.0009***	0.00105***	0.00092***	0.00155***	0.00064***	0.00216***	0.00171***	0.00152***	0.00098***	0.00128***
	$Density$	0.20084***	0.09057	0.17931***	0.14358	0.11192*	-1.69882***	0.29074	0.18977	0.2299	0.43962***
	Rp_u	-5e-05***	-6e-05***	-5e-05***	-6e-05***	-6e-05***	-0.00025***	-0.00015***	-0.00013***	-0.00019***	-9e-05***
	$Gini_u$	-0.02004***	-0.00963	-0.02278***	-0.01924*	-0.01451*	-0.03729	-0.02959	-0.00679	-0.05815***	-0.03274***
	$Gini_i$	0.02541***	0.02513*	0.02647***	0.00909	-0.02864***	0.24251***	0.05867*	0.00657	0.08546***	0.05314***
	Pop_{avg}	0.04931***	0.07545***	0.05904***	0.05846*	0.06679***	1.06203***	0.29591***	0.29901***	0.36944***	0.11726***
	Pop_{skew}	0.00313***	0.00597***	0.00265***	0.00583***	0.00146***	0.02369***	0.00503*	0.00763***	0.00831***	0.00654***
	$LTail_{avg}$	-0.10792***	-0.12076***	-0.12429***	-0.13506***	-0.08756***	-0.2235*	-0.26582***	-0.18109*	-0.34633***	-0.12707***
	$LTail_{skew}$	0.00831***	0.01299***	0.00858***	0.0096***	0.00801***	0.03779***	0.01682***	0.01499***	0.02148***	0.01585***
	Std_{rating}	-0.02189*	-0.01749	-0.02178*	-0.00467	-0.00743	-0.06878*	0.05447	0.1109***	-0.0138	-0.01484
	Accuracy	0.0161 ± 0.0059	0.019 ± 0.0085	0.0163 ± 0.006	0.0195 ± 0.0091	0.0207 ± 0.0052	0.0864 ± 0.0264	0.0442 ± 0.0143	0.0443 ± 0.013	0.0516 ± 0.0149	0.0226 ± 0.0103
ML-1M 20% of samples	R^2 (adj.R)	0.858 (0.855)	0.859 (0.855)	0.876 (0.872)	0.853 (0.849)	0.871 (0.868)	0.926 (0.924)	0.696 (0.688)	0.762 (0.755)	0.855 (0.852)	0.871 (0.868)
	$SpaceSize$	0.0***	0.0	0.0***	0.0*	0.0***	0.0*	0.0***	0.0***	0.0***	0.0***
	$Shape$	0.00095***	0.00109***	0.00094***	0.00157***	0.00062***	0.00192***	0.00169***	0.00159***	0.00103***	0.00111***
	$Density$	0.25582***	0.00711	0.13943*	0.32581***	0.1643***	-1.90918***	0.4385	0.39799	0.35887	0.40915***
	Rp_u	-5e-05***	-6e-05***	-5e-05***	-6e-05***	-5e-05***	-0.00026***	-0.00016***	-0.00013***	-0.00018***	-0.0001***
	$Gini_u$	-0.01741*	-0.00131	-0.02743***	-0.02452*	-0.00512	-0.04004	-0.04061	-0.01095	-0.00243	-0.01744
	$Gini_i$	0.03119***	0.02783*	0.03682***	0.00139	-0.03652***	0.28185***	0.04554	-0.02479	0.05609***	0.04457***
	Pop_{avg}	0.04203*	0.06436*	0.06825***	0.02093	0.06696***	1.14946***	0.29732***	0.29459***	0.36885***	0.15916***
	Pop_{skew}	0.00249***	0.00665***	0.00213***	0.00638***	0.00207***	0.02497***	0.00527*	0.00705***	0.01064***	0.00504***
	$LTail_{avg}$	-0.10169***	-0.15129***	-0.12087***	-0.08273	-0.04943	-0.16398	-0.182	-0.19372*	-0.29175***	-0.13853***
	$LTail_{skew}$	0.00871***	0.01449***	0.00927***	0.00955***	0.00751***	0.03884***	0.01703***	0.01398***	0.01762***	0.01699***
	Std_{rating}	-0.02628*	-0.01519	-0.0127	-0.01942	0.00218	-0.07307*	0.06119	0.10775***	0.01393	0.00179
	Accuracy	0.0162 ± 0.0059	0.0193 ± 0.009	0.0165 ± 0.0062	0.0194 ± 0.009	0.0207 ± 0.0051	0.0863 ± 0.0257	0.0445 ± 0.0149	0.0445 ± 0.0136	0.0514 ± 0.0143	0.0227 ± 0.01
ML-1M 50% of samples	R^2 (adj.R)	0.832 (0.825)	0.853 (0.847)	0.863 (0.858)	0.856 (0.851)	0.875 (0.87)	0.931 (0.928)	0.672 (0.659)	0.753 (0.744)	0.853 (0.848)	0.875 (0.871)
	$SpaceSize$	0.0***	0.0	0.0***	0.0*	0.0***	0.0*	0.0***	0.0***	0.0***	0.0***
	$Shape$	0.00076***	0.00122***	0.00088***	0.00161***	0.00065***	0.00193***	0.00154***	0.00142***	0.00116***	0.00121***
	$Density$	0.26872***	0.12033	0.09572	0.38869***	0.15933*	-1.39887***	-0.05485	0.35969	0.61439*	0.35838***
	Rp_u	-5e-05***	-5e-05***	-5e-05***	-5e-05***	-5e-05***	-0.00024***	-0.00015***	-0.00013***	-0.00018***	-0.00011***
	$Gini_u$	-0.00356	0.02361	-0.02964***	-0.03039*	-0.00642	-0.07353***	-0.01156	-0.02729	0.01201	0.00126
	$Gini_i$	0.02148*	0.00885	0.03564***	0.00256	-0.03564***	0.24642***	0.04755	-0.04167	0.06687***	0.06128***
	Pop_{avg}	0.04051	0.05217	0.08452***	0.01873	0.0632***	1.05682***	0.32432***	0.33649***	0.36431***	0.21573***
	Pop_{skew}	0.00375***	0.00696***	0.00159*	0.00703***	0.00192***	0.02774***	0.00578*	0.00804***	0.0099***	0.00393***
	$LTail_{avg}$	-0.12143***	-0.18236***	-0.09734*	-0.11712*	-0.08977***	-0.15476	-0.11594	-0.13081	-0.36575***	-0.0435
	$LTail_{skew}$	0.00863***	0.01416***	0.00991***	0.00798***	0.00665***	0.03953***	0.01389***	0.01537***	0.01768***	0.01556***
	Std_{rating}	-0.02889*	-0.01667	-0.00888	-0.03846	0.02176*	-0.10871*	0.09504*	0.1227***	0.00972	0.01441
	Accuracy	0.0159 ± 0.0053	0.0196 ± 0.0095	0.0164 ± 0.0061	0.0196 ± 0.0087	0.0207 ± 0.0052	0.0859 ± 0.0252	0.0438 ± 0.0138	0.0445 ± 0.0141	0.052 ± 0.0152	0.0226 ± 0.0099
ML-1M full samples	R^2 (adj.R)	0.863 (0.861)	0.845 (0.842)	0.864 (0.861)	0.861 (0.858)	0.866 (0.863)	0.931 (0.93)	0.709 (0.704)	0.746 (0.742)	0.849 (0.847)	0.878 (0.875)
	$SpaceSize$	0.0007***	0.00051*	0.0007***	0.00056*	0.00234***	0.00087	0.00459***	0.0031***	0.00255***	0.00125***
	$Shape$	0.00258***	0.00301***	0.00258***	0.00448***	0.00198***	0.00644***	0.00517***	0.00515***	0.00278***	0.00363***
	$Density$	0.00063***	0.00041	0.00063***	0.00051	0.00033*	-0.00503***	0.00126	0.00073	0.00037	0.00137***
	Rp_u	-0.00208***	-0.00288***	-0.00208***	-0.00258***	-0.00251***	-0.01121***	-0.00682***	-0.00622***	-0.00855***	-0.00399***
	$Gini_u$	-0.00033***	-0.00019	-0.00033***	-0.00027	-0.00023***	-0.00062*	-0.00051	-0.00056	-0.00073***	-0.0005***
	$Gini_i$	0.00075***	0.00062*	0.00074***	0.00035	-0.00078***	0.00692***	0.00129	0.00037	0.00262***	0.00159***
	Pop_{avg}	0.0005***	0.00054*	0.0005***	0.00053*	0.00067***	0.00937***	0.00249***	0.00239***	0.0034***	0.00113***
	Pop_{skew}	0.0008***	0.00147***	0.00079***	0.00146***	0.00032*	0.006***	0.00145***	0.00139***	0.00213***	0.0016***
	$LTail_{avg}$	-0.00038***	-0.00057***	-0.00038***	-0.00048***	-0.00033***	-0.00093***	-0.00077*	-0.001***	-0.00139***	-0.00046***
	$LTail_{skew}$	0.00168***	0.0026***	0.00168***	0.00188***	0.00154***	0.00745***	0.0031***	0.00292***	0.00432***	0.00307***
	Std_{rating}	-0.00024*	-0.00018	-0.00024*	-8e-05	-9e-05	-0.00082***	0.00069*	0.00068*	-0.00036	-0.00019
	Accuracy	0.0162 ± 0.0059	0.019 ± 0.0085	0.0162 ± 0.0059	0.0193 ± 0.0089	0.0207 ± 0.0053	0.0867 ± 0.0266	0.044 ± 0.0141	0.0445 ± 0.0137	0.0517 ± 0.0149	0.0226 ± 0.0101