ECAI 2023
K. Gal et al. (Eds.)

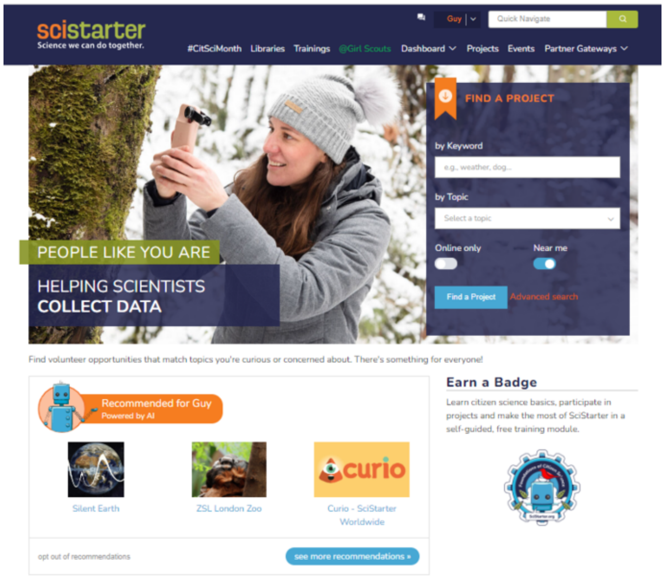
doi:10.3233/FAIA230646

Online Evaluation of Tail Project Boosting in Citizen Science

Amit Sultan[a;*], Avi Segal[a], Guy Shani[a], Darlene Cavalier and Kobi Gal[a+b]

[a]Ben-Gurion University of the Negev
[b]University of Edinburgh
ORCiD ID: Avi Segal https://orcid.org/0000-0003-0915-5730, Guy Shani https://orcid.org/0000-0003-4131-0382

Abstract. In citizen science, regular people provide invaluable information by contributing to scientific projects. Citizen science platforms, such as SciStarter, provide easy access to numerous such projects. Often, users contribute mainly to a relatively small set of popular projects, while it is difficult for many projects to draw the attention of users. Thus, increasing the contribution of users to such low-popularity projects may increase scientific and societal impact. In this paper, we explore the power of a recommender system to draw attention to less popular projects. Standard use of recommendation systems often leads to limited exposure of less popular (tail) projects. We thus propose a re-ranking approach based on "lift boosting," which uses the statistical lift measure to enhance the exposure of tail projects. By combining lift and traditional relevance measures, our method re-ranks the recommendation list to emphasize projects that are both relevant to the user while also have a high lift value. We implement our approach on SciStarter, one of the biggest citizen science platforms on the web. We conduct an online experiment involving over 2000 real users. Our results show a positive shift towards less popular projects without compromising overall contribution rates. This work demonstrates the potential of our lift-boosting method for promoting the discovery of tail projects in citizen science platforms, thereby fostering a more diverse range of scientific contributions.



Find volunteer opportunities that match topics you're curious or concerned about. There's something for everyone!

Learn citizen science basics, participate in projects and make the most of SciStarter in a self-guided, free training module.

Figure 1. The SciStarter homepage shows the project finder (top right), and the personal recommendation widget (bottom).

1 Introduction

In citizen science, scientists create projects that require the help of the public in tasks such as data collection and analysis, allowing the general public to participate in the scientific process. Citizen science is hence a collaborative approach to scientific research that involves the active participation of non-professional individuals, typically volunteers, in data collection, analysis, and dissemination of scientific findings [8]. Citizen science projects often focus on environmental monitoring, and public health, among other areas [27].

The importance of citizen science is multifaceted. First, it enables the collection of large-scale data over vast geographical regions and extended periods, which is often challenging for traditional scientific methods [13]. This helps to fill knowledge gaps and provide essential information for decision-making processes, such as policymaking and resource allocation. Second, citizen science fosters public engagement in science, and scientific literacy [11], developing a deeper understanding of scientific concepts and methodologies.

Citizen science platforms provide a gateway to citizen science projects, hosting a collection of projects, and allowing users for easy access to many projects. Indeed, the prevalence of active involvement in online citizen science platforms has been rapidly increasing [15, 22]. One of the popular platforms for citizen science is SciStarter[1]. SciStarter offers a variety of projects for users to participate in and contribute to, with over 180 affiliated projects on the website platform. In the majority of SciStarter projects, volunteers collect, categorize, identify, annotate, and label data, which is then combined and examined to derive scientific findings. These activities can be either indoor, necessitating only internet access for completion, or outdoor, involving participants in physical tasks such as capturing images of wildlife, natural surroundings, and more.

An important goal of SciStarter is to increase the number of people that participate in multiple projects that it hosts. However, it has been observed that while participants often engage with several projects

[*] Amitsultan@gmail.com

[1] https://scistarter.org/

on citizen science platforms, they consistently contribute mainly to a select few on a regular basis.

The SciStarter platform hence provides several ways for users to discover new projects. Figure 1 shows the SciStarter homepage, featuring a Project Finder tool allowing interested users to search for additional projects to participate in based on the project properties.

Another popular method to expose users to new projects is through a recommender system — a system that actively suggests projects to users [17, 24]. Indeed, SciStarter currently employs a recommendation system [6]. This system aims to encourage users to interact with additional projects on the platform, thereby increasing traffic and engagement. To increase user's acceptance, the system computes personalized recommendations tailored to their preferences and interests. The system employs a hybrid method combining user-based and content-based approaches to generate recommendations and has been operational since December 2019.

In many recommendation tasks, such as in e-commerce, or in content applications, such as Facebook or YouTube, the main goal of recommendations is to provide additional interesting items for the user, in order to increase the overall sales, or to keep the user engaged. In SciStarter, as we explained above, another goal is to increase the participation of users in projects. As in many other domains, in SciStarter, some projects are much more popular, drawing many users, while other projects are less popular. This phenomenon is known as the popularity bias [2, 18, 30]. Many recommendation systems hence end up recommending more popular items, as these recommendations are more likely to be accepted by the user.

In the context of citizen science, on the other hand, an important objective is for all projects to receive traffic, promoting overall scientific advancement. Often, in the interest of promoting science, it is much more important to draw users to the less popular projects, than to add another user to very popular projects, following the law of diminishing returns. Thus, a recommendation system that optimizes the probability that the user will accept the recommendation, is perhaps not the best choice in this domain.

The popularity bias may be reduced by showing to users recommendations of less popular projects, that are relevant for them. Computing such recommendations can be achieved by combining the user-project relevance score that is computed by the recommendation algorithm, with a statistical method that prefers less popular projects, such as lift. Indeed, in a previous paper, a lift-boosting reranking approach was shown to provide good recommendation lists in offline studies [30]. However, previous research has not evaluated how lift-boosting affects the behavior of users in SciStarter.

This paper focuses on an online evaluation of the effects of lift-boosting on users and projects in SciStarter. We conduct an online study, where some users received recommendations from the original, relevance-based system (cohort 1), while other users received recommendations from the lift-boosting method (cohort 2). We compare the behavior of users with respect to the acceptance of recommended projects between the two cohorts. To do so, we measure well-known metrics in recommendation system literature, such as click-through rate (CTR) and hit-rate.

Our results show that our lift-boosting re-ranking method increases users' exposure to less popular projects without reducing overall CTR. That is, users are as likely to click on a recommended project produced by our lift-boosting method, as with the original, relevance-based methods. Additionally, lift-boosting presents many more projects that are less popular, increasing their visibility.

2 Background

Citizen science involves the participation of individuals in scientific research by gathering, classifying, transcribing, or examining scientific data [8]. Citizen science platforms provide access to a multitude of diverse projects, all depending on the efforts of volunteers to advance the scope of scientific understanding. Citizen science offers considerable scientific, educational, and societal advantages. It introduces novel methods for the public to participate in scientific research, allowing volunteers to collaborate in data monitoring and data gathering initiatives without making strong assertions.

Recommender systems are frequently employed by e-commerce websites to provide personalized product recommendations, tailored to their customers' preferences [26]. These systems often analyze users by examining their historical data. Through these methods, recommender systems generate personalized suggestions of potential items that are specifically designed to cater to the user's preferences.

Collaborative filtering (CF) is a widely utilized technique in the realm of recommender systems, primarily employed to generate personalized suggestions for users based on their historical preferences and those of similar users [14]. The foundation of this approach lies in the assumption that individuals with similar tastes in the past will continue to exhibit similar preferences in the future. Collaborative filtering can be divided into two main categories: user-based and item-based approaches. User-based collaborative filtering involves identifying users who share comparable interests with the target user and recommending items that similar users have enjoyed [25]. Both approaches have proven to be effective in a variety of domains, such as e-commerce, online media platforms, and social networks, enhancing user experiences by offering tailored recommendations.

The SciStarter platform contains an active recommendation system, which employs a hybrid approach, combining both user-based and item-based methods [6, 12, 1]. The system utilizes matrix factorization (MF), a widely recognized technique, to compute user-based scores, specifically through Bayesian Personalized Ranking (BPR). In addition to user-based scores, content-based scores are determined using various project attributes, including topics, locations, and project descriptions. This integrated approach enables the generation of more accurate and relevant recommendations for users.

Bayesian Personalized Ranking (BPR) is a widely recognized matrix factorization technique that represents the relevance between users and projects in a latent space, characterizing both entities. Given the latent space size, k, for users and projects, the two decomposed matrices $P_{U,k}$ and $Q_{I,k}$ are derived by minimizing the discrepancy between the approximated matrix $\hat{M} = PQ^T$ and the original matrix M. Consequently, a set of vectors, $\vec{u}$ for user u and $\vec{p}$ for project p, is produced, and the relevance score can be computed:

$$\hat{r}MF = \vec{u} \cdot \vec{p} = \sum i = 1^k \vec{u}_i \cdot \vec{p}_i \tag{1}$$

The content-based approach utilizes various information about projects on the platform to generate relevant recommendations. This method takes into account factors such as project popularity, geographic proximity to users, and project tags (e.g., space, wildlife). The cosine similarity between two projects, i and j, is:

$$sim(i, j) = \frac{|F_i \cap F_j|}{|F_i| * |F_j|} \tag{2}$$

In this equation, F_i represents the set of features for project i. The recommendation score for project i, denoted as $\hat{R}^{CB}_{u,i}$, is calculated by summing the similarities between project i and all projects in $Proj_u$, the set of projects that u has previously contributed to:

$$\hat{R}^{CB}u, i = \Sigma j \in Proj_u sim(i, j) \tag{3}$$

In the context of recommendation systems, it is often the case that the distribution of user selection of items follows the power law distribution. That is, the amount of users who select an item rapidly decreases, where a relatively small set of items, often called "head" items, are highly popular, selected by many users, while the majority of the items, the "tail" items, are much less popular receiving little user attention. The term head items often refers to the top 20%, while the tail items capture the bottom 80%. This division is related to the Pareto Principle, a concept that is frequently observed in various fields of study [23].

One of the challenges associated with recommendation systems is the popularity bias, which is closely related to the division of head and tail items. Popularity bias occurs when the system predominantly recommends popular items, or head items, to users, while neglecting less popular items, or tail items. Thus, popular items are getting much more visibility, and hence, more users choose them, while tail items are rarely presented to users, further decreasing their chance of getting user's attention.

3 Related Work

The problem of popularity bias is well-known in recommender systems literature. Many recommendation algorithms often rank higher items with greater popularity [2, 18]. In many applications, this phenomena is not necessarily problematic. For example, an e-commerce website may be mainly interested in increasing sales, regardless of whether it sells popular or unpopular items. Content websites, such as YouTube or Facebook, are primarily interested in increasing user engagement, not in enhancing traffic to less popular videos.

Many methods were suggested for debiasing of popular items. Such methods attempt to increase the portion of recommendations of tail items.

The Opinion-Based Collaborative Filtering (OBCF) [33] method aims to reduce the popularity bias. The proposed method is designed to enhance recommendations by assigning weights to items based on their influence on the user model in relation to their popularity. Building on the foundations of traditional user-based collaborative filtering, OBCF incorporates weighting functions to evaluate user similarities effectively. They run offline experiments using the MovieLens dataset, and the proposed methods were tested and evaluated using NDCG, demonstrating higher accuracy and diversity.

In another study, the impact of social network effects on popularity bias is investigated [9]. The authors identify two types of biases: discovery bias, which pertains to the sources through which users discover items, and decision bias, which often leads users to be more inclined to rate items they like as opposed to those they dislike. To examine the relationship between the efficacy of popularity and various social behavior configurations, the authors develop a probabilistic model and conduct simulations. The experiments collect social network data from Facebook, similar in scale to MovieLen 1M, and use it to simulate ratings. They run simulations, demonstrating that popular items can boost perceived precision.

Explicit Query Aspect Diversification (XQUAD) is a method that originated from information retrieval and is used for result diversification, especially in web search engines. This approach was adapted by Himan [3] to recognize the difference among users in their interest in tail items. The approach suggested is a re-ranking approach, taking the output from a given recommendation system which generates a score r_i for each potential project i, and re-ranking these scores to manage popularity bias. Unlike our proposed method, this re-ranking approach is a temporal approach based on the xQuAD diversification algorithm which aims to improve tail recommendation over time. In their experiments, the authors also used the MovieLen dataset, along with another dataset called the Epinions dataset, which is gathered from a consumer opinion site where users can review items.

UPD (User Popularity Deviation) [4] emphasizes the shortcomings of existing metrics in evaluating the effectiveness of popularity bias mitigation. The authors note the lack of attention given to user-centered evaluations of this bias, which can lead to different impacts on users depending on their interest levels in popular items. In their experiments, they assess various methods for mitigating popularity bias and introduce a new metric to address these limitations. Their proposed approach, called Calibrated Popularity (CP), is a re-ranking method inspired by [29]. It aims to balance the distribution of recommended items between popular and less-popular items. For instance, if a user consumes twice as many popular items, the recommendation set should maintain the same ratio between popular and less-popular items. They demonstrate in an offline study the variety between users, and how popularity debiasing methods may affect different users with different tastes. The proposed UPD metric may not be suitable for certain online applications. For example, in our citizen science application, fewer than 10% of users participate in more than 10 projects [12], posing a challenge in accurately calculating UPD.

The concept of fairness is also considered with respect to popularity bias. In some applications, it may be considered unfair to avoid recommending less popular items.

Smith's work [28] delves into the issue of fairness, specifically examining its meaning within the context of recommendation systems where multiple stakeholders are involved. Their user study involved 30 participants who were asked to share their thoughts on fairness in recommendation systems. Similarly to this work, we also encounter multiple stakeholders, such as users, project administrators, and SciStarter platform representatives. The participants' views echoed the sentiment that an algorithm must be unbiased to be considered fair. One participant, for example, highlighted the "wealthy getting wealthier" issue, referring to the phenomenon where popular projects receive additional exposure. They show, in offline experiments, that their approach is capable of recommending unique long-tail items while maintaining comparable ranking accuracy.

An intriguing study in AI fairness is presented by [31]. In their research, they conducted both offline and online evaluations using a debiased recommender system. The offline results were gathered using the Facebook dataset, demonstrating that the debiased recommender was fairer without compromising prediction accuracy. Although the offline results were promising, the online study, conducted with 200 students from a U.S. university, indicated that users on average preferred the original biased system over the debiased one. While the paper primarily focuses on gender bias, its findings suggest that for some applications, one could argue that fairness is not necessarily beneficial to the user.

An increasing number of researchers work on GANs in several domains including recommender systems [7, 10]. GANs often focus on maximizing user utilities rather than fairness between recommending popular items to non-popular items. FairGAN authors [20] suggested approach aims to tackle that by balancing the learning process of a GAN by giving the model fairness signal in the training process. The approach has three phases, (i) training a "Ranker" component which aims to maximize user utility by capturing user preference; (ii) training a "Controller" to capture the current item distribution; (iii) controlling fairness by generating a fairness signal and feeding it to the

Ranker from phase 1. In their experiment, the authors [20] compare various GANs, including 'regular' GAN, FairGAN, and NaiveFairGAN. They show, in offline experiments that producing recommendations that are more fair, that is, providing recommendations over less popular items, can increase the accuracy.

Another study explores the evaluation of unfairness in popularity bias [32]. This research assesses the popularity bias issue from the users' point of view and seeks to address it by treating items as essential stakeholders. The authors pinpoint five critical discriminative features based on users' rating behavior and examine the connections between these features and users' original preferences for item popularity, as well as the possible unfairness concerns arising from popularity bias in recommendations. Their offline experiments involve various state-of-the-art methods, which are tested across four different datasets, including MovieLens. For each dataset, the authors classify users into three groups based on the previously defined features of their rating profiles. In their findings, they discovered that the popularity propensities of individual users are strongly correlated with the majority of the suggested features, leading to varying trends. Specifically, they mention user groups who interact frequently, are 'picky', and are difficult to predict. These users often receive unfair and less accurate recommendations that do not entirely fulfill their needs, despite being important users.

Several studies attempted to measure the user's response to recommending non-popular items.

In the study of item familiarity [16], the authors conduct a user study where they expose participants recruited on a crowd-sourcing platform to system-provided recommendations, based on the MovieLen dataset. Following that, and based on meta-data information, participants had to rate the movies and assess the recommendations lists as a whole regarding additional aspects such as diversity, transparency, or surprise. The results of this study showed that users found the best recommendations to be non-personalized recommendations of popular items. One of their analysis revealed a correlation between item familiarity and user acceptance. We observe similar trends in our study, as users, even when shown recommendations for less popular projects, still, click more often on the more popular projects.

A method for discovering novel recommendations using a graph was introduced by [19]. This graph-based recommender system constructs a highly-connected, undirected graph by utilizing only positively-rated items as nodes and positive correlations as edges. By using both entropy and the linked items in the graph, the system is able to find both novel and relevant recommendations. Obviously, there is a clear (negative) correlation between novelty and popularity, where popular items tend to be less novel. A user study was conducted on a custom-built website to evaluate their graph-based recommendation system. At the custom website, each algorithm output was a top-5 recommendation presented to the user in a randomized order. The results showed that the suggested algorithm provided novel recommendations, without impacting relevance too much.

To conclude, while many previous studies suggested methods for managing the popularity bias, and there were some attempts to provide user studies that measure the effect of such methods on people, we did not find any previous study measuring the effect of popularity debiasing methods on a large number of users in a real application.

4 Lift-based Reranking of Low Popularity Projects

We now describe the method for improving recommendations of less popular projects [30]. This method first computes a set of projects that are predicted to be relevant to the particular user using our hybrid engine, which combines a collaborative filtering algorithm, and a content-based approach. Then, the projects in the relevant set are ranked such that lower-popularity projects take precedence.

First, we describe the necessary notations. Let U be the set of users and P be the set of projects. We denote by P_u the set of projects that user u has taken part in, and U_p is the set of users who have contributed to project p.

In data mining and associative rule learning, the lift measure [5, 21] measures the increase in the posterior probability of event j given event i, versus the prior probability of event j. Alternatively, lift measures the increase in the probability of the occurrence of event j when it coincides with the occurrence of event i, relative to the probability of observing events j and i independently:

$$Lift(j,i) = \frac{p(j \mid i)}{p(j)} = \frac{p(j \cap i)}{p(j) \cdot p(i)} \tag{4}$$

In practice, we do not know the exact prior and posterior probabilities of items, and we hence use a maximum likelihood estimator over the observed participation in these projects in our dataset. Hence, we compute the empirical lift:

$$Lift(j,i) = \frac{|U_j \cap U_i| \cdot |U|}{|U_j| \cdot |U_i|} \tag{5}$$

A well-known problem with the empirical lift is that for projects with only a handful of visits, the maximum likelihood estimator often grossly underestimates the true prior. For such an unpopular project i, the empirical $Lift(i,j)$ is estimated to be very high for any other project j. It is hence common to use a threshold on the number of observations, and if the project was visited less than this threshold, set its lift to 0, denoting an unknown value. This nullifies the probability that this project will be shown to the user (see equation 6).

As previously mentioned, the $Lift(p_j, p_i)$ measures the likelihood that a user will engage in a new project, p_j, given their prior participation in the project p_i. We define the lift of a recommended project p_j for a target user u as the median lift between p_j and all projects in which u has previously participated:

$$Lift(p_j, u) = median_{p_i \in P_u} Lift(p_j, p_i) \tag{6}$$

The lift value for a recommended project p_j targeting user u will be higher if more users have participated in both p_j and another project that user u has been involved in before, and if the overall popularity of project p_j is low.

Our method is a combination of lift and relevance measures to rank the recommendation list. Our system uses a base recommendation algorithm, denoted as a, which takes a user u and a project p_j as inputs and produces the relevance score $\hat{r}_a(u, p_j)$, indicating the relevance of project p_j to user u.

More specifically, in the context of the SciStarter platform, a is the hybrid collaborative filtering, content-based, method.

To generate a list of recommended projects for user u, we first use a to provide a relevance score for all projects that u has not yet participated in. Then, we order the list of relevant projects by decreasing $Lift(u, p_j) \times \hat{r}_a(u, p_j)$. This approach emphasizes projects that are relevant to user u and have a high lift value for them. We refer to this approach as "lift boosting".

5 Online Evaluation

We now discuss the main contribution of the paper — evaluating how our lift-boosting reranking method affects project participation in the

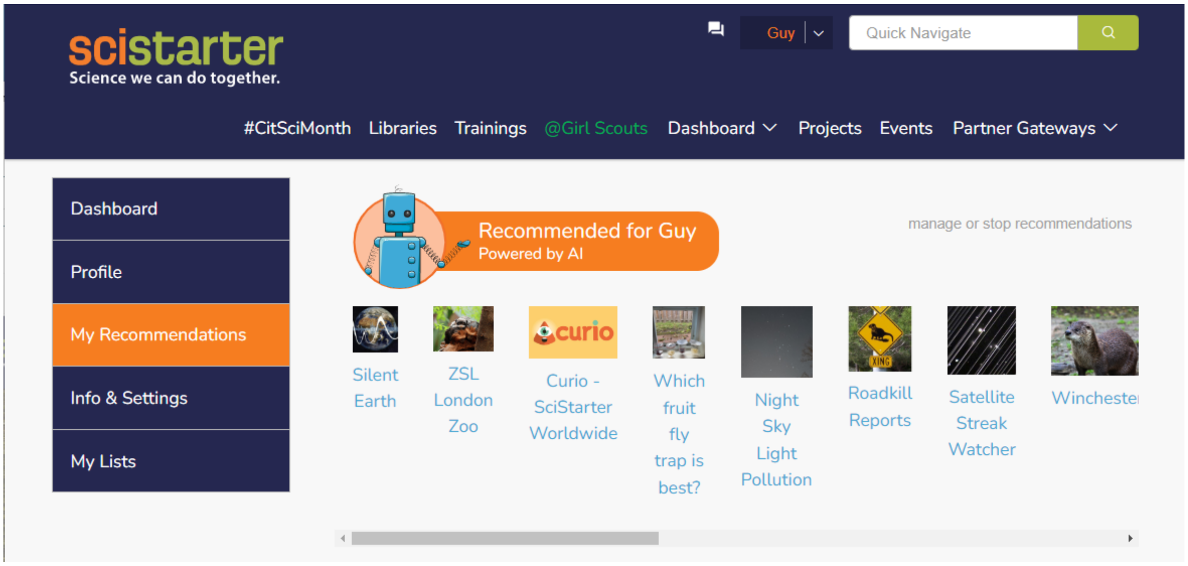


Figure 2. A snippet of a set of recommendations, shown to user "Guy" when clicking the "see more recommendations" button at the homepage.

SciStarter platform. We conducted an online study, where some users received recommendations from the reranking method, while other users received the recommendations ranked by the existing SciStarter recommender system, based only on the relevance to the user.

Because projects are hosted on websites and apps external to SciStarter, APIs between projects and SciStarter are used to validate participation. However, our recommendation system uses data from SciStarter's clickstream and is only exposed to the activity within SciStarter platform (not the data from the APIs). Hence, in many cases, we only know whether a user has observed, liked, saved, shared, or clicked to visit the project's website from the project page on SciStarter, not whether the user has contributed to the project. This data is available to the SciStarter team through an Because of this, recommendations displayed to logged in users are based on clickstream data not API/participation data.

5.1 Displaying Recommendations in SciStarter

SciStarter provides a web platform that enables users to participate in various projects. These projects may either be hosted online or require physical attendance, with some managed directly within SciStarter, called affiliate projects, and others directing users to external sources to complete their contributions. SciStarter offers several ways for users to discover new projects. Users can use the Project Finder tool to search for projects based on properties such as topic or location. Personalized recommendations are presented immediately in the homepage. Then, if users click the "see more recommendations" they are transferred to a page showing additional recommended projects (Figure 2). In this page users can see and click any recommended project, with a thumbnail image for each project alongside the recommended project name.

5.2 Case Study

To demonstrate the effects of our re-ranking method, we present an example of the recommended projects generated for two users on the SciStarter platform – one user who received relevance-based recommendations, and another user that received lift-boosting recommendations. Table 1 shows the top-10 recommendations produced by the relevance recommendation algorithm, and the top 10 recommendations produced by lift-boosting re-ranking algorithm for both users, although each user received just one type of recommendations.

The Relevance-Based user was presented with relevance-based recommendations (grayed column, top-left). We also show for this user which Lift-boosting recommendations they could have received (top-right). In the same manner, the Lift-Boosting user was presented with the lift-based recommendations (grayed column, bottom-right). We also show which relevance-based recommendations that user could have received (bottom-left).

The popularity column for each recommendation type presents the project's popularity rank, as well as whether it is a head project (H), or a tail project (T). The recommendations column for each recommendation type shows the results of each method, ordered by decreasing score.

As can be seen, in the lift-boosted recommendations, many tail items are promoted to the top of the list. On the other hand, the Stall Catchers project, by far the most popular project in SciStarter, was pushed out of the recommended projects in the lift-boosted list.

The bolded recommendations indicate that the specific user had accepted (clicked on) the recommended project during the study. As seen in the table, the particular user that was assigned to the lift-boosted cohort clicked on two tail projects, while the user in the relevance-based cohort, did not click on any tail item. Of course, the relevance-based user was not shown the lift-boosted recommendations (top-right), which would have exposed her to many tail projects, as demonstrated in the table. Thus, this user may have been unaware of these tail projects and missed the option to contribute to them. Clearly, one cannot expect users to choose less popular projects if they are unaware that these projects exist.

Popularity Rank	Relevance-based Recommendations	Popularity Rank	Lift-boosting Recommendations
Relevance-Based User			
4 (H)	**Globe at night**	80 (T)	Dragonfly Swarm Project
6 (H)	The neureka project	47 (T)	never home alone
12 (H)	**globe observer: Clouds**	53 (T)	Herp Mapper
1 (H)	**Stall Catchers**	**12 (H)**	**globe observer: clouds**
61 (T)	Roadkill reports	50 (T)	globe observer: mosquito habitat mapper
10 (H)	Phylo	61 (T)	roadkill reports
24 (H)	Squirrel mapper	12 (H)	I See Change
5 (H)	Crowd the tap	48 (T)	Satellite Streak Watcher
33 (H)	Globe Observer: Land Cover	42 (T)	questagame
22 (H)	Osa Camera Trap Network	70 (T)	Curio
Lift-Boosting User			
1 (H)	Stall Catchers	47 (T)	Never Home Alone
2 (H)	**Project Squirrel**	**37 (T)**	**Caterpillars Count!**
4 (H)	Globe at Night	41 (T)	Land Loss Lookout
36 (H)	Silent earth	**2 (H)**	**Project Squirrel**
12 (H)	globe observer: Clouds	**46 (T)**	**Tree Snap**
12 (H)	**I See Change**	4 (H)	Globe At Night
47 (T)	Never Home Alone	5 (H)	Crowd The Tap
14 (H)	ant picnic	48 (T)	Satellite Streak Watcher
18 (H)	Globe Observer: Trees	**12 (H)**	**I See Change**
44 (T)	zoombee watch	38 (T)	Disk Detective

Table 1. Case study — recommendation lists for two example users in our study, relevance-based and lift-boosting. H, T denote head and tail items.

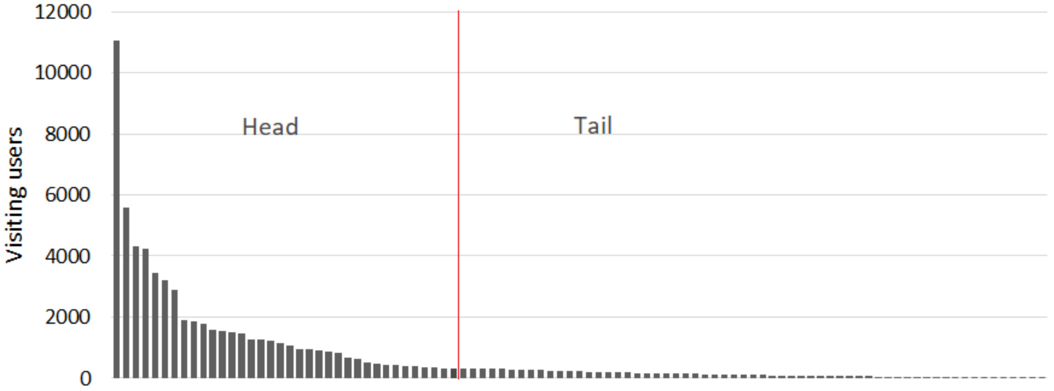


Figure 3. Project popularity — amount of unique users who participated in a project. We show here only the 97 projects (of 184 projects in total) that had at least 20 participants.

5.3 Study Procedure

Throughout the course of the study, which took place between March 18, 2023, and May 05, 2023, a total of 2101 of registered repeated users who logged into SciStarter were randomly assigned to one of two cohorts. Cohort 0 (relevance-based recommendations) contained 1043 users, while cohort 1 (lift-boosted recommendations) contained 1058 users. We also computed the average user participation in projects during the study period. Users participated on average in 1.18 projects, with a standard deviation of 1.84, showing that some users participate in several projects, while many participate only in a single project. There were no statistically significant differences between the cohorts with respect to project participation.

Figure 3 shows the distribution of users who contributed to projects. As can be seen, the data follows the power law distribution, as expected, with the most popular project receiving contributions from about 11000 unique users, 6 projects receiving 2000-5500 unique users, and 12 projects receiving 1000-2000 unique users. The 20% cutoff that we set for the head-tail split is at 313 users. We stress that the head-tail split is not used in the algorithm, but rather only for ease of exposition of the results.

As we explained above, lift computations are grossly inexact when the amount of users that choose a project is very low. In our experiments, we set this value empirically to 20 users. That is, we set the lift of projects that receive less than 20 users to 0, and hence, they do not get recommended in the lift-boosting method. Figure 3 does not show the 87 projects that received less than 20 users.

5.4 Results

To evaluate the effectiveness of our lift-boosted recommendation set, we analyze the users' interactions with the recommended projects on the platform, particularly their engagement with less popular projects (Tail) and popular projects (Head).

First, of the 2101 users in our study, only 575 (27.36%) clicked on at least one recommended item. This is not very surprising, because many users treat SciStarter as the entry point to a single project that they participate in, and are not currently interested in joining other projects. This result is almost identical (no statistically significant difference detected) for the two cohorts — 26.55% for users who received relevance-based recommendations, and 28.16% for users who received the lift-boosting recommendations. This shows that, at the very least, users who received the lift-boosting recommendations were not less likely to click on a recommended item.

Table 2 compares the two cohorts' click-through rate (CTR), which is the amount of clicked recommendations of all presented recommendations. As seen in the table, the CTR between the two cohorts is comparable (no statistically significant difference detected). The CTR value of less than 3% click rate is customary in the field and is not surprising, as in this application type it is unlikely for a user to click on more than a single item in a list.

We hence also report the hit rate — the portion of recommendation lists where at least one item was clicked. As can be seen in the table, the hit rate is comparable for the two conditions and is about 30% (no statistical significant difference detected). As the lists can be long (see Figure 2), a single click in a list corresponds to a relatively low CTR. In other words, these results demonstrate that despite lift-boosting by presenting users with recommendations that may be perceived as less popular, they remain sufficiently pertinent and relevant such that there is no decrease in the users' overall interest.

Delving deeper into the hit rate, we split the hits between lists where the clicked item was a head project (denoted Head Hit Rate), and lists where the clicked item was a tail project (Denoted Tail Hit Rate). As can be seen, lift-boosting causes people to click more often on tail items, although obviously, people still click twice as much on head projects. This is not surprising — popular projects are popular because people like them. It is not reasonable to expect that people would not choose popular projects. However, our lift-boosting method improves the number of tail projects that got the user attention.

Cohort	CTR	Hit-Rate (HR)	Tail HR	Head HR
Relevance-based	2.81%	29.2%	7.1%	22%
Lift-boosted	2.62%	30.43%	10.38%	20.9%

Table 2. CTR and Hit-Rate for relevance-based and lift-boosted recommendations.

Figure 4 shows the ratio of clicked projects for each cohort, separated into head and tail projects. While it seems that there is an increase in clicks of tail items in the lift-boosting condition, the differences are not statistically significant. We hypothesize that as additional data would be collected, a statistically significant improvement in users clicks on less popular projects may be observed.

We end our analysis by analyzing the portion of head and tail items presented to users in the two cohorts (Figure 5). As can be seen, in the relevance-based cohort, only 26.14% of the recommendations correspond to tail items. In the lift-boosted cohort, on the other hand, 59.23% of the recommendations are for tail projects. This difference is statistically significant (chi-squared test, p<0.0001). Still, the lift-boosting maintains some recommendations for head items. This is expected and desirable — as we said above, popular projects are often liked by the user, and we would not want to bar them from the user altogether. However, as Figure 5 shows, the lift-boosting method significantly increases the visibility of tail projects.

To conclude, the results above show that our lift-based method provides much more visibility for less popular projects, at the expanse of the more popular projects. This is done while preserving the overall Click Through Rate in the system.

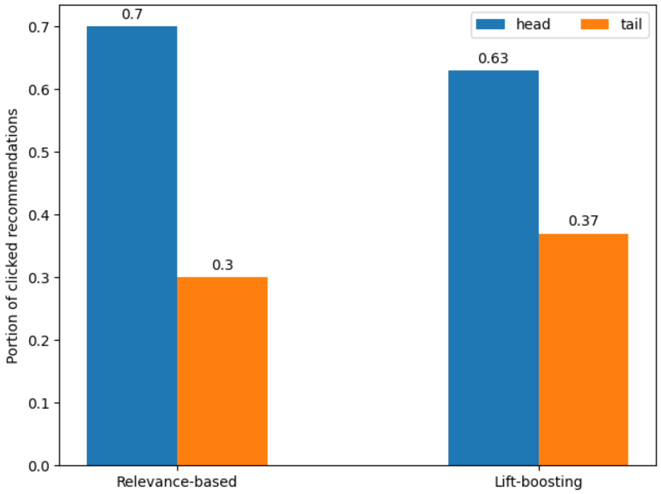


Figure 4. Portion of recommended items clicked by users, compared based on the popularity of the items.

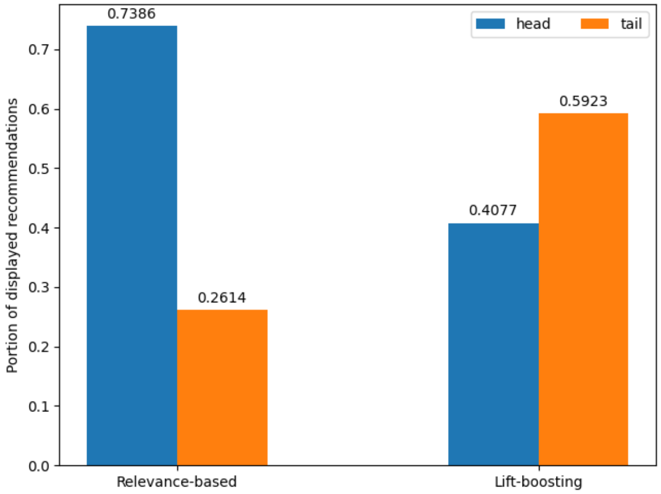


Figure 5. Recommended projects shown to users, compared based on the popularity of the projects.

6 Conclusion

In this paper we described an online study to evaluate the effects of using lift-boosting to rerank recommended citizen science projects in the SciStarter platform. Our approach increased the visibility of less popular (tail) projects, allowing them to receive more traffic, and helping them to promote their research. We showed that while lift-boosting increases the visibility of tail projects, it does not reduce user's interest, as reflected by the similar CTR between users who receive recommendations based solely on personal relevance, and users who received lift-boosted recommendations. These results demonstrate the potential efficacy of lift-boosting for increasing exposure of tail items in recommendation based systems.

In future research, we intend to explore user interface modifications that would further draw the attention of users to tail projects, for example, by adding a banner on the project thumbnail with a message such as “This project needs your help”. Such cues should be well thought of, so as to not cause negative attitudes towards the projects. Additionally, we plan to run larger scale experiments, to further investigate the impact of lift-boosting on the CTR of head and tail projects. Finally, we intend to investigate lift-boosting approaches in other recommendation based systems outside the citizen science domain.

7 Acknowledgements

This work was supported in part by the European Union Horizon 2020 WeNet research and innovation program under grant agreement No 823783. We also would like to thank very much to Daniel Arbuckle for his assistance in the experiment design and programming of the SciStarter platform.

References

[1] ‘Intelligent recommendations for citizen science’, **35**.

[2] Himan Abdollahpouri, ‘Popularity bias in ranking and recommendation’, in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES ’19, p. 529–530, New York, NY, USA, (2019). Association for Computing Machinery.

[3] Himan Abdollahpouri and Robin Burke. Reducing popularity bias in recommendation over time, 2019.

[4] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward Malthouse, ‘User-centered evaluation of popularity bias in recommender systems’, in *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP ’21, p. 119–129, New York, NY, USA, (2021). Association for Computing Machinery.

[5] R.J. Bayardo, R. Agrawal, and D. Gunopulos, ‘Constraint-based rule mining in large, dense databases’, in *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pp. 188–197, (1999).

[6] Daniel Ben Zaken, Avi Segal, Darlene Cavalier, Guy Shani, and Kobi Gal, ‘Generating recommendations with post-hoc explanations for citizen science’, in *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP ’22, p. 69–78, New York, NY, USA, (2022). Association for Computing Machinery.

[7] Homanga Bharadhwaj, Homin Park, and Brian Y. Lim, ‘Recgan: Recurrent generative adversarial networks for recommendation systems’, in *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys ’18, p. 372–376, New York, NY, USA, (2018). Association for Computing Machinery.

[8] Rick Bonney, Caren B. Cooper, Janis Dickinson, Steve Kelling, Tina Phillips, Kenneth V. Rosenberg, and Jennifer Shirk, ‘Citizen Science: A Developing Tool for Expanding Science Knowledge and Scientific Literacy’, *BioScience*, **59**(11), 977–984, (12 2009).

[9] Rocío Cañamares and Pablo Castells, ‘Exploring social network effects on popularity biases in recommender systems’, in *RSWeb@RecSys*, (2014).

[10] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. Generative adversarial user model for reinforcement learning based recommendation system, 2020.

[11] Sarah A. Collins, Michelle Sullivan, and Heather J. Bray, ‘Exploring scientists’ perceptions of citizen science for public engagement with science’, *JCOM*, **21**(07), A01, (2022).

[12] Na’ama Dayan, Kobi Gal, Avi Segal, Guy Shani, and Darlene Cavalier, ‘Intelligent recommendations for citizen science’, *CEUR Workshop Proceedings*, **2697**, (January 2020). Publisher Copyright: Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)..; 2020 Workshops on Recommendation in Complex Scenarios and the Impact of Recommender Systems, ComplexRec-ImpactRS 2020 ; Conference date: 25-09-2020.

[13] Janis L. Dickinson, Benjamin Zuckerberg, and David N. Bonter, ‘Citizen science as an ecological research tool: Challenges and benefits’, *Annual Review of Ecology, Evolution, and Systematics*, **41**(1), 149–172, (2010).

[14] Jon Herlocker, Joseph Konstan, Loren Terveen, John C.s Lui, and T. Riedl, ‘Evaluating collaborative filtering recommender systems’, *ACM Transactions on Information Systems*, **22**, 5–53, (01 2004).

[15] Aisling Irwin, ‘No phds needed: how citizen science is transforming research’, *Nature*, **562**, 480–482, (10 2018).

[16] Dietmar Jannach, L. Lerche, and M. Jugovac, ‘Item familiarity effects in user-centric evaluations of recommender systems’, **1441**, (01 2015).

[17] Yehuda Koren, Robert Bell, and Chris Volinsky, ‘Matrix factorization techniques for recommender systems’, *Computer*, **42**(8), 30–37, (2009).

[18] Dominik Kowald and Emanuel Lacic, ‘Popularity bias in collaborative filtering-based multimedia recommender systems’, *arXiv preprint arXiv:2203.00376*, (2022).

[19] Kibeom Lee and Kyogu Lee, ‘Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items’, *Expert Systems with Applications*, **42**(10), 4851–4858, (2015).

[20] Jie Li, Yongli Ren, and Ke Deng, ‘Fairgan: Gans-based fairness-aware learning for recommendations with implicit feedback’, in *Proceedings of the ACM Web Conference 2022*, WWW ’22, p. 297–307, New York, NY, USA, (2022). Association for Computing Machinery.

[21] P.D. McNicholas, T.B. Murphy, and M. O’Regan, ‘Standardising the lift of an association rule’, *Computational Statistics & Data Analysis*, **52**(10), 4712–4721, (2008).

[22] Oded Nov, Ofer Arazy, and David Anderson, ‘Scientists@home: What drives the quantity and quality of online citizen science participation?’, *PloS one*, **9**, e90375, (04 2014).

[23] Vilfredo Pareto, *Cours d”economie politique profess’e ‘a l’Universit’e de Lausanne*, volume I, II, Université de Lausanne, 1896-1897.

[24] Michael J. Pazzani and Daniel Billsus, *Content-Based Recommendation Systems*, 325–341, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[25] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl, ‘Grouplens: An open architecture for collaborative filtering of netnews’, in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW ’94, p. 175–186, New York, NY, USA, (1994). Association for Computing Machinery.

[26] J. Ben Schafer, Joseph Konstan, and John Riedl, ‘Recommender systems in e-commerce’, in *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC ’99, p. 158–166, New York, NY, USA, (1999). Association for Computing Machinery.

[27] Jonathan Silvertown, ‘A new dawn for citizen science’, *Trends in Ecology and Evolution*, **24**(9), 467–471, (2009).

[28] Jessie Smith, Nasim Sonboli, Casey Fiesler, and Robin Burke. Exploring user opinions of fairness in recommender systems, 03 2020.

[29] Harald Steck, ‘Calibrated recommendations’, in *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys ’18, p. 154–162, New York, NY, USA, (2018). Association for Computing Machinery.

[30] Amit Sultan, Avi Segal, Guy Shani, and Ya’akov (Kobi) Gal, ‘Addressing popularity bias in citizen science’, in *Proceedings of the 2022 ACM Conference on Information Technology for Social Good*, GoodIT ’22, p. 17–23, New York, NY, USA, (2022). Association for Computing Machinery.

[31] Clarice Wang, Kathryn Wang, Andrew Bian, Rashidul Islam, Kamrun Naher Keya, James Foulds, and Shimei Pan, ‘Do humans prefer debiased ai algorithms? a case study in career recommendation’, in *27th International Conference on Intelligent User Interfaces*, IUI ’22, p. 134–147, New York, NY, USA, (2022). Association for Computing Machinery.

[32] Emre Yalcin and Alper Bilge, ‘Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis’, *Information Processing and Management*, **59**(6), 103100, (2022).

[33] Xiangyu Zhao, Zhendong Niu, and Wei Chen, ‘Opinion-based collaborative filtering to solve popularity bias in recommender systems’, volume 8056, pp. 426–433, (08 2013).