User Modeling and User-Adapted Interaction (2022) 32:839–881
https://doi.org/10.1007/s11257-022-09339-6

Fair performance-based user recommendation in eCoaching systems

Ludovico Boratto[1] · Salvatore Carta[1] · Walid Iguider[1] · Fabrizio Mulas[1] · Paolo Pilloni[1]

Received: 19 June 2020 / Accepted in revised form: 24 June 2022 / Published online: 5 August 2022

Abstract

Offering timely support to users in eCoaching systems is a key factor to keep them engaged. However, coaches usually follow a lot of users, so it is hard for them to prioritize those with whom they should interact first. Timeliness is especially needed when health implications might be the consequence of a lack of support. In this paper, we focus on this last scenario, by considering an eCoaching platform for runners. Our goal is to provide a coach with a ranked list of users, according to the support they need. Moreover, we want to guarantee a fair exposure in the ranking, to make sure that users of different groups have equal opportunities to get supported. In order to do so, we first model their performance and running behavior and then present a ranking algorithm to recommend users to coaches, according to their performance in the last running session and the quality of the previous ones. We provide measures of fairness that allow us to assess the exposure of users of different groups in the ranking and propose a re-ranking algorithm to guarantee a fair exposure. Experiments on data coming from the previously mentioned platform for runners show the effectiveness of our approach on standard metrics for ranking quality assessment and its capability to provide a fair exposure to users. The source code and the preprocessed datasets are available

✉ Ludovico Boratto
ludovico.boratto@acm.org

Salvatore Carta
salvatore@unica.it

Walid Iguider
walid.iguider@unica.it

Fabrizio Mulas
fabrizio.mulas@unica.it

Paolo Pilloni
paolo.pilloni@unica.it

[1] Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale, 72, 09124 Cagliari, Italy

at: https://github.com/wiguider/Fair-Performance-based-User-Recommendation-in-eCoaching-Systems.

Keywords User recommendation · eCoaching · Ranking · Algorithmic fairness · Running

1 Introduction

eCoaching systems support users in achieving their personal goals (Kamphorst 2017). In the context of health, they assist users in their self-care, sometimes through the promotion of physical activity routines (Klein et al. 2015). Human coaches have a key role in keeping users engaged (Boratto et al. 2017). However, keeping users engaged in the long-term is a challenging task, since a coach usually supports a lot of people.[1] In the physical activity domain, this means that after a workout session a coach should get in touch with the people they support (e.g., via a chat). In this sense, prioritizing users after their workout is key, in order to get in touch first with those who need more support (e.g., because they completed a workout with a bad performance). A lack of prioritization might have consequences that go beyond engagement and might have direct implications on the health and well-being of users, since those with the worst performances would have delayed support.

In order to fit our problem in the context of real eCoaching systems that support users in their physical activity, let us introduce *u4fit*, the platform considered in this study. *u4fit* is an innovative tool for online Personal Training that exploits the knowledge and the experience to foster users to an active lifestyle.

The u4fit eCoaching platform.

The platform is made up of a web application and a mobile client. The mobile application uses the devices' sensors to record training statistics, while the web application provides athletes with an area where they can manage their workout settings and find workout session statistics; it also serves as a dashboard for the coaches, so that they can find all the tools needed to handle requests of tailored workout plans. Figure 1 depicts the typical interaction between an athlete and a coach.

After the athlete chooses a coach and specifies their objectives and current physical skills, the coach receives the athlete's data and creates a tailored workout plan and sends it to the athlete's app. (See points 1 and 2 in the figure)

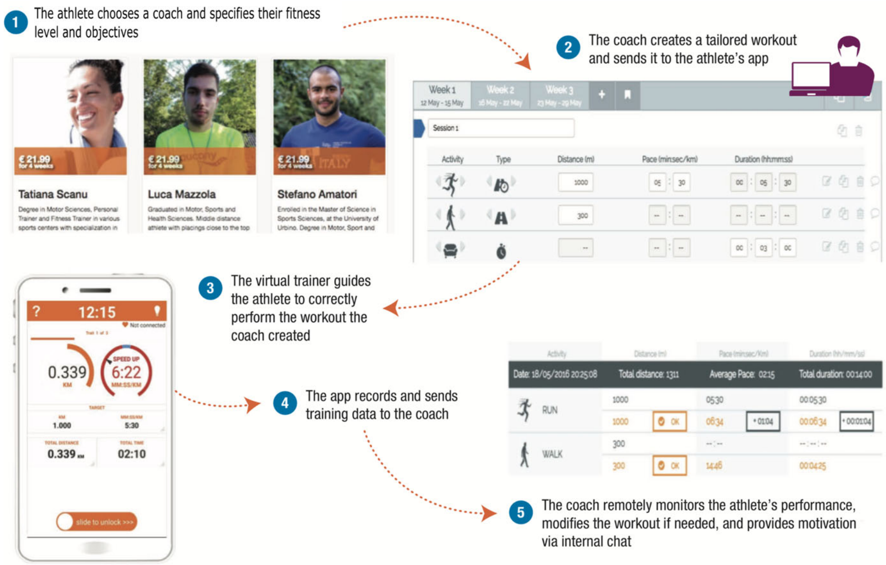
When the athlete receives the workout plan, the virtual personal trainer functionality of the mobile app guides them to correctly complete the workout, and the mobile app records training data. (See points 3 and 4 in the figure)

At the end of the workout, the coach receives training statistics and remotely monitors the athlete's performance, modifies the workout (if needed), and motivates them by means of the internal messaging system. (See point 5 in the figure).

Our contributions.

In this paper, we propose a recommender system that suggests to a coach the athletes who performed a workout, according to their performance. User recommendation

[1] In the platform considered in this paper, a coach follows on average 21.3 users.



1 The athlete chooses a coach and specifies their fitness level and objectives

2 The coach creates a tailored workout and sends it to the athlete's app

3 The virtual trainer guides the athlete to correctly perform the workout the coach created

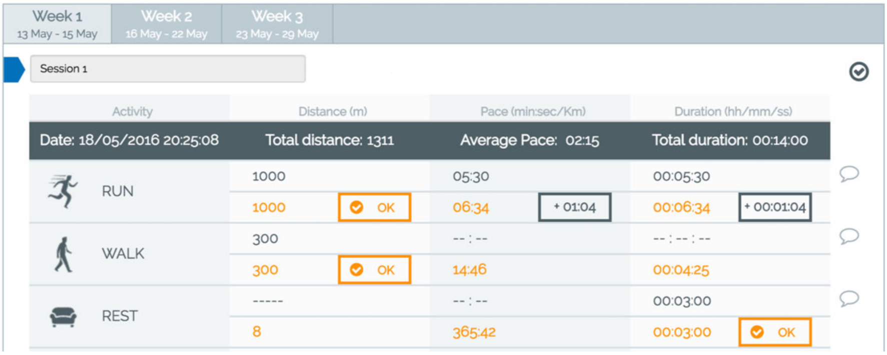
4 The app records and sends training data to the coach

5 The coach remotely monitors the athlete's performance, modifies the workout if needed, and provides motivation via internal chat

Fig. 1 User-trainer interaction flow in the *u4fit* platform

(a.k.a. contact recommendation) is usually intended as the task of suggesting one user to another, in order for them to connect (Sanz-Cruzado and Castells 2019). In our domain, an athlete can be recommended to a coach multiple times (even very frequently, according to the athlete's training schedule).

Our approach first models athletes according to their workout performance, then ranks them in ascending order of workout quality, thus suggesting first those with the worst performances. The choice of introducing a recommender system between the end of a workout and the support offered by the coach is not only motivated by the large number of athletes that a coach follows, but also by the complexity of workout results (a workout is usually composed by different activities, such as running, walking, and resting, and each activity is in turn made up of several statistics, like the speed and covered distance, as illustrated in Fig. 2), which need to be contextualized with the characteristics of the athletes (e.g., gender, age, and workout objective). With our proposal, we are offering coaches an initial filtering of the workout results, to facilitate their work. Hence, in the flow presented in Fig. 1, our solution enriches step 4. Concretely, when the app returns the workout results, it does not only provide the coach with the fine-grained results of each athlete, but the coach also sees a ranking of the athletes, thus being able to analyze first the results of those who are supposed to be more in need. The goal of this recommender system is to spare coaches the time of analyzing and contextualizing the current workout results with the previous workout sessions for each athlete and let them directly identify the athletes that need to be contacted earlier than others (the shrink in the performance of athletes w.r.t. their usual behavior could be caused by different factors that the coach needs to identify timely to eventually prevent injuries, over-training, and loss of motivation among others).



Week 1 13 May - 15 May | Week 2 16 May - 22 May | Week 3 23 May - 29 May

Session 1

Activity	Distance (m)		Pace (min:sec/Km)		Duration (hh/mm/ss)	
Date: 18/05/2016 20:25:08	Total distance: 1311		Average Pace: 02:15		Total duration: 00:14:00	
RUN	1000		05:30		00:05:30	
	1000	✓ OK	06:34	+ 01:04	00:06:34	+ 00:01:04
WALK	300		-- : --		-- : -- : --	
	300	✓ OK	14:46		00:04:25	
REST	-----		-- : --		00:03:00	
	8		365:42		00:03:00	✓ OK

Fig. 2 Workout results for an athlete. The report is divided into one tab reporting the results of a week. For each workout session performed in a given week, a summary of it is presented (date, total distance, average pace, total duration). Then, for each type of activity performed by the athlete, the performance in terms of distance, pace, and duration is presented

In this work, we model the recommendation problem as a ranking problem, since our goal is not to predict the quality of a workout with a score (rating), but to provide the coach with an effective ranking of the athletes to support, in so-called Personalized Learning to Rank approaches (Amatriain and Basilico 2015). Indeed, predicting the rating of a workout is not enough to provide a coach with effective information about the athlete, since the performance in the last workout should be contextualized with the usual behavior of the athlete, e.g., it would be much more urgent to support an athlete who does a poor workout but usually does well, than to support an athlete who performed a poor workout, but always does so (in this second case, a coach expects that the performance of that athlete would not be optimal). In a nutshell, we model athletes' workout performance by contextualizing it to their recent behavior and use this modeling to provide a personalized ranking of these athletes to the coaches.

As previously mentioned, sensitive attributes of the athletes, such as the gender, are used by our ranking algorithm. Hence, there might be the risk for the athletes who belong to a certain gender to receive a *disparate treatment*, i.e., to receive less timely support, because of an attribute that should not affect their ranking position. Hence, it is important that users receive a *fair exposure*, i.e., that their ranking positions are not affected by their gender. However, relevance estimation by itself does not guarantee fairness of exposure (Biega et al. 2018; Singh and Joachims 2018). In order to deal with this issue, we provide metrics to assess fairness of exposure and an efficient algorithm to re-rank the unfair lists.

To the best of our knowledge, in the athletic field, no one has ever developed a system able to support athletes by ranking them in a fair way, helping their human coaches prioritize who needs the most immediate support.

Although contact recommenders have been widely studied in the literature (Guy and Pizzato 2016; Sanz-Cruzado and Castells 2019), our novelty goes much beyond the application domain. Indeed, classic contact recommenders are not necessary anymore after two people connect, while a user connects to a coach *through a recommendation* multiple times (i.e., each time they perform a workout). Later in the paper, we will also highlight differences at the algorithmic level, which make our problem fundamentally new.

Specifically, our contributions can be summarized as follows:

- We present an approach to model the performance of the athletes in a running workout session;
- We introduce an algorithm to rank the athletes according to the support they need and recommend them to the coach;
- We provide, for the first time in the literature of athletic-related user recommendation, algorithms to provide fairness of exposure in the results;
- We validate our proposal on a real-world dataset collected from an eCoaching platform on standard metrics to assess ranking quality.

Paper structure. The rest of the paper is structured as follows: Sect. 2 presents related work. Section 3 presents the preliminaries, to provide foundations to our work. In Sect. 4, we introduce the dataset and our approach to workout modeling. Section 5 describes the user recommendation algorithm, and in Sect. 6 we present the experimental framework and results. We conclude the paper in Sect. 7.

2 Related work

2.1 Recommender systems for health and wellness

Several studies emphasized the importance of providing users with personalized recommendations, to support them in having a healthy and active lifestyle, and to design effective interventions (Smyth 2019; Kroeze et al. 2006; Yom-Tov et al. 2017).

In this context, some studies focused on recommending physical activities tailored to the user profile. Donciu et al. (2011) bring together the social dimension acquired from a growing community and from expert knowledge defined within an ontology, to provide users with diet and workout recommendations based on their profile information, preferences, and declared purpose. In He et al. (2014), He et al. suggest recommending physical activities to users based on the context (e.g., risk tolerance, budget, location, weather). Ahire and Khanuja (2015) use semantic web technology to analyze users' preferences, build a user profile based on this knowledge, then recommend to users food, and exercise inquiries based on their profile. Khwaja et al. suggest recommending physical activities to users by considering the type of personality (Khwaja et al. 2019). Finally, in Nassabi et al. (2014), Nassabi et al. propose tailoring the recommendations according to the user's health status, goals, and preferences.

Other approaches, instead, have focused on making recommendations to users with specific characteristics. Tseng et al. provide people suffering from chronic diseases (e.g., metabolic syndrome) with diet and exercise guideline recommendations (Tseng et al. 2015). Dobrican and Zampuniéris (2016) focus on cardiac patients with the goal of rehabilitation and, thereby, aims to provide the optimum of both automated and manual interventions (McMurray et al. 1787). Santos-Gago et al. suggest making personalized recommendations to sportswomen by considering their menstrual cycle (Santos-Gago et al. 2019). Berndsen et al. (2019), Smyth and Cunningham (2018), Smyth and Cunningham (2018) propose supporting users in marathon preparation using recommender systems that suggest to runners a challenging, but achievable goal-time in addition to a tailored plan based on a pace.

Even though the use of recommender systems for health and wellbeing is an emerging trend, recent studies put in evidence that having a health care expert-based intervention is necessary when using these kinds of support (Petsani et al. 2018; Martin et al. 2016). However, only a few works on technology-based physical activity promotion have included expert knowledge in their recommendation process.

In our work, we propose recommending athletes to the coaches following them, by considering athletes' performance during their last workout sessions and by assessing the quality of the last workout they have performed. The recommendation process will be described in detail in Sect. 5.

2.2 Fairness in rankings

Across time, there have been many debates on fairness and justice in moral philosophy that led to many and different points of view and thus to different definitions of fairness that are not well-agreed (Binns 2018). Hence, in the Machine Learning field, it is common to evaluate the fairness of an algorithm using measures that assess how much this algorithm is discriminating against a protected group. Fairness in the field of Information Retrieval and, more precisely in ranking problems, has been approached from different perspectives.

Yang and Stoyanovich (2017) suggest assessing fairness in rankings by adopting measures based on statistical parity, that compute the difference in the distribution of different groups for different prefixes of the ranking (top-10, top-20, and so on).

Zehlike et al. (2017) face the challenge of generating a trade-off between fairness and utility in "Top-k ranking" by satisfying two levels of constraints. The first level consists of making sure that the more relevant items are above less relevant ones within the same group, while the second consists of a fairness constraint that ensures that the proportion of protected group items in every prefix of the top-k ranking is above a minimum threshold.

Several other works proposed different fairness constraints that mainly present parity constraints restricting the fraction of items with each attribute in the ranking (Singh and Joachims 2018). However, Biega et al. (2018) go beyond such parity constraints and present a framework that ensures amortized fairness in rankings, based on equity of attention, by focusing on individual fairness while making exposure proportional to relevance for all subjects, using an integer linear program to generate a series of rankings.

In parallel with this work, Singh and Joachims in Singh and Joachims (2018) tackle the challenge of the fairness of exposure in rankings by suggesting a more generic framework for finding rankings that maximize the utility for the user while satisfying a specifiable notion of fairness. The authors propose three fairness constraints:

1. *Demographic Parity* enforces that the average exposure of the documents in the protected and non-protected groups are equal;
2. *Disparate Treatment* enforces that exposure of the protected and non-protected groups to be proportional to their average utility;
3. *Disparate Impact* assures that the click-through rates for the groups as determined by the exposure and relevance are proportional to their average utility.

As mentioned in Singh and Joachims (2018), there is no single definition of a fair ranking, but fairness constraints depend on context and application. Indeed, in some works that presented real-world applications of user recommendation under fairness constraints, the authors have chosen measures that best fit their domain and context.

In Hutson et al. (2018), Hutson et al. highlighted the issue of bias, discrimination, and exclusion w.r.t. race during the matchmaking process, in the study and design of intimate platforms. Also in the people recommendation domain, Geyik et al. (2019) proposed a framework for ensuring fairness in the hiring domain. More precisely, they exploited the concepts of equality of opportunity (Hardt et al. 2016) and fairness through awareness (Dwork et al. 2012) to create fair opportunities for all users seeking a job in the LinkedIn Talent Search platform. In the context of educational recommender systems, Marras et al. introduced a novel fairness metric that monitors the equality of learning opportunity according to a novel set of educational principles and proposed a re-ranking approach to mitigate unfairness in online educational platforms (Marras et al. 2021). Algorithmic fairness has also found attention in other domains, such as speaker verification (Marras et al. 2019; Fenu et al. 2020).

In Sect. 5.4, we will describe in detail the fairness constraints that best fit the context of our work.

3 Preliminaries

Here, we present the preliminaries to provide foundations to our work.

3.1 Recommendation scenario

Let U be a set of athletes, and C be a set of coaches, both belonging to the eCoaching platform. The subscriptions of athletes to the services of the coaches is a binary relation $S \subseteq U \times C$; we denote as S_c the athletes that are followed by a coach $c \in C$. Moreover, we denote as R_w the set of raw features captured by the eCoaching platform during a workout $w \in W$.

Our first goal is to build a model of each workout, denoted as M_w, which captures information about the workout performance of an athlete and contextualizes this performance with the previous behavior of the athlete. More formally, we will build a function $f : R \to M$, which takes the raw features R, to build a new set of features M. Given the set of workout plans prepared by a coach, which is a binary relation $P \subseteq C \times M$, we denote as P_c the plans prepared by a coach $c \in C$. Given a coach $c \in C$ our final goal is to build a function, $g : P_c \times S_c \to S_c$, which considers the set

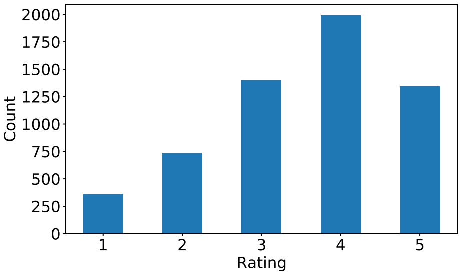


Fig. 3 Ratings distribution in the dataset. The x axis (Rating) reports each rating that could be assigned by a trainer, and y axis (Count) reports the number of workout that received that rating

of workouts of the athletes followed by a coach and ranks those athletes according to their performance. The athletes will be ranked from the best performing to the worst performing one.

4 Dataset and workout modeling

In this section, we provide the details of the dataset employed in this study, and a first characterization of the data. Later, we present the pre-processing steps we performed on the obtained workouts and our approach to model workouts.

Our research is based on a real-world dataset, containing 47,555 activities that compose 8,486 workouts (our set W). This means that each workout is composed of several activities. The workouts were performed by the set U of 412 athletes. Athletes have a different running experience and the coach is aware of the background of the athletes they follow.

The coaches in the platform evaluated these workouts by assigning a rating (denoted as r_w, where w is the workout who received that rating) ranging between 1 and 5. As we are dealing with real-world data, we encountered the problem of class imbalance. Figure 3 represents graphically the distribution of ratings, where "Count" indicates the number of samples having the corresponding rating. We will deal with these phenomena before the classification process, as described in Sect. 6.1.

Table 1 describes the raw features of each user, workout, and activity in the original dataset (our set R). While Fig. 4 presents the distribution of activities and workouts.

4.1 Dataset characterization

In this section, we delve into our data, to understand how it is distributed. This characterization also serves as a motivation to our problem, since we provide insights on data imbalance from multiple perspectives, and conjecture on the possible implications it can have when ranking workout results.

From the left part of Fig. 4, we can see that almost 70% of the activities in the dataset compose only the first 3000 workouts (hence, less than one-third of the workouts comprise 70% of the activities). This means that those workouts are composed of a lot of activities, which makes it very challenging for coaches to analyze and evaluate in a short time. For this reason, it would be helpful to provide coaches with a ranking

Table 1 Description of the raw features

ID	Feature	Type	Description
$u1$	User ID	int	ID of the athlete
$u2$	User Birth Date	Date	Date of birth of the athlete
$u3$	User Gender	string	Gender of the athlete (M for male, and F for female)
$u4$	User Height	int	Height of the athlete (in meters)
$u5$	User Weight	int	Weight of the athlete (in kg)
$w1$	Workout ID	int	ID of the workout
$w2$	Burnt Calories	float	Amount of calories burnt during the workout session
$w3$	Workout Date	date	The date when the workout was performed
$a1$	Activity ID	int	ID of the activity
$a2$	Distance Objective	int	The distance goal given by the coach to the athlete for that activity (in meters)
$a3$	Covered Distance	float	The distance covered by the athlete when performing that activity
$a4$	Speed Objective	int	The speed goal given by the coach to the athlete for that activity (in km/h)
$a5$	Average Speed	float	The average speed performed by the athlete for that activity (in km/h)
$a6$	Time Objective	int	The time goal given by the coach to the athlete for that activity (in seconds)
$a7$	Time Elapsed	float	The time performed by the athlete for that activity (in seconds)
$a8$	Pace Objective	int	The pace goal given by the coach to the athlete for that activity (in min/km)
$a9$	Average Pace	float	The average pace performed by the athlete for that activity (in min/km)
$a10$	Activity Type	string	The type of that activity (either $walking$, $running$, or $resting$)
$a11$	Activity Label	string	The label of that activity (either, $pace$, $distance$, $time$, or $unknown$, indicating the type of objective the activity has; the $unknown$ label is taken by those activities that do not have an objective)

We use four columns to characterize each feature. Concretely, we report the feature's ID (the u prefix denotes a $user$ feature, the w prefix a $workout$ feature, and a an $activity$ feature), the feature's name, the type with which its values can be represented, and a textual description of it

of users in order to spot immediately the athletes that need timely support. Thus, by optimizing the coaches' workload, our system will certainly help increase the efficiency and effectiveness of eCoaching. Observing the right part of the figure, we can remark that almost 70% of the workouts in the dataset were performed by the first 100 users (hence, by around one-fifth of the users). This means that the first 100 users performed a considerable number of workouts, which makes it interesting to contextualize our modeling also with the workout history of users.

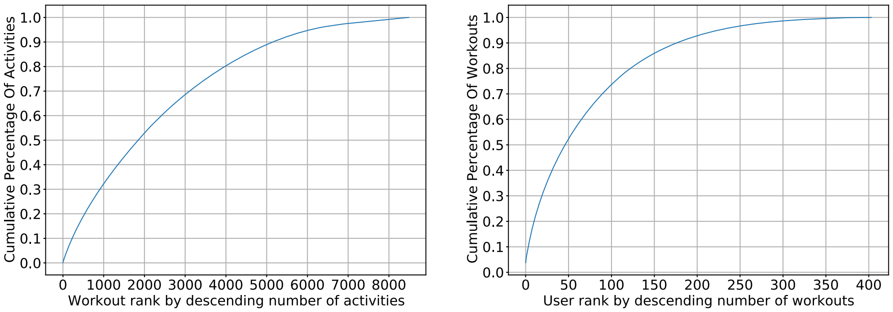
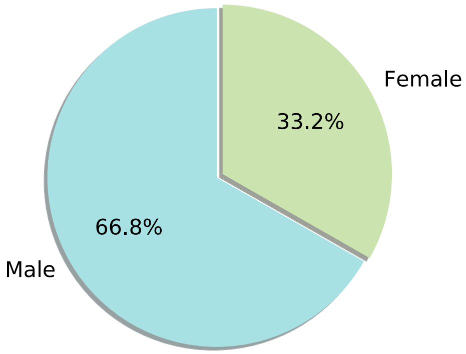


Fig. 4 Distributions of activities and workouts. Cumulative distribution of activities per workouts (left). Cumulative distribution of workouts per users (Right)

Fig. 5 Percentage of workouts performed by each gender. For each gender of the dataset, we report the percentage of workout performed by the users that recognize themselves as belonging to that gender



From Fig. 5, instead, we notice that the percentage of workouts performed by male athletes is mostly twice the percentage of workouts performed by their female counterparts. In line with this, Fig. 6 shows that according to our dataset the male athletes performed on average more workouts per person w.r.t. female athletes (21.72 workouts per person for male athletes vs. 17.29 workouts per person for female athletes) which might have an impact on their support needs. In general, the dataset is composed of 1,936 workouts performed by 112 female athletes and 3,887 workouts performed by 179 male athletes.

Hence, in our dataset, the male users represent the majority group. The difference in the number of workouts performed by different genders in our dataset may lead our system to be biased w.r.t. the workouts performed by the users of the gender that performed more workouts (i.e., males). We would like to remark once again that gender should not impact the ratings, since the coaches that created the workout plans and rated the performance of users take into consideration the gender of users, their experience, and their health conditions.

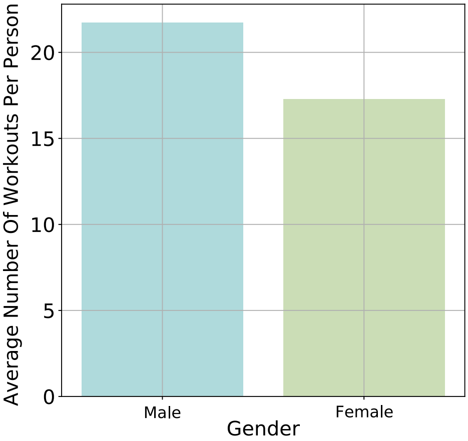


Fig. 6 Average number of workouts per person performed by each gender. For each gender of the dataset, we report the average number of workouts performed by the users that recognize themselves as belonging to that gender

4.2 Data pre-processing and feature extraction

Data Pre-processing. From all the workouts in the dataset, we removed all those that are not reliable. According to the *u4fit* coaches, a workout is not reliable when at least one of the following conditions is met: (i) $covered\ distance > 43,000\ meters$, (ii) $workout\ duration > 5\ hours$, (iii) $rest\ time > 1\ hour$, (iv) $average\ speed > 16\,km/h$. We also removed the workouts that were not performed under the supervision of a coach. After removing the irrelevant workouts, the final dataset consists of 5,823 workouts performed by 291 users.

Feature Extraction. Given the raw features available in our dataset and presented in Table 1, the next goal is to model each workout, by doing some feature engineering. We regrouped all the activities that belong to each workout and excluded the activities that have $resting$ as $activity\ type$ (feature $a10$) since, according to coaches, they are not considered when evaluating workout quality; for this reason, they should not be part of our user modeling and recommendation algorithm.

In Table 2, we describe the features we created, and how they are derived from the original ones. We can observe that some features reflect the phenomena we have observed in Fig. 4. Specifically, feature $f24$ (Days From Previous Workout) measures the number of days from the previous workouts session and feature $f30$ (User Fidelity) measures the number of workout sessions the user has performed from the first time they used the platform, thus giving an idea about the workout intensity and frequency of a certain athlete. According to Fig. 4, a small percentage of users in the dataset performed more than 50% of the workouts, which means that it is crucial to capture information about the workout intensity and the workout frequency of athletes. The feature $f31$ (Rating Weighted Sliding Average) captures the temporal evolution of athletes' performance by giving more importance to the last workout session without neglecting the past ones. The choice of giving a 75% weight to the last workout was made after collecting insights from the $u4fit$ coaches on how, in their evaluation process, the last workout contextualizes with the previous history of the users. Hence, this engineering of the feature became our way to operationalize these insights.

Table 2 Workout modeling features

Category	ID	Feature	Type	Description
	$f1$	Workout ID	Int	ID of the given workout, directly derived from $w1$
Distance-based features	$f2$	Distance Objective	Float	Sum of the distance objectives of the activities of the considered workout (feature $a2$ in Table 1)
	$f3$	Covered Distance	Float	Sum of all the covered distances of the activities of the considered workout (feature $a3$ in Table 1)
	$f4$	Distance Gap	Float	Obtained by first calculating the difference between the distance objective (feature $a2$ in Table 1) and covered distance (feature $a3$ in Table 1) for each activity in the workout, and then averaging the obtained values (this feature indicates how well the users respected their distance objective)
	$f5$	Distance Gap Variance	Float	Variance of the distance gaps in each activity considered to compute feature $f4$ (this feature indicates how far are the individual values from the average)
	$f6$	Distance Gap Standard Deviation	Float	Standard deviation of the distance gaps in each activity considered to compute feature $f4$ (this feature also indicates how far are the individual values from the average, but it is expressed in the same units as the data)

Table 2 continued

Category	ID	Feature	Type	Description
Temporal features	$f7$	Time Objective	Int	Sum of the time objectives of the activities of the considered workout (feature $a6$ in Table 1)
	$f8$	Workout Duration	Float	Sum of all the time the user has taken to complete the activities of the considered workout (feature $a7$ in Table 1)
	$f9$	Temporal Gap	Float	First create the difference between the time objective (feature $a6$ in Table 1) and elapsed time (feature $a7$ in Table 1) for each activity in the workout, and then average the obtained values (this feature indicates how well the user respected their time objective)
	$f10$	Temporal Gap Variance	Float	Variance of the temporal gaps in each activity considered to compute feature $f9$
	$f11$	Temporal Gap Standard Deviation	Float	Standard deviation of the temporal gaps in each activity considered to compute feature $f9$
Pace-based features	$f12$	Pace Objective	Int	Average of the pace objectives of the activities of the considered workout (feature $a8$ in Table 1)
	$f13$	Average Pace	Float	Average of the paces of the activities of the considered workout (feature $a9$ in Table 1)

Table 2 continued

Category	ID	Feature	Type	Description
	$f14$	Pace Gap	Float	Obtained by first calculating the difference between the pace objective (feature $a8$ in Table 1) and average pace (feature $a9$ in Table 1) for each activity in the workout, and then averaging the obtained values (this feature indicates how well the user respected their pace objective)
	$f15$	Pace Gap Variance	Float	Variance of the pace gaps in each activity considered to compute feature $f14$
	$f16$	Pace Gap Standard Deviation	Float	Standard deviation of the pace gaps in each activity considered to compute feature $f14$
Workout characteristics	$f17$	Walking Activities' Percentage	Float	Percentage of activities in a workout where feature $a10$ is equal to $walking$
	$f18$	Running Activities' Percentage	Float	Percentage of activities in a workout where feature $a10$ is equal to $running$
	$f19$	Percentage of Activities with an Objective	Float	Percentage of activities in a workout where feature $a11$ is not equal to $unknown$
	$f20$	Percentage of Well-performed Activities	Float	Percentage of activities in a workout that have any gap equal to 0
	$f21$	Week Day	Int	The day of week when the workout was performed; this feature takes values from 1 to 7 and is obtained from the feature $w3$ in Table 1

Table 2 continued

Category	ID	Feature	Type	Description
	$f22$	Week Number	Int	The week of year when the workout was performed; this feature takes values from 1 to 53, to account for years who have 53 weeks, and is obtained from the feature $w3$ in Table 1
	$f23$	Month	Int	The month when the workout was performed; this feature takes values from 1 to 12, and is obtained from the feature $w3$ in Table 1
	$f24$	Days From Previous Workout	Int	The number of days from the previous workouts session.
User characteristics and behavior	$f25$	User Age	Int	Created using feature $u2$ described in Table 1, in order to contextualize the workout performance with the age of the user
	$f26$	User Gender	Categorical	Directly computed from feature $u3$ described in Table 1 (0 for female, 1 for male)

Table 2 continued

Category	ID	Feature	Type	Description
	$f27$	User Height	Int	Directly computed from feature $u4$
	$f28$	User Weight	Int	Directly computed from feature $u5$
	$f29$	User BMI	Float	Computed using features $f27$ and $f28$
	$f30$	User Fidelity	Int	Number of workout sessions the user has performed from the first time they used the platform.
	$f31$	Rating Weighted Sliding Average	Float	Decaying average of the ratings r_w obtained by the user in previous workouts. This feature allows us to monitor the evolution in the performance of a user, by giving more importance to the last sessions without neglecting the past ones. The decaying average of the element X at the position N is: $\overline{X}_N = \frac{3}{4} \cdot X_{N-1} + \frac{1}{4} \cdot \overline{X}_{N-2}$.

We use five columns to model each workout. Concretely, we report the feature's category, its ID, the feature's name, the type with which its values can be represented, and a textual description of it

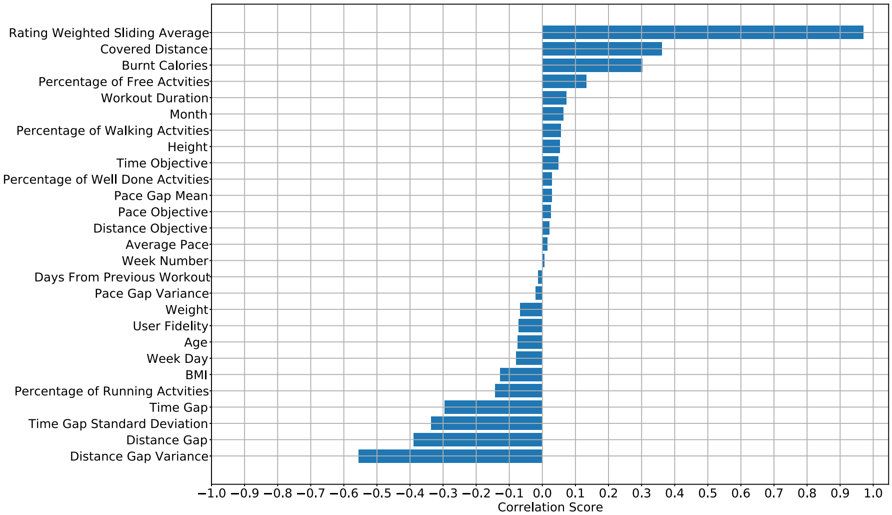


Fig. 7 Features' correlation with the target classes. Each line reports the relative correlation of a feature with the target classes, in a score between −1 and 1. A positive correlation score implies that the target class gets higher as the feature grows, while a negative score implies that the target class gets lower when the feature's value grows

Figure 7 shows the correlations between the target classes and the features in Table 2. The features that are highly correlated with the target classes and have a positive correlation score are the Weighted Sliding Average of the ratings obtained by the athletes during their previous workout sessions, Covered Distance, and Burnt Calories. The features that are highly correlated with the target classes and have a negative correlation score are the Distance Gap Variance, Distance Gap and Time Gap. Namely, the highly correlated features with the target classes are the ones that model the previous athletes' performance during the previous workout session, then the ones that model the effort and adherence to the objectives set by the trainer during the current workout session.

Figure 8 shows how the features with a positive high correlation score with the target classes relate to each other and to the target classes. We can observe from this figure that the athletes that usually achieve a good performance are more likely to get a high score and vice versa.

5 Fair user recommendation

In this section, we describe the algorithm we implemented to recommend users who need the support of the coach.

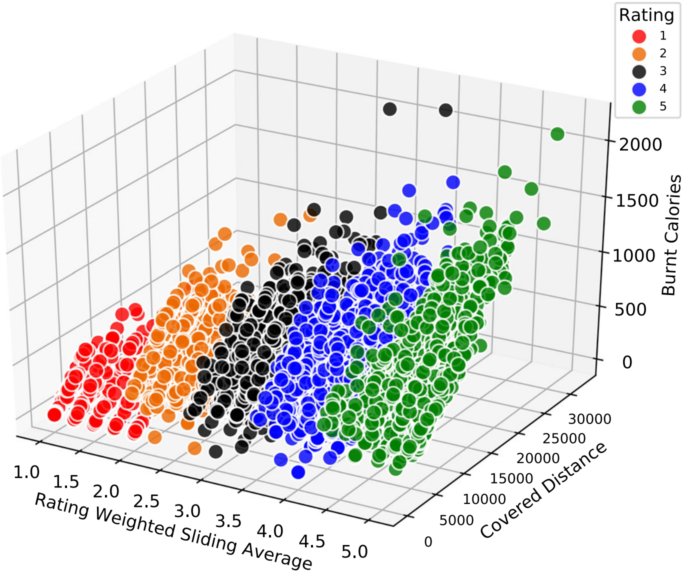


Fig. 8 The impact of the most correlated features on the target classes and how they relate with each other. Each point represents a workout in terms of the Rating Weighted Sliding Average (x axis), Covered Distance (y axis), and Burnt Calories (z axis)

5.1 Motivation

Before we go into the detailed steps of our approach, it is important to highlight why our approach departs from the main classes of recommender systems (collaborative filtering and content-based approaches) and from classic people recommender systems:

- *Classic people recommenders* exploit the topology of the social network ("since you are connected to these users, you might connect to these"); this would not fit our work, since in this work we are not recommending athletes to coaches that might suit them, but we recommend to coaches those who need support after a workout;
- *Collaborative-filtering approaches* do not consider item features, which are essential to predict if a user needs support or not (we are basing support on a prediction of workout quality). Moreover, collaborative filtering approaches consider *static items* (e.g., a movie does not change over time), while in our domain there is no such thing as two identical workouts. Hence, collaborative algorithms would not fit our approach either;
- *Content-based approaches* match two users based on the content they post. While training results are a form of content exploited by our algorithm, the matching between the coach and the athlete is not what triggers our recommendations.

Workout quality and its relation to previous users' behavior and their objectives are what drive the recommendation of a user to a coach, thus making our problem new from a recommendation point of view. Hence, no direct comparison of our work to existing people recommenders is possible.

Continuing, we motivate our choice to provide fairness via a re-ranking approach and how our method departs from the existing ones. Mitigation methods for unfairness in rankings can be categorized into pre-processing, in-processing, and post-processing methods:

- *Pre-processing methods* aim to mitigate disparities in user ranks by intervening at the level of training data, either before these candidates are processed by a ranking algorithm or during the ranking process;
- *In-processing methods* intervene on the ranking algorithm such that it produces a ranked outcome that meets the specified fairness criteria;
- *Post-processing methods* intervene on the output ranking in such a way that it meets the specified fairness criteria.

From the perspective of fairness, we opted for a post-processing method by making a classic assessment of *the exposure given to the different genders in the ranking*. Here, the application domain is new, by providing fairness to users in need of support in eCoaching platforms.

A re-ranking algorithm is the best option when optimizing ranking-based metrics, such as visibility and exposure. An in-processing regularization, such as those that have been presented in Kamishima et al. (2018), Beutel et al. (2019), would not be possible, since at the prediction stage the algorithm does not predict *if and where* an item will be ranked in a recommendation list; hence, no direct comparison with these approaches is possible. While list-wise approaches might support an in-processing approach, the applicability of a solution would be limited only to this class of algorithms, thus limiting the transferability of our work. Instead, a re-ranking approach, like the one we propose in this study, can be applied to any class of ranking algorithms. Re-rankings have been introduced to reduce disparities, both in the context of non-personalized rankings (Zehlike et al. 2017; Singh and Joachims 2018; Biega et al. 2018; Celis et al. 2018; Zehlike and Castillo 2020; Patro et al. 2020) and of recommender systems (Mehrotra et al. 2018; Burke et al. 2018), with approaches such as Maximal Marginal Relevance (Carbonell and Goldstein 1998).

However, all these algorithms optimize only one property (either utility or exposure). As we will show later in our ablation study, optimizing for one metric is not enough, so no direct comparison with these approaches is possible.

5.2 Our approach

The user recommendation process is divided into two main steps:

1. *Performance-based ranking*: we rank the users based on the performance in the last workout, contextualized to their recent behavior.
2. *Fair re-ranking*: we assess how fair is the ranking algorithm in terms of exposure of the users and provide a re-ranking algorithm for the cases in which users of a given gender are affected by disparate exposure.

The steps are now described in detail.

5.3 Performance-based ranking

The intuition behind this algorithm is that predicting the quality of a workout is a central element in order to provide a recommendation to a coach. For this reason, we initially predict the rating that the coach would assign to a given workout. The input received by the classifier is the workout model composed of the 30 features we engineered in Sect. 4.2. Different classes of classification algorithms can be employed for the purpose of predicting workout quality, from ordinal to multi-class approaches. As we will show in Sect. 6.2, the chosen class of algorithms implies treating the ground truth as a continuous or disjoint set of classes (ordinal and multi-class classification, respectively); in our evaluation, we explore the effectiveness of the two classification strategies in our context. The output of a classifier is a predicted rating, denoted as $\hat{r}_w$.

Finally, we rank the users based on the predicted rating $\hat{r}_w$. The "urgency" with which they will get support mostly depends on their performance during their previous workout sessions. In general, a high $\hat{r}_w$ leads to a high rank. Instead, if $\hat{r}_w$ is low, the user will get more timely support.

A recommendation list **R** for a coach is represented by the list of users followed by them, ranked by ascending $\hat{r}_w$.

Since coaches and athletes have a continuous relationship, we simulate the recommendation scenario of the real-world application. Under this scenario, we assume that the coach will check who might need support by checking the *u4fit* application at regular intervals. To simulate these intervals, we start by ranking the users that performed the first 5 workouts for each coach, then we update the ranking for each coach whenever the athletes followed by this coach perform 5 new workouts.

5.4 Fair re-ranking

Every output generated by the previous step is a list of users to be recommended to a coach, based on their likelihood of needing support, according to their performance.

The classification algorithm uses the gender of the users as a feature used in the classification process (feature $f24$). For this reason, systematically under-exposing the users of a given gender would mean that the ranking is affected by the so-called *disparate treatment*. Disparate treatment means that users belonging to a given gender might be ranked lower w.r.t. to their counterpart, even though they might need the same (or more) support.

Hence, the first step is to assess *how fair* is the ranking, in terms of the exposure given to the users (Singh and Joachims 2018). The exposure that a user gets in a ranking is given by:

$$\text{Exposure}\ (u|\mathbf{R}) = \frac{1}{\log(1+j)} \tag{1}$$

where j is the position the user covers in **R**.

In order to measure how "deserving" is that user to cover position j in the ranking, we measure their utility according to:

$$\text{Utility } (u|\mathbf{R}) = \frac{2^{\text{rel(u)}} - 1}{\log(1+j)} \tag{2}$$

where $rel(u) = max(r_w) - \hat{r}_w$.

It should be trivial to note that the utility of a user corresponds to their DCG, which is a common practice in the literature (Singh and Joachims 2018).

Let G_i denote the subgroup of users having the same gender. The Exposure and Utility for that group are calculated as follows:

$$\text{Exposure } (G_i|\mathbf{R}) = \frac{1}{|G_i|} \sum_{u \in G_i} \text{Exposure(u)} \tag{3}$$

and

$$\text{Utility } (G_i|\mathbf{R}) = \frac{1}{|G_i|} \sum_{u \in G_i} \text{Utility(u)}\,. \tag{4}$$

We first assume a recommendation list (ranking) to be fair if the two groups get the same $Exposure$, defined as follows:

$$\text{Exposure } (G_0|\mathbf{R}) = \text{Exposure } (G_1|\mathbf{R})\,. \tag{5}$$

In order to assess if a recommendation list is fair, we measure Demographic Parity Ratio (DPR) as follows:

$$\text{DPR } (G_0, G_1|\mathbf{R}) = \frac{\text{Exposure } (G_0|\mathbf{R})}{\text{Exposure } (G_1|\mathbf{R})} \tag{6}$$

A DPR equal to 1 indicates the users of a given gender get a fair exposure, while a value lower or greater than 1 tells us which group is disadvantaged in terms of disparate exposure.

The DPR metric only accounts for the position in which users are ranked, without accounting for their utility, in demographic parity fashion. To account also for the $Utility$ of the users of a given group, we introduce another constraint that considers it, to balance $Exposure$ of the two groups while preserving ranking quality:

$$\frac{\text{Exposure } (G_0|\mathbf{R})}{\text{Utility } (G_0|\mathbf{R})} = \frac{\text{Exposure } (G_1|\mathbf{R})}{\text{Utility } (G_1|\mathbf{R})}. \tag{7}$$

In order to assess if a recommendation list is fair, we measure Disparate Treatment Ratio (DTR) as follows:

$$\text{DTR } (G_0, G_1|\mathbf{R}) = \frac{\text{Exposure } (G_0|\mathbf{R}) \text{ / Utility } (G_0|\mathbf{R})}{\text{Exposure } (G_1|\mathbf{R}) \text{ / Utility } (G_1|\mathbf{R})} \tag{8}$$

A DTR equal to 1 indicates fair exposure for the users, while a value lower or greater than 1 tells us which group is disadvantaged in terms of disparate exposure.

In case our two metrics, DPR and DTR, report scores different from 1, we developed a re-ranking approach to generate a fair exposure. The intuition behind our algorithm is that each pair of users that have different gender and appear consequently in a ranking is a candidate for a swap, so that the disadvantaged gender can be given more exposure. Our approach is summarized in Algorithm 1.

Algorithm 1: Order-Based Re-ranking

```
   input  : $X$: users sorted by rank, $D$: fairness metric (either $DTR$ or $DPR$)
   output: $R$: ranked list of users that respects group fairness constraints
 1 $d \leftarrow empty\,dictionary$;
 2 $s \leftarrow empty\,dictionary$;
 3 $d[X] \leftarrow D$;
 4 $s[X] \leftarrow$ getAllSwappableRows($X$);
 5 while $s[X]$ is not empty do
 6     $p \leftarrow getNextPair(s[X])$;
 7     remove $p$ from $s[X]$;
 8     $X\_temp \leftarrow$ swapPair($X$, $p$);
 9     $D\_temp \leftarrow$ calculateD($X\_temp$);
10     if $D\_HasImproved(D, D\_temp)$ then
11         $X \leftarrow X\_temp$;
12         $D \leftarrow D\_temp$;
13         $d[X] \leftarrow D$;
14         $s[X] \leftarrow$ getAllSwappableRows($X$);
15     end
16 end
17 $R \leftarrow$ the ranking in $d$ that have the best $D$ value;
18 return $R$;
```

The algorithm takes as input the list of users in a ranking update and a metric D that measures either Disparate Treatment Ratio (DTR) or Demographic Parity Ratio (DPR). First, the algorithm creates two empty dictionaries; in the first, we save the rankings as keys, with the metric D associated with that ranking stored as value and, in the second, we save the ranking as key and the list of possible pairs of users to swap as value. In line 3, we save in the first dictionary (d) the original ranking as key and the respective D as value. Then, in line 4, the function *getAllSwappableUsers* looks for the disadvantaged gender (if $D < 1$, then G_0 is disadvantaged, while if $D > 1$, then G_1 is disadvantaged) and returns a list containing the pairs of indexes of the users that could be swapped, ordered by their occurrence, such that the disadvantaged gender may get more attention. Then, we save the ranking and the users to swap, respectively, as key, value in the second dictionary (s).

In line 6, we take the first pair of users to swap and check if D has improved (i.e., $abs(1 - D) > abs(1 - D\_temp)$). If this is the case, we save the new ranking and the respective D to d, update the users to swap given this new ranking, and repeat this process until we make sure there are now users that we can swap and that can improve the value of D for the ranking (lines 5-16). Finally, from d we take the ranking that has the best D value.

6 Experimental framework

This section describes the experiments performed to validate our proposal.

6.1 Experimental setup

The experimental framework exploits the Python scikit-learn 0.19.1 library. The experiments were executed on a computer equipped with a 3.1 GHz Intel Core i7 processor and 16 GB of RAM.

The learning phase and consequently the prediction of most Machine Learning classifiers may be biased toward the occurrences that are frequently present in the dataset (Rathore and Kumar 2017; Klement et al. 2009).

Researchers have suggested two main approaches to deal with data imbalance: the first approach consists of tuning the data by performing a sampling, and the other is to tweak the learning algorithm (Klement et al. 2009). Due to its effectiveness in our data, we employed the first approach. More specifically, we have considered the oversampling approach, since it is more effective for small dimension datasets (Sáez et al. 2016). We opted for *Synthetic Minority Over-sampling Technique Tomek* (SMOTETomek), since it creates completely new samples and eliminates only examples belonging to the majority class instead of replicating the existing ones, which offers more examples to the classifier to learn from. This means that the minority class examples are over-sampled, whereas the majority class examples are under-sampled (Chawla et al. 2002; Batista et al. 2004).

In our framework, we applied SMOTETomek using *imbalanced-learn*, which is a package that provides a set of sampling approaches used in datasets showing high class imbalance (Lemaître et al. 2017).

6.2 Evaluation strategy

In this section, we present our strategy to evaluate our proposal.

6.2.1 Workout quality prediction

In order to rank the users that need timely support, we first predict the quality of their performance during the last workout that the coach assigned to them. To this end, we compare two kinds of classification, the first is Ordinal classification (which takes into account the order of ratings) and the second is multi-class classification (which does not take into account the order of ratings). The comparison of these two classification strategies will allow us to assess if the classification process should follow the same process a coach uses to rate a workout (i.e., by considering that additional points in the rating mean a better workout, in ordinal fashion) or if workouts follow given patterns and a rating should be treated as a class, in multi-class fashion.

Ordinal Classification.

In this study, we compared four ordinal classifiers, which consider as classes the ordered set of ratings.

1. *Ordinal Ridge (OR).* This classifier overwrites the Ridge classifier in scikit-learn, so that it uses the (minus) absolute error as a score function. Ridge regression provides biased estimates and is the best-known penalization approach (Brooks and Dulá 2013). Ridge regression approximates parameter estimates to zero value without making them completely zero (Landschoot et al. 2013);
2. *Least Absolute Deviation (LAD)* is a statistical optimization technique that minimizes the sum of the absolute values of the residuals. LAD can be classified as a nonlinear optimization problem. This provides a robust estimator. However, LAD regression is not robust when the data has outliers in the illustrative variables (Gao and Feng 2018);
3. *Logistic Immediate-Threshold (LIT).* This classifier implements the ordinal logistic model, considering the Immediate-Threshold variant. If the threshold values defined for each class are violated, a penalty is imposed. However, the immediate threshold method does not guarantee that the threshold values will be consecutive (Rennie 2005);
4. *Logistic All-Threshold (LAT).* This classifier implements the ordinal logistic model, considering the All-Threshold variant. The All-Threshold-based method was introduced to guarantee that the thresholds will be ordered by imposing more penalties (Rennie and Srebro 2005). The all-threshold loss corresponds to a total value of all threshold violation penalties. Therefore, solutions in the All-Threshold method are desired to have the minimum number of crossed thresholds (Topal et al. 2010).

Multi-class Classification. To treat the workout-quality prediction problem as multi-class classification, we compared four tree-based classifiers, as these perform better compared to those that are not tree-based when it comes to low-dimensional data (Rathore and Kumar 2017).

1. *Gradient Boosting (GB)* is an ensemble algorithm that improves the accuracy of a predictive function through incremental minimization of the error term. After the initial base learner (almost always a tree) is grown, each tree in the series is fit to the so-called “pseudo residuals” of the prediction from the earlier trees with the purpose of reducing the error (Brown and Mues 2012);
2. *Random Forest (RF)* is a meta-estimator of the family of the ensemble methods. It fits a number of decision tree classifiers, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all the trees in the forest (Breiman 2001);
3. *Extra Trees (ET)* is another ensemble method. Similarly to Random Forest, it uses a random subset of candidate features while splitting a tree node; however, instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule (Geurts et al. 2006);
4. *Decision Tree (DT)* is a non-parametric supervised learning method used for classification and regression. One of the main advantages of decision trees with respect to other classifiers is that they are easy to inspect, interpret, and visualize, given they are less complex than the trees generated by other algorithms addressing nonlinear needs (Boratto et al. 2018).

6.2.2 Strategy

To validate our proposal, we performed five sets of experiments:

1. *Classifiers comparison*. We evaluated the ordinal and multi-class classifiers, by running them on all the features. We compared the accuracy metrics they obtained, in order to determine the most effective one;
2. *Feature sets importance evaluation*. After choosing the most effective ordinal and multi-class classifiers, we evaluated the importance of the used features by measuring the correlation between the value of each feature and the values predicted using the best performing classifier, to understand how each feature impacts the quality of workouts;
3. *Ablation study*. We took away the least important feature one by one, and evaluated the classification accuracy, to check how the less relevant features affected the effectiveness of the classifiers;
4. *Re-training simulation*. To simulate the real-world scenario, we re-train and monitor the performance of the best ordinal and multi-class classifiers every 100 workouts (i.e., we first train the classifier on the first 100 workouts and evaluate its performance on the following 100 on then we train on the first 200 and evaluate its performance on the following 100).
5. *Ranking Under Fairness Constraints* To rank the users, we sort them according to the rating predicted by the best classifier. Then, we compare the effectiveness and the fairness of the resulting ranking, before and after applying Algorithm 1 described in Sect. 5.4. To simulate the real-world scenario, we re-rank the users for each coach every n new workouts.

6.3 Metrics

Workout Quality Prediction. Our approach ranks users based on their workout performance. For this reason, the first set of evaluation metrics should be capable of capturing how effective is a classification approach at predicting workout quality. Given that the ground truth is represented by 5-star ratings, we had to choose metrics that are most suitable for multi-class datasets. Nevertheless, the majority of the performance measures present in the literature are designed only for two-class problems (Galar et al. 2011).

However, several performance metrics for two-class problems have been adapted to multi-class ones. Some measures that fit well our needs, give us relevant information about the performance of our classifier, and are successfully applied for multi-class problems are Accuracy, Recall, Precision, F2-score, and Informedness (Galar et al. 2011). In what follows, we present these metrics in detail.

Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{P + N} \tag{9}$$

where P represents positively labeled instances, and N represents negatively labeled ones. TP represents the true positives (i.e., instances of the positive class that are correctly labeled as positive by a classifier), TN represents the true negatives (i.e., instances of the negative class that are correctly labeled as negative by a classifier). It represents the fraction of all instances that are correctly classified.

Recall is defined as:

$$Recall = \frac{TP}{P} \tag{10}$$

and it measures the completeness of a classifier.

Precision is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

and it measures the exactness of a classifier.

F2-score is defined as:

$$F2 = 5 \cdot \frac{Precision \cdot Recall}{4 \cdot Precision + Recall} \tag{12}$$

and it is a metric that considers both recall and precision.

None of the metrics presented so far takes into account the true negative rate (defined as TN/N) and this is an issue when dealing with imbalanced datasets (Powers 2011). Considering this, we decided to measure *Informedness*, which is the clearest measure of the predictive value of a system (Powers 2012). *Informedness* is defined as:

$$Informedness = Recall + true\_negative\_rate - 1 \tag{13}$$

where $true\_negative\_rate$ is TN/N. It ranges between -1 and 1, where 1 represents a perfect prediction, 0 no better than random prediction, and -1 indicates total disagreement between prediction and observation. This metric is particularly effective for multi-class problems as opposed to the accuracy (Galar et al. 2011).

Ranking Under Fairness Constraints. To evaluate the ranking quality, we compare the ranking lists generated as output by the model and those given as the ground truth (i.e., the user rankings shaped based on the ratings assigned to each coach for the workouts in the test set). The most suitable metric for this purpose is the Normalized Discounted Cumulative Gain (NDCG).

We compared our rankings effectiveness using an exponential gain and logarithmic decay based on the graded relevance judgments. In our case, NDCG at position k is defined as:

$$NDCG@k(R) = \frac{1}{N} \sum_{j=1}^{k} \frac{2^{rel(u_j)} - 1}{\log(j+1)} \tag{14}$$

where N is the maximum possible DCG given the known relevant users, u_j is the u^{th}-ranked user returned by R, and $rel\left(u_j\right)$ is the binarized relevance assessment of this user (Radlinski and Craswell 2010). NDCG values range between 0 and 1; the higher the value, the better.

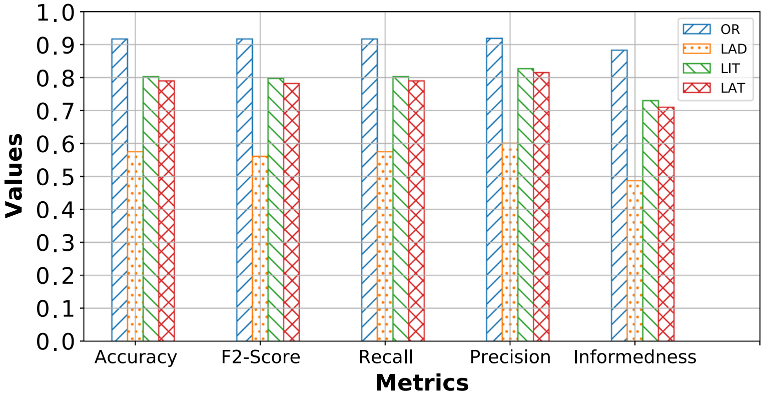


Fig. 9 Ordinal classifiers comparison. Each block of columns reports the results obtained for each metric. Each column denotes an ordinal classifier. The higher the value, the better is the classifier

6.4 Experimental results

In this section, we present our results.

6.4.1 Classifiers comparison

- *Ordinal classification*. Figure 9 shows that LIT, LAT, and OR achieved the best performance, where LAD achieved the worst results. The ordinal classifier that gets the best scores for all the metrics is OR. It achieves an F2-Score of almost 92% and an Informedness of 0.88, which means that we are correctly predicting the rating of a workout in 92% or more of the cases. Based on these results, OR is the ordinal classifier chosen for the subsequent analyses.
- *Multi-class classification*. Figure 10 shows that almost all the classifiers have a good performance, but RF is the one that gets the best scores for all the metrics. Concretely, RF achieves an F2-Score of almost 93% and an Informedness of 0.91, which means that we are correctly predicting the rating of a workout in 93% or more of the cases. Based on these results, RF is the multi-class classifier chosen for the subsequent analyses.
- *Ordinal vs. Multi-class classification*. The best ordinal classifier and the best multi-class classifier achieve similar performance for the rating prediction task, nevertheless, RF performs slightly better than OR for all the metrics. This is true for all the metrics we consider to evaluate classification quality. This leads us to our first observation.

Observation 1. *The ratings that the coaches use to assess workout quality are in an ordinal*

scale and, conceptually, an ordinal classifier would better suit this task. However, the multi-class classifiers outperform the ordinal ones. Hence, we conjecture that coaches might have a more schematic way of evaluating workouts, better captured by multi-class approaches.

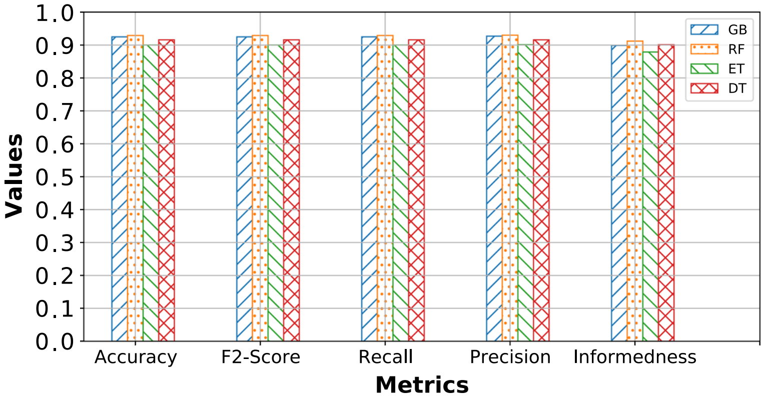


Fig. 10 Multi-class classifiers comparison. Each block of columns reports the results obtained for each metric. Each column denotes a multi-class classifier. The higher the value, the better is the classifier

6.4.2 Feature sets importance evaluation

Figure 11 illustrates the impact of each feature on the performance of OR, using a scale ranging from 0 (no importance) to 100 (very important). We can see that the features that have more impact on the classification process are mainly those that model the recent behavior of the users and their adherence to their workout objectives. The "Rating Weighted Sliding Average" average is the most important feature, and we assume that this is due to the fact that it represents the decaying average of the recent ratings achieved by the users, and since users usually tend to change their behavior gradually their performance is very correlated with their recent ratings. We can see also that the effort (Burnt Calories), the month when the workout sessions were planned, and the percentage of well-performed activities have a significant impact on the workout quality prediction. However, user characteristics were not very relevant to the classifier. The "Rating Weighted Sliding Average" is also the feature with the highest correlation score with the target classes, when the importance of the other features is captured in a different way. Nevertheless, the features judged to have a high impact on both classifiers' prediction reflect the correlations of these features with the target classes presented at the end of Sect. 4.

Figure 12 illustrates the impact of each feature on the performance of RF, using a scale ranging from 0 (no importance) to 100 (very important). We can see that the features that have more impact on the classification process are mainly those that model the recent behavior of the users and their adherence to their workout objectives. The rating weighted sliding average is the most important feature also according to RF. We can see also that the covered distance, average pace, and the week number have a significant impact on the workout quality prediction. However, user characteristics and workout characteristics were not very relevant to the classifier.

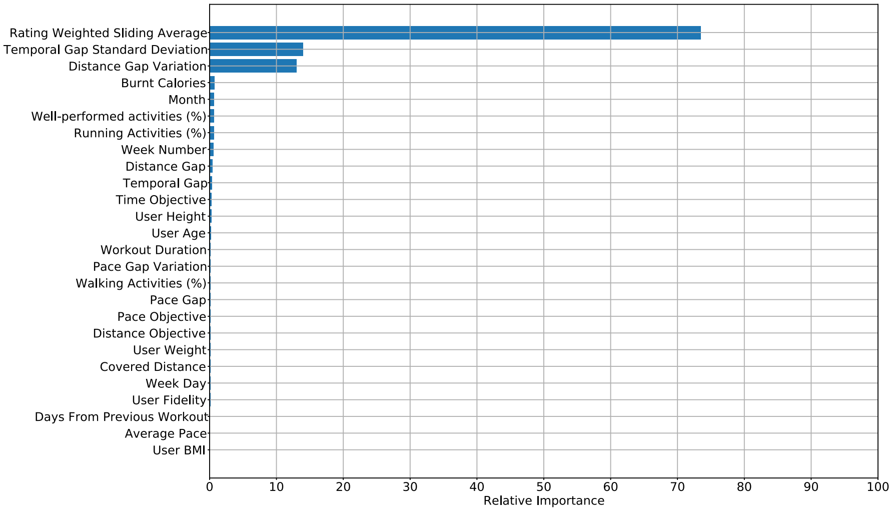


Fig. 11 Features' importance for the OR classifier. Each line reports the relative importance of a feature, in a score between 0 and 100. The higher is the score, the more important is the feature

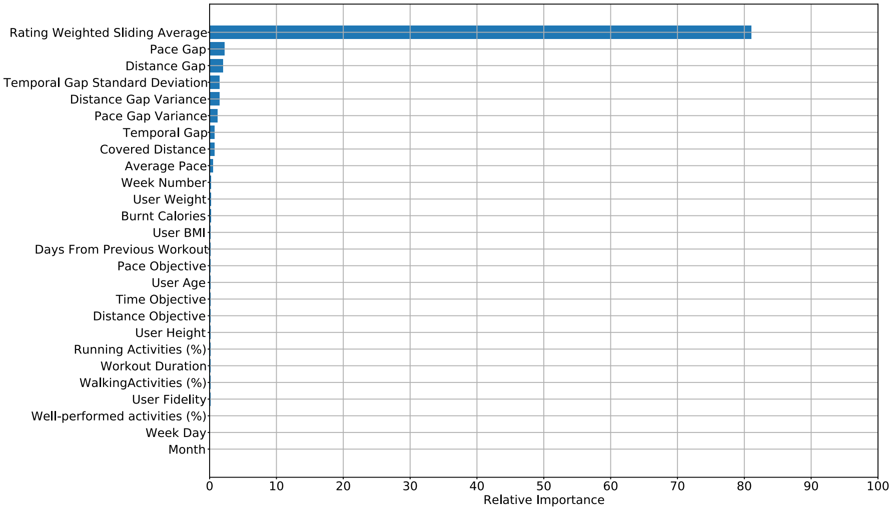


Fig. 12 Features' importance for the RF classifier. Each line reports the relative importance of a feature, in a score between 0 and 100. The higher is the score, the more important is the feature

6.4.3 Ablation study

During the ablation study, we train the classifier on different feature settings by removing features one by one, starting from the least important (i.e., for OR, the first setting runs the classifier without the User BMI, while in the second setting we removed User BMI and Average Pace, and so on).

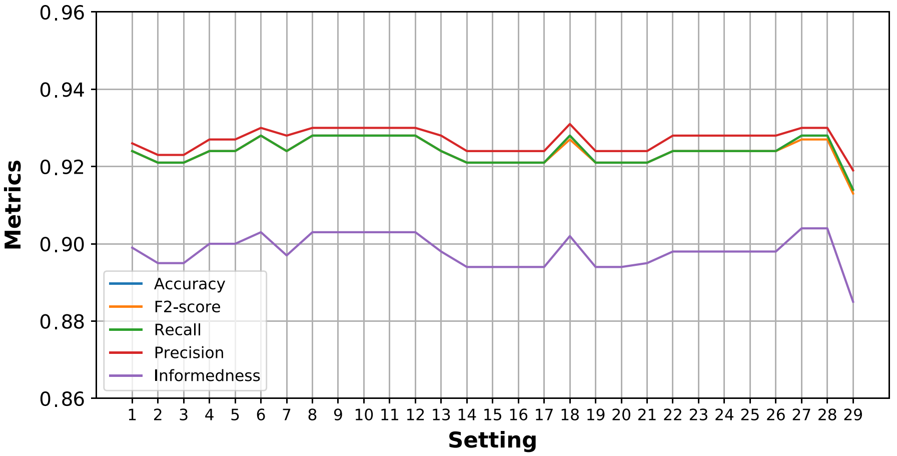


Fig. 13 Results returned by training OR with different sets of features. For each set of features, denoted in the *x* axis (Setting), we report the value obtained by each metric

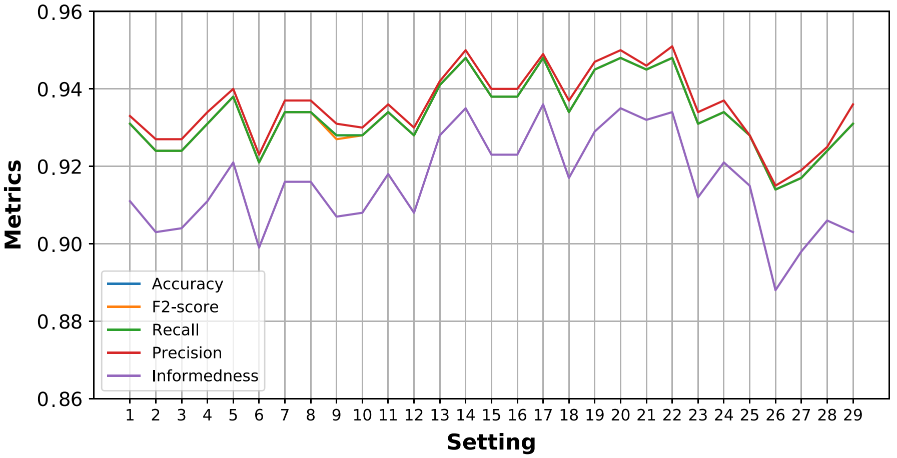


Fig. 14 Results returned by training RF with different sets of features. For each set of features, denoted in the *x* axis (Setting), we report the value obtained by each metric

Training OR on fewer features showed that it achieves a better performance using the feature set 18 (i.e., when we do not consider the first 18 less important features while training the classifier), as reported in Fig. 13.

Training RF on fewer features showed that it achieves a better performance using the feature set 14 (i.e., when we do not consider the first 18 less important features while training the classifier), as reported in Fig. 14.

Table 3 shows the best performance of OR and RF after training them on fewer features. Both classifiers achieve a very good performance, though RF outperforms OR for all the metrics we considered.

Table 3 Performance of OR and RF when trained on the best feature sets

Classifier	OR	RF
Accuracy	0.928	**0.945**
F2	0.927	**0.945**
Recall	0.928	**0.945**
Precision	0.931	**0.948**
Informedness	0.902	**0.930**

Each line reports the results of a metric and each column the classifier associated with the reported results (The values that are in bold represent the best results for each metric)

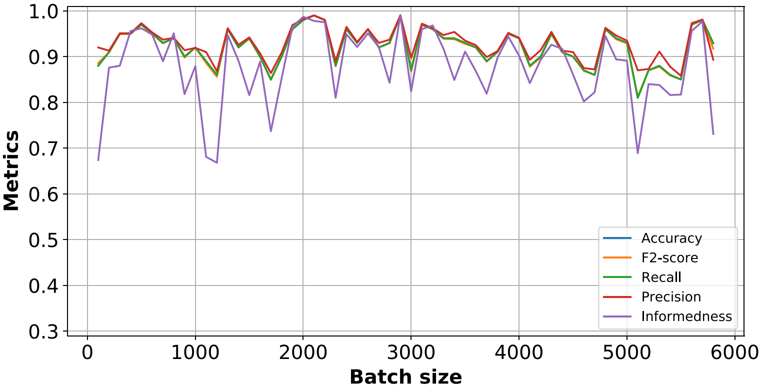


Fig. 15 Evolution of the performance of OR. The x axis (Batch size) contains a point every 100 workouts, that is when a classifier gets retrained. The y axis (Metrics) reports the value obtained by each metric with the associated batch size

Observation 2. *Regardless of the users' characteristics and how a workout is composed, the workout quality depends above all on how much the runners stick to their workout objectives and how much effort they are putting in during workouts. Apart from being adherent to the goals set by the coach, the period of the year when the workouts are planned can also influence the performance of runners; we conjecture that this last phenomenon means that good weather positively influences workout quality.*

6.4.4 Re-training simulation

In this evaluation, we re-train and assess the effectiveness of the classifiers for every 100 workouts.

Considering ordinal classification, Fig. 15 shows that OR maintains a good performance over time, with the F2-score values ranging between 81% and 99%. A peculiarity of this classifier is that it can predict effectively even when training on a subset of workouts. We can also remark that the performance of the model changes a lot with the variation of the batch size. This change in terms of performance could be associated with the fact that with the growth of the batch size, some of the data

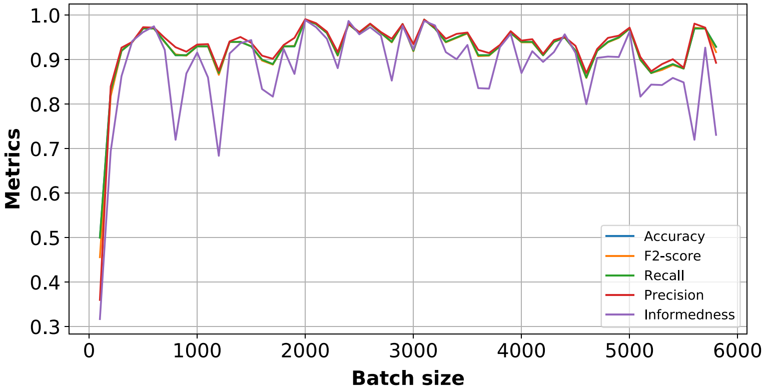


Fig. 16 Evolution of the performance of RF. The x axis (Batch size) contains a point every 100 workouts, that is when a classifier gets retrained. The y axis (Metrics) reports the value obtained by each metric with the associated batch size

used for the model evaluation could contain workouts performed by new users that were not present in the training data; since we know very little about those users, this decreases the performance of the model.

Considering the most effective multi-class classifier, Fig. 16 shows that when training on fewer workouts (less than 300) the performance of RF is low but, when training on 300 workouts or more, the classifier maintains a good performance over time (F2-score ranges between 86% and 99%).

6.4.5 Ranking under fairness constraints

For each coach, we started by ranking the users that performed the first five workouts, and we updated the rankings for every new five workouts. We do this for the last 50 workouts performed by the users followed by all the coaches. Then, we mitigated unfairness for each ranking update w.r.t. our disparate treatment metrics, DPR and DTR.

- *Ranking quality*. Figure 17 shows the evolution of the average NDCG@10 over time before and after mitigating unfairness in ordinal and multi-class-based rankings. We notice that before mitigating unfairness, all the rankings achieved an NDCG@10 of 1 using both multi-class-based and ordinal-based rankings. This means that the ratings predicted by the classifier reflect both workout performance and the timeliness with which athletes should be contacted. This is also reflecting the fact that the classifiers are predicting correctly the athletes' performance as shown by the results of the previous sections. After mitigating unfairness, we remark that the values of NDCG@10 get lower as the number of ranking updates grows. This could be explained by the fact that, while we are mitigating unfairness, we reorder the users such that they get assisted in a fair way and this influences the quality of rankings. However, we can see that after mitigating unfairness, the ordinal-based rankings maintain a slightly higher NDCG@10 than the multi-class-based ones.

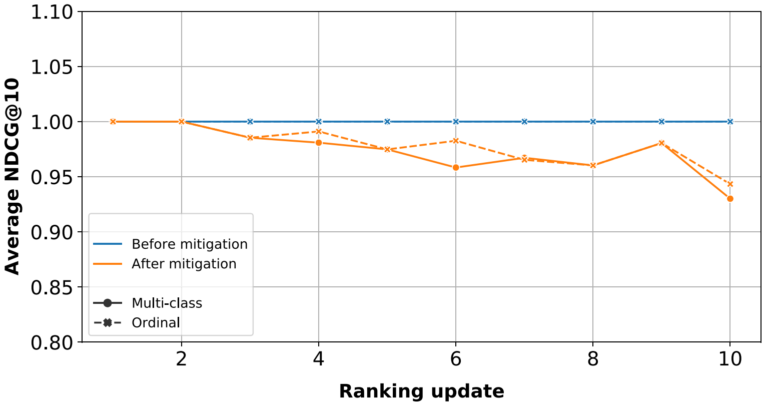


Fig. 17 Ranking accuracy results. The x axis (Ranking update) contains a point every 5 workouts, that is when a classifier gets retrained. The y axis (Average NDCG@10) reports the NDCG@10 obtained by each classifier in the associated ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

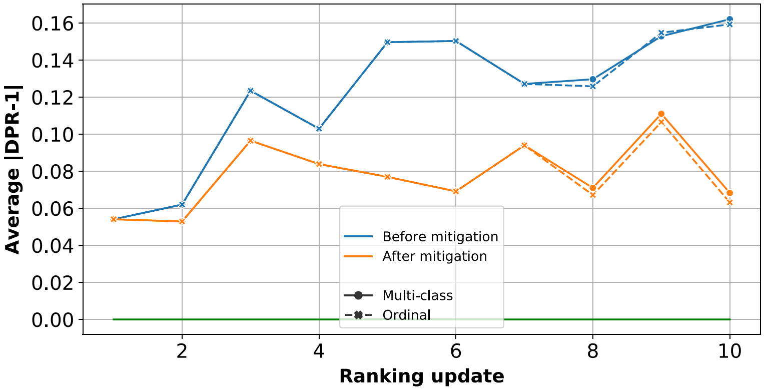


Fig. 18 Fairness in terms of demographic parity. The x axis (Ranking update) contains a point every 5 workouts, that is when a classifier gets retrained. The y axis (Average $|DPR - 1|$) reports the distance of each classifier with respect to the expected DPR score in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

– *Global evolution of DPR and DTR over time before and after mitigating unfairness.* Figure 18 illustrates the evolution of the average $|DPR - 1|$ (how far is DPR w.r.t. its perfect value) before and after mitigating unfairness in ordinal and multi-class-based rankings for each ranking update. From Fig. 18, we see that over time the average $|DPR - 1|$ gets closer to 0 after applying Algorithm 1 to mitigate unfairness, but its values get higher as the number of ranking updates grows. The values of $|DPR - 1|$ for ordinal and multi-class-based rankings are very similar, though, ordinal-based rankings achieved a slightly better DTR compared to multi-class-based rankings.

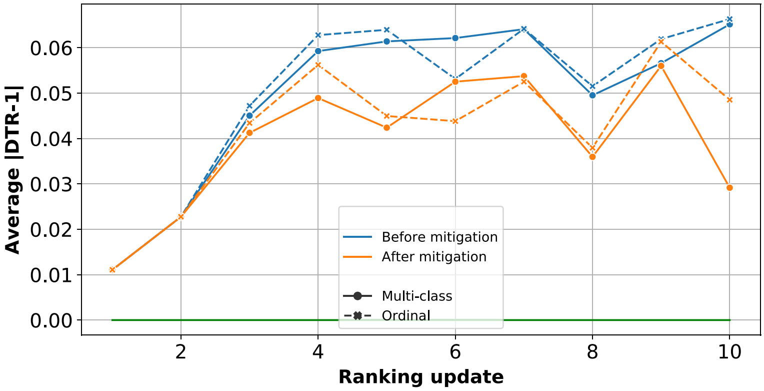


Fig. 19 Fairness in terms of disparate treatment. The x axis (Ranking update) contains a point every 5 workouts, that is when a classifier gets retrained. The y axis (Average $|DTR-1|$) reports the distance of each classifier with respect to the expected DTR score in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

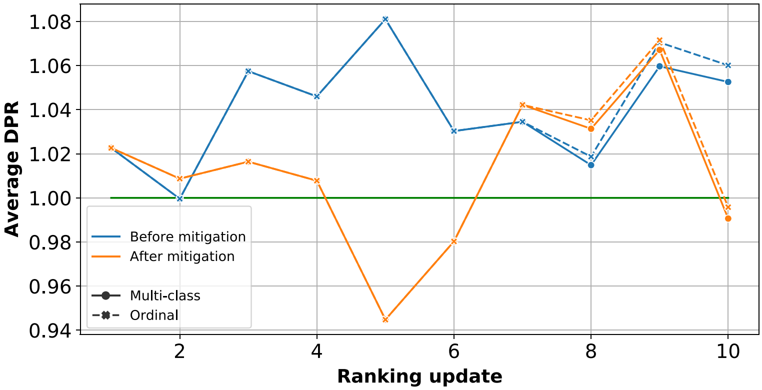


Fig. 20 Demographic parity scores. The x axis (Ranking update) contains a point every five workouts, that is when a classifier gets retrained. The y axis (Average DPR) reports raw DPR score returned in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

Figure 19 illustrates the evolution of the average $|DTR-1|$ (how far is DTR w.r.t. its perfect value) before and after mitigating unfairness in ordinal and multi-class-based rankings for each ranking update. From Fig. 18, we see that, like $|DPR-1|$, over time the average $|DTR-1|$ gets closer to 0 after applying Algorithm 1 to mitigate unfairness, but its values get higher as the number of ranking updates grows. Nevertheless, the values of $|DTR-1|$ are closer to 0 comparing to the values of $|DPR-1|$ before and after mitigating unfairness.

In contrast with what we have seen in Fig. 18, according to DTR, the multi-class-based strategy is the one that generates more fair rankings.

Figure 20 illustrates the evolution of DPR before and after mitigating unfairness in ordinal and multi-class-based rankings for each ranking update. From Fig. 20, we see

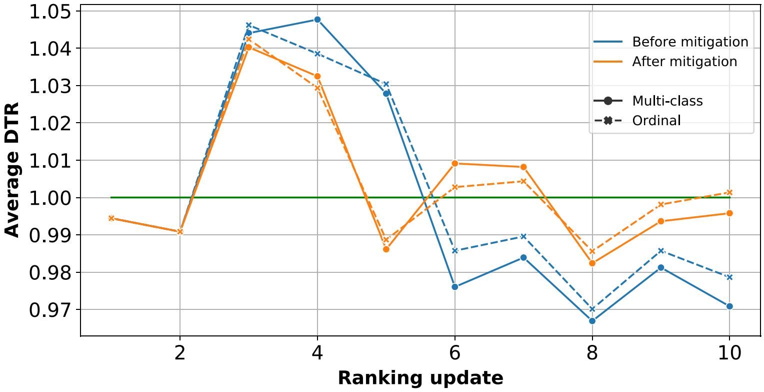


Fig. 21 Disparate treatment scores. The x axis (Ranking update) contains a point every five workouts, that is when a classifier gets retrained. The y axis (Average DTR) reports raw DTR score returned in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

that the average DPR mostly ranges between 0.95 and 1.09 for all the ranking updates in ordinal and multi-class-based rankings. Before mitigating unfairness, the values of DPR are ranging between 1 and 1.09 and, after unfairness mitigation, the values are ranging between 0.95 and 1.06. We can deduce that the values of DPR after mitigation vary more, but are closer to 1.

In addition, we can notice that in the majority of cases the discriminated gender in terms of demographic parity is the male gender ($DPR > 1$).

Figure 21 illustrates the evolution of DTR before and after mitigating unfairness in ordinal and multi-class-based rankings for each ranking update. Figure 21 shows that the average DTR values mostly range between 0.97 and 1.05 for all the ranking updates in ordinal and multi-class-based rankings. Before mitigating unfairness, the values of DTR are ranging between 0.97 and 1.05; meanwhile, after the mitigation, the values are ranging between 0.98 and 1.04. We can deduce that the values of DTR after mitigation are less variate compared to the values of DPR and closer to 1. Furthermore, we notice that for almost all the ranking updates, the ordinal classification-based rankings are achieving better results compared to multi-class classification-based rankings with respect to DTR.

Moreover, we notice that, in terms of disparate treatment, the discriminated gender is the female gender ($DTR < 1$), unlike what we have observed earlier when assessing fairness using DPR.

At this point, one may pose the question: *Which metric is telling the truth about the discriminated gender?* Both metrics are somehow right about the discriminated group, except that DPR is not considering the performance of athletes when measuring unfairness, while DTR includes also the utility of the rankings instead, and thus considers also the performance of athletes when assessing unfairness. For this reason, we may consider that DTR is more suited to our application's context, especially for the fact that not considering the utility of rankings when mitigating unfairness could influence negatively the quality of the users' experience by attributing athletes an exposure that is not proportional to their performance during their last workout session.

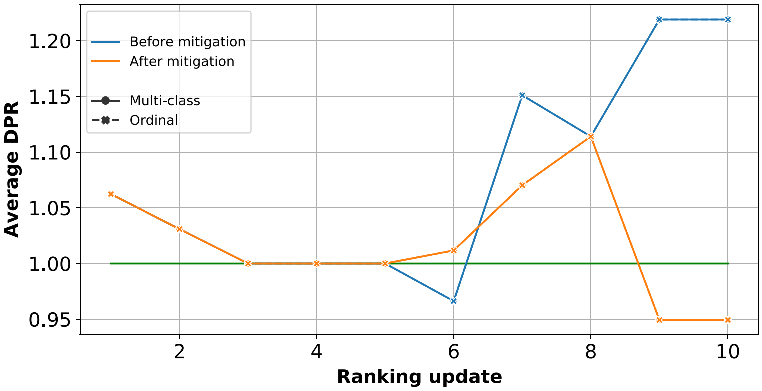


Fig. 22 Demographic parity scores, when females are more than males. The x axis (Ranking update) contains a point every five workouts, that is when a classifier gets retrained. The y axis (Average DPR) reports raw DPR score returned in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

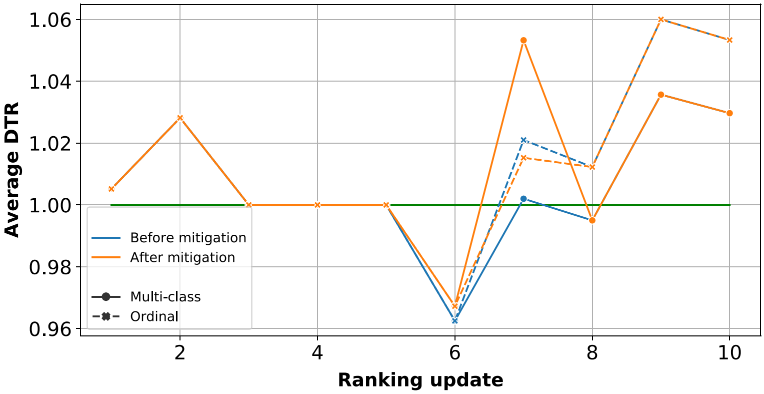


Fig. 23 Disparate treatment scores, when females are more than males. The x axis (Ranking update) contains a point every five workouts, that is when a classifier gets retrained. The y axis (Average DTR) reports raw DTR score returned in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

To explore more in depth this phenomena, we represented graphically the evolution of DPR and DTR before and after mitigating unfairness in all the rankings where females are more than males and the ones where females are more than males (Figs. 22, 23, 24, and 25). We analyze our results in our following analysis.

- *Evolution of DPR and DTR over time before and after mitigating unfairness when females are more than males*. Figure 22 illustrates the evolution of DPR in the rankings where females are more than males, before and after mitigating unfairness in ordinal and multi-class-based rankings for each ranking update. We observe that in most cases the discriminated group is the male group, which is the minority group in this case, and that the minority group gets ranked in a more unfair way as the

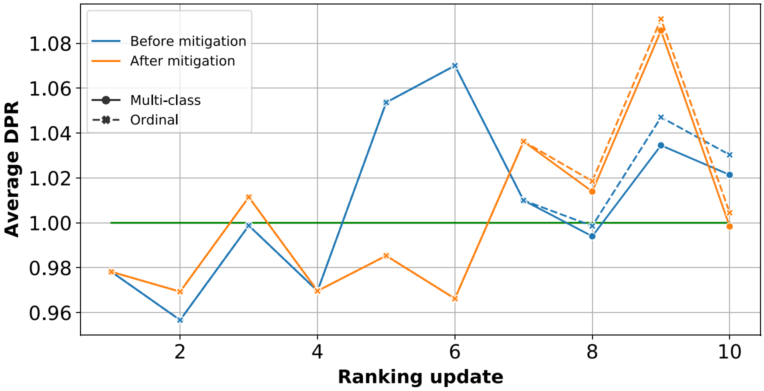


Fig. 24 Demographic parity scores, when females are less than males. The x axis (Ranking update) contains a point every five workouts, that is when a classifier gets retrained. The y axis (Average DPR) reports raw DPR score returned in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

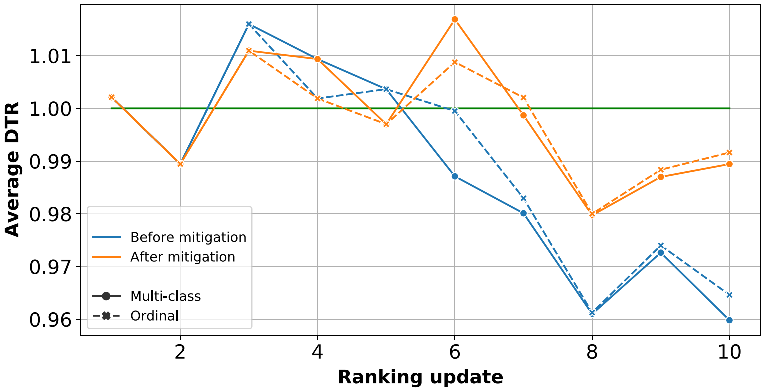


Fig. 25 Disparate treatment scores, when females are less than males. The x axis (Ranking update) contains a point every five workouts, that is when a classifier gets retrained. The y axis (Average DTR) reports raw DTR score returned in a ranking update. We report these results before and after mitigation (blue and orange line, respectively), for multi-class and ordinal classifiers (continuous and dashed lines, respectively)

number of ranking updates increases. This could be explained by the fact that, as the rankings are updated, the number of ranked users gets larger and more diverse which makes the original rankings more unfair. After mitigating unfairness, the average DPR gets closer to its perfect value, and the discriminated group could change for some ranking updates. Hence, we do not observe any difference in terms of DPR values when comparing ordinal and multi-class-based ranking strategies.

Figure 23 illustrates the evolution of DTR in the rankings where females are more than males, before and after mitigating unfairness in ordinal and multi-class-based rankings for each ranking update. According to DTR, the discriminated group is mostly the male one, and after the mitigation of unfairness, the average DTR got closer to 1 in all the cases for the ordinal classification-based ranking strategy. Instead, for the multi-class classification-based rankings, we can notice that in one case the average DTR is higher than the average DTR after mitigating unfairness, and this could be explained by the fact that the values of DTR for that ranking update are more variate than before mitigating unfairness.

- *Evolution of DPR and DTR over time before and after mitigating unfairness when females are less than males*. Figure 24 illustrates the evolution of DPR in the rankings where females are less than males, before and after mitigating unfairness in ordinal and multi-class classification-based rankings for each ranking update. This figure shows that the discriminated group according to DPR before mitigating unfairness is the male group, when it appears that after mitigating unfairness the discriminated group is mostly the female one. Since the values of average DPR before and after mitigating unfairness are very close for the multi-class and the ordinal classification-based ranking strategies. Figure 25 illustrates the evolution of DTR in the rankings where females are less than males, before and after mitigating unfairness in ordinal and multi-class classification-based rankings for each ranking update. According to the average DTR, the discriminated group for both ranking strategies is mostly the female one of females before and after mitigating unfairness.

Observation 3. *The discriminated gender when assessing fairness using DTR coincides with the gender of the minority group. This phenomenon aligns our work with what is usually observed in the fairness literature, where the demographic group representing the minority in the training data is the discriminated one (Boratto et al. 2021).*

7 Conclusions and future work

In this paper, we proposed and validated an approach to identify and rank athletes that need timely support due to low performance in workouts and recommend them to their coaches so that they can be contacted with a higher priority. Furthermore, we guarantee a fair exposure in the ranking, to make sure that users of different groups have equal opportunities to get supported. Our approach models the performance and running behavior of the users, in order to apply a ranking algorithm to recommend users to coaches, according to their performance in the last running session and the quality of the previous ones. Then, we presented a re-ranking algorithm to provide fair exposure to users.

The results show the effectiveness of our ranking algorithm even under fairness constraints, which allow us to provide unbiased ranking w.r.t. the users' sensitive attributes without losing a lot in terms of ranking utility.

The limitations of this work are related to the following perspectives:

- In this study, we only considered one dataset, however, no other works in the literature studied these phenomena, so no similar dataset exists;
- The available dataset is small but, with our modeling, we managed to achieve very good results;

- Our prediction tasks are based on existing classification models. While models tailored on our domain and prediction task would be relevant and their development is left as future work, this work has allowed us to establish effective and fair user rankings in this domain.

As future work, we look to introduce explainability and coach-in-the-loop insights to improve the recommendations. Furthermore, we are currently preparing a live user evaluation, to see how coaches perceive our ranking and the fairness dimensions we introduced in this work.

Funding Open access funding provided by Università degli Studi di Cagliari within the CRUI-CARE Agreement.

References

Ahire, S.B., Khanuja, H.K.: A personalized framework for health care recommendation. In: 2015 International Conference on Computing Communication Control and Automation, pp. 442–445. IEEE (2015)

Amatriain, X., Basilico, J.: Recommender systems in industry: A netflix case study. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 385–419. Springer (2015). https://doi.org/10.1007/978-1-4899-7637-6_11

Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. **6**(1), 20–29 (2004)

Berndsen, J., Smyth, B., Lawlor, A.: Pace my race: recommendations for marathon running. In: Bogers, T., Said, A., Brusilovsky, P., Tikk, D. (eds.) Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019, pp. 246–250. ACM (2019). https://doi.org/10.1145/3298689.3346991

Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E.H., Goodrow, C.: Fairness in recommendation ranking through pairwise comparisons. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2019., pp. 2212–2220. ACM (2019). https://doi.org/10.1145/3292500.3330745

Biega, A.J., Gummadi, K.P., Weikum, G.: Equity of attention: Amortizing individual fairness in rankings. In: Collins-Thompson, K., Mei, Q., Davison, B.D., Liu, Y., Yilmaz, E. (eds.) The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, pp. 405–414. ACM (2018). https://doi.org/10.1145/3209978.3210063

Binns, R.: Fairness in machine learning: Lessons from political philosophy. In: Friedler, S.A., Wilson, C. (eds.) Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA, *Proceedings of Machine Learning Research*, vol. 81, pp. 149–159. PMLR (2018). http://proceedings.mlr.press/v81/binns18a.html

Boratto, L., Carta, S., Iguider, W., Mulas, F., Pilloni, P.: Predicting workout quality to help coaches support sportspeople. In: Elsweiler, D., Ludwig, B., Said, A., Schäfer, H., Torkamaan, H., Trattner, C. (eds.) Proceedings of the 3rd International Workshop on Health Recommender Systems, HealthRecSys 2018, co-located with the 12th ACM Conference on Recommender Systems (ACM RecSys 2018), Vancouver, BC, Canada, October 6, 2018, *CEUR Workshop Proceedings*, vol. 2216, pp. 8–12. CEUR-WS.org (2018). http://ceur-ws.org/Vol-2216/healthRecSys18_paper_2.pdf

Boratto, L., Carta, S., Mulas, F., Pilloni, P.: An e-coaching ecosystem: design and effectiveness analysis of the engagement of remote coaching on athletes. Pers. Ubiquit. Comput. **21**(4), 689–704 (2017). https://doi.org/10.1007/s00779-017-1026-0

Boratto, L., Fenu, G., Marras, M.: Interplay between upsampling and regularization for provider fairness in recommender systems. User Model. User Adapt. Interact. **31**(3), 421–455 (2021). https://doi.org/10.1007/s11257-021-09294-8

Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)

Brooks, J.P., Dulá, J.H.: The l1-norm best-fit hyperplane problem. Appl. Math. Lett. **26**(1), 51–55 (2013)

Brown, I., Mues, C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst. Appl. **39**(3), 3446–3453 (2012)

Burke, R., Sonboli, N., Ordonez-Gauger, A.: Balanced neighborhoods for multi-sided fairness in recommendation. In: Conference on Fairness, Accountability and Transparency, FAT 2018, *Proceedings of Machine Learning Research*, vol. 81, pp. 202–214. PMLR (2018). http://proceedings.mlr.press/v81/burke18a.html

Carbonell, J.G., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336. ACM (1998). https://doi.org/10.1145/290941.291025

Celis, L.E., Straszak, D., Vishnoi, N.K.: Ranking with fairness constraints. In: 45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, *LIPIcs*, vol. 107, pp. 28:1–28:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018). https://doi.org/10.4230/LIPIcs.ICALP.2018.28

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)

Dobrican, R., Zampuniéris, D.: A proactive solution, using wearable and mobile applications, for closing the gap between the rehabilitation team and cardiac patients. In: 2016 IEEE International Conference on Healthcare Informatics, ICHI 2016, Chicago, IL, USA, October 4-7, 2016, pp. 146–155. IEEE Computer Society (2016). https://doi.org/10.1109/ICHI.2016.23

Donciu, M., Ionita, M., Dascalu, M., Trausan-Matu, S.: The runner - recommender system of workout and nutrition for runners. In: Wang, D., Negru, V., Ida, T., Jebelean, T., Petcu, D., Watt, S.M., Zaharie, D. (eds.) 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2011, Timisoara, Romania, September 26-29, 2011, pp. 230–238. IEEE Computer Society (2011). https://doi.org/10.1109/SYNASC.2011.18

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S.: Fairness through awareness. In: Goldwasser, S. (ed.) Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012, pp. 214–226. ACM (2012). https://doi.org/10.1145/2090236.2090255

Fenu, G., Lafhouli, H., Marras, M.: Exploring algorithmic fairness in deep speaker verification. In: Gervasi, O., Murgante, B., Misra, S., Garau, C., Blecic, I., Taniar, D., Apduhan, B.O., Rocha, A.M.A.C., Tarantino, E., Torre, C.M., Karaca, Y. (eds.) Computational Science and Its Applications - ICCSA 2020 - 20th International Conference, Cagliari, Italy, July 1-4, 2020, Proceedings, Part IV, *Lecture Notes in Computer Science*, vol. 12252, pp. 77–93. Springer (2020). https://doi.org/10.1007/978-3-030-58811-3_6

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recogn. **44**(8), 1761–1776 (2011)

Gao, X., Feng, Y.: Penalized weighted least absolute deviation regression. Stat. Interface **11**(1), 79–89 (2018)

Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Mach. Learn. **63**(1), 3–42 (2006)

Geyik, S.C., Ambler, S., Kenthapadi, K.: Fairness-aware ranking in search and recommendation systems with application to linkedin talent search. In: Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., Karypis, G. (eds.) Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, pp. 2221–2231. ACM (2019). https://doi.org/10.1145/3292500.3330691

Guy, I., Pizzato, L.: People recommendation tutorial. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp. 431–432. ACM (2016). https://doi.org/10.1145/2959100.2959196

Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Lee, D.D., Sugiyama, M., von Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-

10, 2016, Barcelona, Spain, pp. 3315–3323 (2016). https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

He, Q., Agu, E., Strong, D.M., Tulu, B.: Recfit: a context-aware system for recommending physical activities. In: Gupta, S.K.S., Banerjee, A. (eds.) Proceedings of the 1st Workshop on Mobile Medical Applications, MMA '14, Memphis, Tennessee, USA, November 3-6, 2014, pp. 34–39. ACM (2014). https://doi.org/10.1145/2676431.2676439

Hutson, J.A., Taft, J.G., Barocas, S., Levy, K.: Debiasing desire: Addressing bias and discrimination on intimate platforms. Proc. ACM Hum. Comput. Interact. **2**(CSCW), 73:1–73:18 (2018). https://doi.org/10.1145/3274342

Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Recommendation independence. In: Conference on Fairness, Accountability and Transparency, FAT 2018, *Proceedings of Machine Learning Research*, vol. 81, pp. 187–201. PMLR (2018). http://proceedings.mlr.press/v81/kamishima18a.html

Kamphorst, B.A.: E-coaching systems - what they are, and what they aren't. Pers. Ubiquit. Comput. **21**(4), 625–632 (2017). https://doi.org/10.1007/s00779-017-1020-6

Khwaja, M., Ferrer, M., Iglesias, J.O., Faisal, A.A., Matic, A.: Aligning daily activities with personality: towards a recommender system for improving wellbeing. In: Bogers, T., Said, A., Brusilovsky, P., Tikk, D. (eds.) Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019, pp. 368–372. ACM (2019). https://doi.org/10.1145/3298689.3347020

Klein, M.C.A., Manzoor, A.R., Middelweerd, A., Mollee, J.S., te Velde, S.J.: Encouraging physical activity via a personalized mobile system. IEEE Int. Comput. **19**(4), 20–27 (2015). https://doi.org/10.1109/MIC.2015.51

Klement, W., Wilk, S., Michaowski, W., Matwin, S.: Dealing with severely imbalanced data. In: Proceedings of the PAKDD Conference, p. 14. Citeseer (2009)

Kroeze, W., Werkman, A., Brug, J.: A systematic review of randomized trials on the effectiveness of computer-tailored education on physical activity and dietary behaviors. Ann. Behav. Med. **31**(3), 205–223 (2006)

Landschoot, S., Waegeman, W., Audenaert, K., Haesaert, G., De Baets, B.: Ordinal regression models for predicting deoxynivalenol in winter wheat. Plant. Pathol. **62**(6), 1319–1329 (2013)

Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(17), 1–5 (2017). http://jmlr.org/papers/v18/16-365

Marras, M., Boratto, L., Ramos, G., Fenu, G.: Equality of learning opportunity via individual fairness in personalized recommendations. Int. J. Artif. Intell. Educ. (2021). https://doi.org/10.1007/s40593-021-00271-1

Marras, M., Korus, P., Memon, N.D., Fenu, G.: Adversarial optimization for dictionary attacks on speaker verification. In: G. Kubin, Z. Kacic (eds.) Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019, pp. 2913–2917. ISCA (2019). https://doi.org/10.21437/Interspeech.2019-2430

Martin, C.K., Gilmore, L.A., Apolzan, J.W., Myers, C.A., Thomas, D.M., Redman, L.M.: Smartloss: a personalized mobile health intervention for weight management and health promotion. JMIR Mhealth Uhealth **4**(1), e18 (2016)

McMurray, J., Adamopoulos, S., Anker, S., Auricchio, A., Bohm, M., Dickstein, K et al.: esc guidelines for the diagnosis and treatment of acute and chronic heart failure 2012: The task force for the diagnosis and treatment of acute and chronic heart failure 2012 of the european society of cardiology. developed in collaboration with the heart failure association (hfa) of the esc (1787)

Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., Diaz, F.: Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness and satisfaction in recommendation systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, pp. 2243–2251. ACM (2018). https://doi.org/10.1145/3269206.3272027

Nassabi, M.H., op den Akker, H., Vollenbroek-Hutten, M.M.R.: An ontology-based recommender system to promote physical activity for pre-frail elderly. In: Butz, A., Koch, M., Schlichter, J.H. (eds.) Mensch and Computer 2014 - Workshopband, 14. Fachübergreifende Konferenz für Interaktive und Kooperative Medien - Interaktiv unterwegs - Freiräume gestalten, 31. August - 3. September 2014, München, Germany, pp. 181–184. De Gruyter Oldenbourg (2014). https://dl.gi.de/20.500.12116/8167

Patro, G.K., Biswas, A., Ganguly, N., Gummadi, K.P., Chakraborty, A.: Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In: WWW '20: The Web Conference 2020, pp. 1194–1204. ACM / IW3C2 (2020). https://doi.org/10.1145/3366423.3380196

Petsani, D., Konstantinidis, E.I., Bamidis, P.D.: Designing an e-coaching system for older people to increase adherence to exergame-based physical activity. In: Bamidis, P.D., Ziefle, M., Maciaszek, L.A. (eds.) Proceedings of the 4th International Conference on Information and Communication Technologies for Ageing Well and e-Health, ICT4AWE 2018, Funchal, Madeira, Portugal, March 22-23, 2018, pp. 258–263. SciTePress (2018). https://doi.org/10.5220/0006821502580263

Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation (2011)

Powers, D.M.: The problem with kappa. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 345–355. Association for Computational Linguistics (2012)

Radlinski, F., Craswell, N.: Comparing the sensitivity of information retrieval metrics. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pp. 667–674. ACM (2010)

Rathore, S.S., Kumar, S.: A decision tree logic based recommendation system to select software fault prediction techniques. Computing **99**(3), 255–285 (2017)

Rennie, J.D.: Ordinal Logistic Regression. MIT (2005)

Rennie, J.D., Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels. In: Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling, vol. 1. Citeseer (2005)

Sáez, J.A., Krawczyk, B., Woźniak, M.: Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. Pattern Recogn. **57**, 164–178 (2016)

Santos-Gago, J.M., Sabucedo, L.Á., González-Maciel, R., Rorís, V.M.A., García-Soidán, J.L., Wanden-Berghe, C., Sanz-Valero, J.: Towards a personalised recommender platform for sportswomen. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) New Knowledge in Information Systems and Technologies - Volume 1, World Conference on Information Systems and Technologies, WorldCIST 2019, Galicia, Spain, 16-19 April, 2019, *Advances in Intelligent Systems and Computing*, vol. 930, pp. 504–514. Springer (2019). https://doi.org/10.1007/978-3-030-16181-1_48

Sanz-Cruzado, J., Castells, P.: Contact Recommendations in Social Networks, chap. Chapter 16, pp. 519–569 (2019). https://doi.org/10.1142/9789813275355_0016

Singh, A., Joachims, T.: Fairness of exposure in rankings. In: Guo, Y., Farooq, F. (eds.) Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018, London, UK, August 19-23, 2018, pp. 2219–2228. ACM (2018). https://doi.org/10.1145/3219819.3220088

Smyth, B.: Recommender systems: A healthy obsession. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pp. 9790–9794. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33019790

Smyth, B., Cunningham, P.: An analysis of case representations for marathon race prediction and planning. In: Cox, M.T., Funk, P., Begum, S. (eds.) Case-Based Reasoning Research and Development - 26th International Conference, ICCBR 2018, Stockholm, Sweden, July 9-12, 2018, Proceedings, *Lecture Notes in Computer Science*, vol. 11156, pp. 369–384. Springer (2018). https://doi.org/10.1007/978-3-030-01081-2_25

Smyth, B., Cunningham, P.: Marathon race planning: A case-based reasoning approach. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, pp. 5364–5368. ijcai.org (2018). https://doi.org/10.24963/ijcai.2018/754

Topal, M., Eyduran, E., Yağanoğlu, A., Sönmaz, A., Keskin, S., et al.: Use of ridge and principal component regression analysis methods in multicollinearity. Journal of the Faculty of Agriculture of Atatürk University (Turkey) (2010)

Tseng, J.C.C., Lin, B., Lin, Y., Tseng, V.S., Day, M., Wang, S., Lo, K., Yang, Y.: An interactive healthcare system with personalized diet and exercise guideline recommendation. In: Conference on Technologies and Applications of Artificial Intelligence, TAAI 2015, Tainan, Taiwan, November 20-22, 2015, pp. 525–532. IEEE (2015). https://doi.org/10.1109/TAAI.2015.7407106

Yang, K., Stoyanovich, J.: Measuring fairness in ranked outputs. In: Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, June 27-29, 2017, pp. 22:1–22:6. ACM (2017). https://doi.org/10.1145/3085504.3085526

Yom-Tov, E., Feraru, G., Kozdoba, M., Mannor, S., Tennenholtz, M., Hochberg, I.: Encouraging physical activity in patients with diabetes: intervention using a reinforcement learning system. J. Med. Int. Res. **19**(10), e338 (2017)

Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fair: A fair top-k ranking algorithm. In: Lim, E., Winslett, M., Sanderson, M., Fu, A.W., Sun, J., Culpepper, J.S., Lo, E., Ho, J.C., Donato, D., Agrawal, R., Zheng, Y., Castillo, C., Sun, A., Tseng, V.S., Li, C. (eds.) Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06–10, 2017, pp. 1569–1578. ACM (2017). https://doi.org/10.1145/3132847.3132938

Zehlike, M., Castillo, C.: Reducing disparate exposure in ranking: A learning to rank approach. In: WWW '20: The Web Conference 2020, pp. 2849–2855. ACM / IW3C2 (2020). https://doi.org/10.1145/3366424.3380048