SN Computer Science (2024) 5:10
https://doi.org/10.1007/s42979-023-02342-7

ORIGINAL RESEARCH

Adaptive Boosting Method for Mitigating Ethnicity and Age Group Unfairness

Ivona Colakovic[1] · **Sašo Karakatič**[1]

Received: 5 November 2022 / Accepted: 20 September 2023

Abstract
Machine learning algorithms make decisions in various fields, thus influencing people's lives. However, despite their good quality, they can be unfair to certain demographic groups, perpetuating socially induced biases. Therefore, this paper deals with a common unfairness problem, unequal quality of service, that appears in classification when age and ethnicity groups are used. To tackle this issue, we propose an adaptive boosting algorithm that aims to mitigate the existing unfairness in data. The proposed method is based on the AdaBoost algorithm but incorporates fairness in the calculation of the instance's weight with the goal of making the prediction as good as possible for all ages and ethnicities. The results show that the proposed method increases the fairness of age and ethnicity groups while maintaining good overall quality compared to traditional classification algorithms. The proposed method achieves the best accuracy in almost every sensitive feature group. Based on the extensive analysis of the results, we found that when it comes to ethnicity, interestingly, *White* people are likely to be incorrectly classified as not being heroin users, whereas other groups are likely to be incorrectly classified as heroin users.

Keywords Fairness · Classification · Boosting · Machine learning

Introduction

The efficiency of machine learning (ML) in automating human tasks is indisputable. The wide use of machine learning shows us its capability and potential to be utilized for solving different tasks. It is used in different fields, such as medicine, finance, logistics, etc. It can perform well in various tasks, such as image and speech recognition, classification, anomaly detection, etc.

ML's value comes from processing a significant amount of data, extracting knowledge from it, and doing it all faster than humans. Discovering patterns in data is the power of ML, especially when the number of important patterns is big and can be overlooked by humans when decisions are made. Although ML is powerful in learning, it does not have power over what it is learning since it learns from data, which is an entirely human product.

Extracting patterns from data and learning from them can, apart from good classification quality, lead to unexpected results. Poor quality data can, despite the data processing methods, highly impact the model, making it learn societal bias. This was seen in some real-world ML-based applications, where the face recognition model worked much better on white men's images than others. Similarly, Google's photo application labelled black men as "gorillas", questioning the quality of ML models used [1].

Fair Machine Learning

The complexity of ML models has grown over the years in order to improve their performance. The ambiguous models that lack transparency in decision-making do not allow the evaluation of their decisions from different aspects, thus leaving us to rely on standard evaluation methods. Although they have shown good classification quality, they can also be unfair. Unfair decisions lead to discrimination of certain groups of samples with the same characteristics, having

This article is part of the topical collection "Advances on Data Science, Technology and Applications" guest edited by Slimane Hammoudi, Alfredo Cuzzocrea and Oleg Gusikhin.

✉ Ivona Colakovic
ivona.colakovic@um.si

Sašo Karakatič
saso.karakatic@um.si

[1] Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, Maribor 2000, Slovenia

Published online: 15 November 2023

an even bigger impact when the samples represent people. For some time now, the law has defined that when humans are making decisions, certain human traits cannot be deal-breakers, thus preventing discrimination. Recently, laws have defined ML-based decision-making, expecting that models are not discriminating.

The term fairness is not strictly defined by psychologists, causing the lack of a standard mathematical definition of fairness, thus making the evaluation of it complicated [2, 3]. There are now more than 20 different measures of fairness. Fairness metrics can address individual or group fairness. Most group fairness metrics are based on statistical parity, measuring each group's possibility of having positive outcomes [4]. In this work, we focus on group fairness and view fairness as equal quality of service. In terms of fairness, the model should offer equal quality of service, allowing the same classification quality for all groups included in sensitive features.

Sensitive or *protected* features are features that should not be deal-breakers in decision-making, mostly defined by the law. Frequently used sensitive features are gender, race, religion, skin color, age, marital status, etc. Sensitive features could be removed, but some work showed that this approach does not always automatically create fair models [5]. Furthermore, removing them could erase important information that can be vital to decision-making, for example, in medicine. Different values of sensitive feature define categories, which we call *sensitive feature group*. Previous studies usually worked with one binary sensitive feature, where it has only two sensitive feature groups, being far from real-world applications that are more likely to have multiple categories. For example, ethnicity can have multiple categories (White, Black, Asian, Other). In this work, we deal with sensitive features that have multiple categories in order to bring it closer to practical use.

Removing sensitive features or not, their values could be reflected in some other feature or a combination of different features. That could happen due to a number of reasons that come from the state of the original dataset. Utilizing data pre-processing techniques can only improve it to some extent. However, an algorithmic approach is needed to track and prevent learning of unfair patterns.

Existing Literature

Fairness in ML has been an emerging topic in the last few years. Work has been published on improving fairness in pre-processing, post-processing, and in-processing phases. We focus on in-processing methods that address fairness during the learning phase. Furthermore, we build upon the existing work, which uses an ensemble of classifiers to address fairness. To improve fairness in decision trees, Fair Forests [6] were introduced. The authors propose adapting how information gain is calculated to consider the sensitive feature. This approach shows improvement in decision trees, but we aim at an iterative process, allowing the ensemble to correct fairness through iterations. To achieve that, instead of using uncorrelated classifiers in an ensemble, we use the boosting technique. The first similar approach [7] was published in 2015, where fairness in the Census Income dataset with the boosted classifier was introduced. Relabelling instances according to the fairness rules focuses on individual fairness. In our work, we focus on group fairness. Using cumulative fairness to mitigate unfairness, AdaFair [8] was introduced. This approach also tackles the problem of class-imbalanced data. The weight adaption step is changed to consider a model's confidence score and equalized odds. Our approach uses the fairness of each sensitive feature group to update the weights. Fair-AdaBoost [9] was proposed with a new error rate and classifier weight, that takes into account sensitive feature. It also deals with hyper-parameter optimization using genetic algorithms. However, this approach uses the binary sensitive feature, whereas our approach can handle categorical sensitive features. Lastly, our previous work [10] proposes Fair AdaBoost to improve fairness in the Drugs dataset. The approach presented adapts how weights are updated in a way that considers fairness. We introduced a new way of measuring the fairness of each sensitive feature group as the difference between the maximum accuracy of any group and the accuracy of a certain group. The model was one of the first to deal with categorical sensitive features.

Contributions

This paper is an extension of our previous work. In the previous paper, we proposed a boosting algorithm to overcome unfairness in the Drug consumption dataset [10]. Here, we perform a more extensive analysis, where we compare more algorithms to our approach and inspect the results of each sensitive feature group in detail.

Therefore, our contribution is the following: *We provide extensive analysis of binary classification results on the Drug consumption dataset with a focus on fairness, while examining the impact of incorrectly classified instances.*

The rest of this paper is organized as follows. In "Methodology" section the proposed algorithm and the motivation behind it are described. In "Experiment" section, the experimental setup is presented, followed by the results and a detailed analysis of it. Finally, "Conclusion" section concludes the study and discusses the future work.

Methodology

Combining more models improves classification performance, suggesting the potential of ensemble models. Different techniques combine models' outputs, such as voting, bagging, or boosting. Boosting has shown outstanding performance due to the iterative process, where in each iteration new model is trying to improve the previous models' mistakes.

One of the algorithms that showed excellent performance in classification tasks is AdaBoost, introduced in 1995 [11]. AdaBoost (Adaptive Boosting) algorithm combines weak learners to create a strong one. Base models are created in iterations, where each of them tries to learn from his own mistakes made in previous iterations, thus boosting his knowledge. At the end of each iteration, a weight in the form of its accuracy is signed to an estimator, making the final output a weighted sum of all estimators.

AdaBoost utilizes instance weight in order to improve performance through iterations. Each instance is assigned an equal weight at the beginning, which is then adapted through iterations. The instance weight of misclassified instances is increased to help the next estimator focus more on more demanding instances. The estimation error is used to adapt weights as long as the perfect estimator with estimation error 0 or a certain number of iterations is achieved.

However, the good performance of the AdaBoost algorithm, which is considered to be *the best off-shelf classifier in the world* [12], is achieved only when standard classification metrics are used, such as accuracy, F-score, etc. On the other hand, the evaluation of fairness in the AdaBoost algorithm did not show promising results.

Fair AdaBoost

To mitigate unfairness in classification, we propose an extension of the AdaBoost algorithm [13] called Fair AdaBoost. Same as AdaBoost, the main phase of Fair AdaBoost is a boosting phase, in which weights are updated until the optimal result is achieved. Instance weight adaptation is an iterative process where estimation error and fairness are considered. Increased weight is assigned to incorrectly classified instances to help the model focus on more challenging instances in the next iteration. The estimation error according to which weights are updated represents the balance between estimation error in accuracy and estimation error of sensitive feature group instance belongs to. As well as in AdaBoost, weights are updated until a certain number of iterations or a perfect estimator is achieved.

The most significant difference between AdaBoost and Fair AdaBoost is that Fair AdaBoost takes into consideration fairness in the boosting stage. As well as in AdaBoost, the weights of all instances are equal in the beginning, so the weight value is $1/S$ with S being the number of instances. The process of boosting in Fair AdaBoost described next is presented as pseudo-code in Algorithm 1. Next, boosting is performed in n number of iterations that are set as one of the initializing parameters of the algorithm. Each iteration estimator learns from train data with weights from previous iterations, upon which it makes predictions. Based on those predictions and ground truth, the accuracy and fairness of each sensitive feature group are calculated. Fairness of kth sensitive group is calculated as shown in Eq. 1, with acc_{max} being the highest accuracy any of the sensitive feature groups achieve and acc_k the accuracy of the kth group.

$$\text{fairness}_k = \frac{\text{acc}_{\text{max}}}{\text{acc}_k} \tag{1}$$

The calculation of estimator error *err* shown in Eq. 2 represents a balance between error in accuracy and fairness. The estimator's accuracy in the ith iteration is denoted as $\text{acc}_{\text{global}}$. The input parameter w_f is fairness weight, and acc_{diff} is the difference between the maximum and minimum accuracy of any sensitive feature groups. If the estimator error achieves 0 before performing a given number of iterations, boosting is stopped.

$$\text{err} = (1 - \text{acc}_{\text{global}}) \times (1 - w_f) + \text{acc}_{\text{diff}} \times w_f \tag{2}$$

Lastly, the boosting stage finishes with the weight adaptation step defined in Eq. 3, where the weight of every instance calculated by original AdaBoost denoted as $w_{i,j(AB)}$ is multiplied by the fairness of sensitive feature group to which the jth instance belongs to. That gives a new weight $w_{i,j}$ of jth

Algorithm 1: Fair AdaBoost weights boosting stage.

$w_0 = 1/S$ ▷ S is a number of instances
for $i = 1, ..., n$ **do** ▷ n is a number of iterations
  $learn(data, w_{i-1})$
  $predict(X)$
  $calculate\ accuracy$
  $calculate\ fairness\ per\ group\ as\ in\ Equation\ 1$
  $calculate\ estimator\ error\ as\ in\ Equation\ 2$
  $update\ weights\ according\ to\ Equation\ 3$
end for

instance in the ith iteration, which is then used in the next iteration.

$$w_{i,j} = w_{i,j(AB)} \times \text{fairness}_k, \quad j \in K \tag{3}$$

Experiment

To evaluate the proposed method, we performed an experiment on the UCI Drug consumption dataset [14] using 5-cross validation. For comparison, we use a support vector machine (SVC), Naive Bayes, Logistic Regression, a Decision Tree (CART), and the base of the proposed algorithm, AdaBoost. For implementation, we use Python, `scikit-learn`, and `fairlearn` libraries. All the above-mentioned algorithms have default parameter values of the scikit-learn library. Boosting algorithms contain 50 decision trees, thus performing boosting in 50 iterations. The experimental setup is shown in Table 1.

We evaluate performance with standard classification metrics like accuracy, F-Score, TPR, TNR, and the maximum and minimum accuracy and F-Score of sensitive feature groups. For fairness purposes, we observe the difference between the maximum and the minimum accuracy and F-Score achieved by any sensitive feature group. We also use fairness metrics Demographic Parity Difference (DMP) and Equalized Odds Difference (EQOD). DMP deems fairness as equal chances for each sensitive feature group to have a positive outcome. On the other hand, EQOD states that the chances of instances in a positive class being correctly classified as positive and an instance in a negative class being incorrectly classified as positive should be similar [15].

Dataset

We used the Drug consumption dataset, which is also used for unfairness problems by other authors [16, 17]. The dataset described in Table 2 is a collection of responses to a survey [14] conducted in 2017, where participants stated their frequency of drug usage. 1885 people participated in the survey, and each of them is described with 12 personal features and a response to the frequency of using all the 18 drugs mentioned in the study. Each drug feature is a categorical feature with the following possible answers: "Never Used", "Used over a Decade Ago", "Used in Last Decade", "Used in Last Year", "Used in Last Month", "Used in Last Week" and "Used in Last Day". From this data, multiple problems can be defined, where we opted for binary classification of heroin usage. To this end, we transformed the target to binary form with possible values of "Used" and "Not Used". Different personal attributes can be used as a sensitive feature, out of which we chose and separately tested age and ethnicity.

Table 1 Experimental setup

Parameters	Values
Number of estimators	50
Algorithm	SAMME.R
Base estimator	CART
Learning rate	1
Fairness weight	0.5

Table 2 Dataset description

	Dataset	
Instances	1885	
Attributes	32	
Sensitive feature	Age	Ethnicity
Class ratio (+:-)	1: 5.73	
Positive class	Used	

Results

In this section, we present the results of performed experiments. We divide the results per sensitive features used, age, and ethnicity, and also in classification metrics considered better when higher and fairness metrics considered better when lower.

In Fig. 1, we can see the results when age is used as a sensitive feature. The results closer to 1 are better results. We can see that AdaBoost and Fair AdaBoost are outperforming other algorithms in almost all metrics. Furthermore, Fair AdaBoost achieves the best accuracy with a score of 0.9 and an F-Score. Fair AdaBoost is also best at classifying positive cases, whereas Naive Bayes best classifies negative cases.

The following Fig. 2 shows the results of an experiment with ethnicity as a sensitive feature. Once again, Fair AdaBoost shows superior performance, being the best in almost every metric. Moreover, it is the best in classifying positive cases, whereas Naive Bayes is the best in classifying negative cases.

In Fig. 3, the results of the experiment, when the sensitive feature is age, are shown. The values shown are better when closer to 0. We observe that SVC and Fair AdaBoost achieve notably better results in DMP and EQOD. Regarding maximum group difference, Fair AdaBoost achieves the best accuracy group difference, whereas SVC achieves the best F-Score group difference.

The experiment's results regarding fairness, when ethnicity was used as a sensitive feature, are shown in Fig. 4. In this case, Logistic Regression, SVC, and Fair AdaBoost

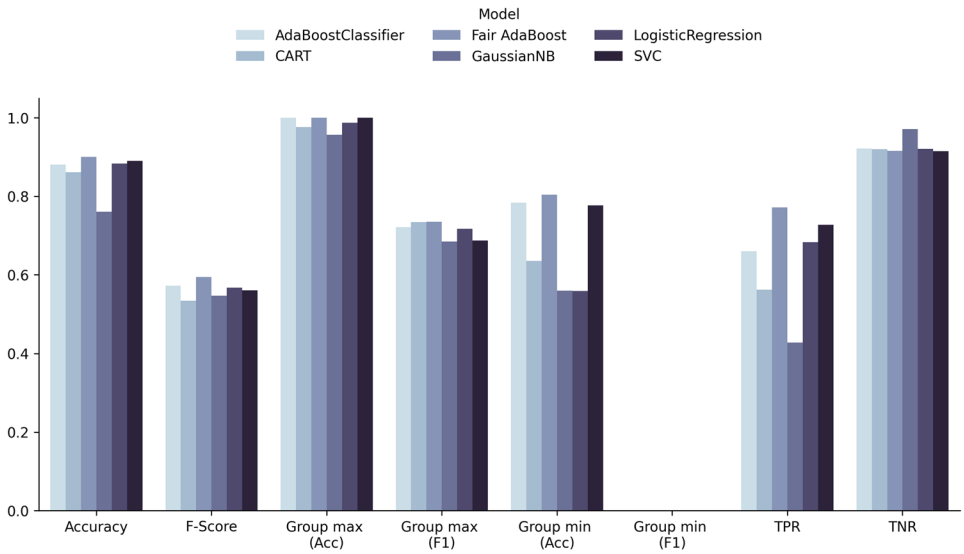

Fig. 1 Evaluation of classification of Drug consumption dataset using attribute age as a sensitive attribute (higher values represent better results)

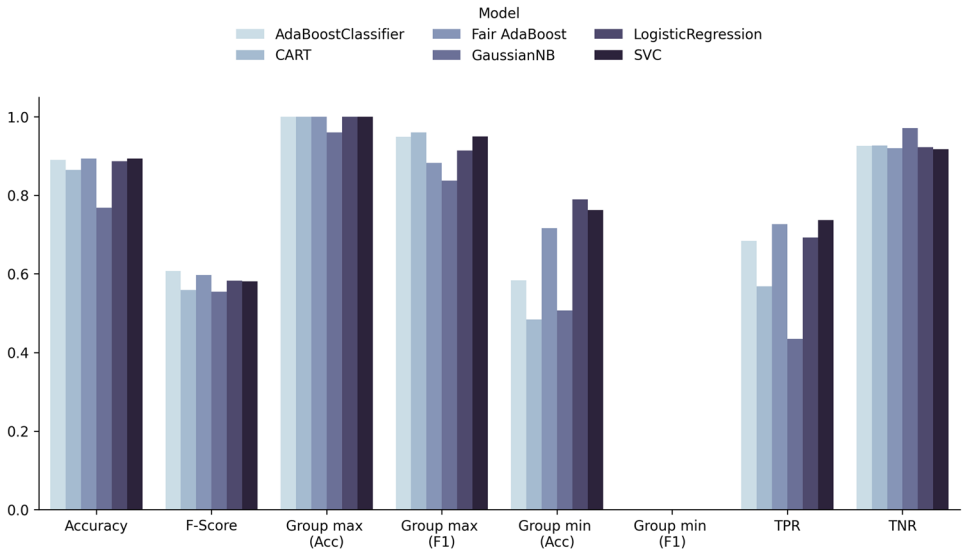

Fig. 2 Evaluation of classification of Drug consumption dataset using attribute ethnicity as a sensitive attribute (higher values represent better results)

achieve the best results in DMP and EQOD. The best accuracy group difference is achieved by Logistic regression, whereas Naive Bayes achieves the best F-Score group difference. In both metrics, Fair AdaBoost does not fall much behind the best algorithm.

In-Depth Analysis of Sensitive Feature Groups' Performance

We further analyze each category of both sensitive features to better understand the models' outcomes. We evaluate their accuracy and present some instances that are incorrectly classified only by AdaBoost and misclassified by both AdaBoost and Fair AdaBoost. With this, we open the discussion of biased outcomes to sensitive features, the importance of the instances incorrectly classified as negative and incorrectly classified as positive, and their harmful potential.

The accuracy of each sensitive feature group, when the sensitive feature is age, is shown in Fig. 5. Although it may seem that Logistic regression and Naive Bayes achieve better-balanced results, we notice that all groups achieve notably lower accuracy, affecting the overall accuracy, especially in comparison with AdaBoost and Fair AdaBoost. It is worth mentioning that AdaBoost and Fair AdaBoost achieve 100% accuracy in the *65+ group*, while Fair AdaBoost achieves slightly better accuracy in the rest of the groups.

Figure 6 shows the accuracy of sensitive feature groups with ethnicity as a sensitive feature. The results show significantly lower accuracy of AdaBoost and Naive Bayes in the *Mixed-Black/Asian* group. Fair AdaBoost achieves 100%

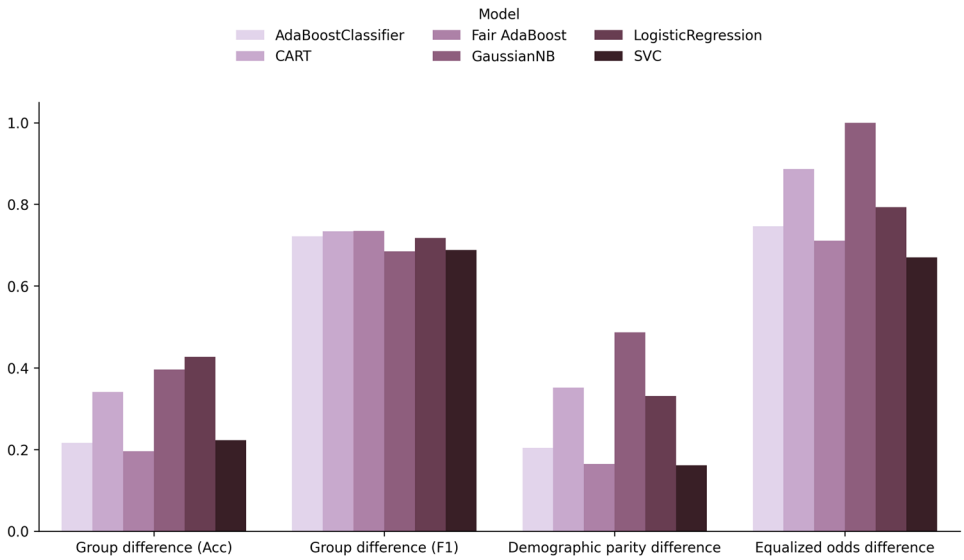

Fig. 3 Evaluation of classification of Drug consumption dataset using attribute age as a sensitive attribute (lower values represent better results)

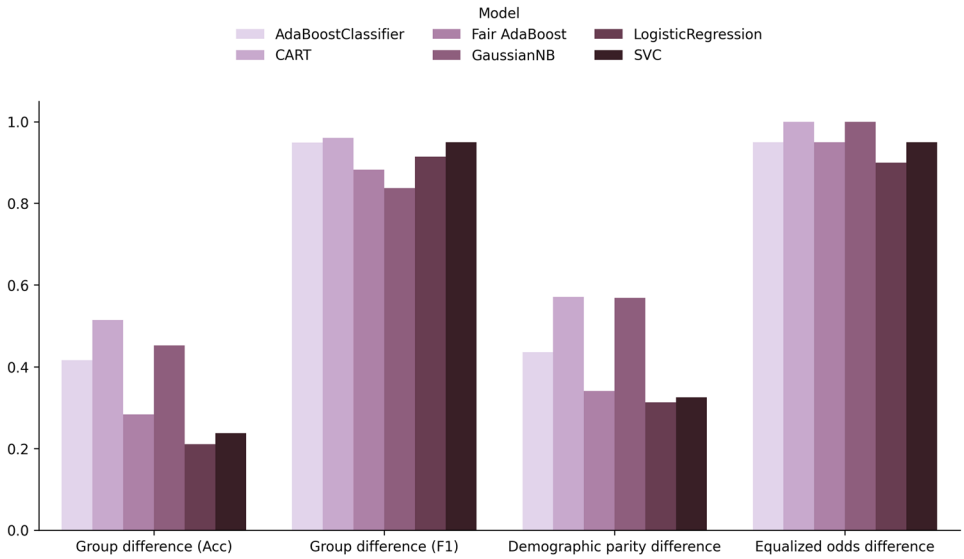

Fig. 4 Evaluation of classification of Drug consumption dataset using attribute ethnicity as a sensitive attribute (lower values represent better results)

accuracy in *Mixed-Black/Asian* group and 95% accuracy in the *Black* group.

Analysis of Error Cases

By analyzing error cases, we unveil a few incorrectly classified instances by AdaBoost and Fair AdaBoost. We compare instances that are misclassified by only AdaBoost, while being correctly classified by Fair AdaBoost, and misclassified instances by both AdaBoost and Fair AdaBoost. We further discuss the sensitive feature groups of false positive and false negative cases and their implications. The index of instances does not correspond to the number of instances in the dataset but is created and used for internal analysis. This is not the analysis of individual fairness but rather an insight into the sensitive feature groups' performance through their representatives.

We recall that the target feature is the use of heroin, where True corresponds to "Used Heroin" and False to "Not Used Heroin". Table 3 shows instances incorrectly classified by AdaBoost and correctly by Fair AdaBoost when the sensitive feature is ethnicity. We can see that AdaBoost incorrectly classifies the white group as negative. Meaning, that AdaBoost classifies the *White* group as having never used heroin, even though they did. On the other hand, the algorithm misclassifies other groups, such as *Mixed-White/Asian*, *Mixed-Black/Asian*, and *Other*, as positive. It classifies those groups as heroin users, whereas they never used it. From this, we can see how AdaBoost outcomes are unfair to certain demographic groups.

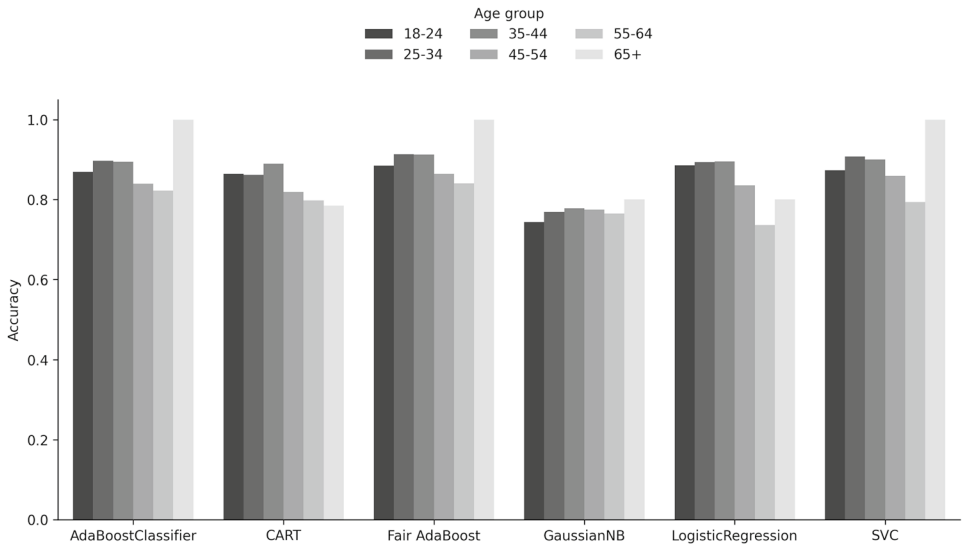

Fig. 5 Accuracy of each sensitive feature group achieved by algorithms for classification of Drug consumption dataset using age as a sensitive attribute

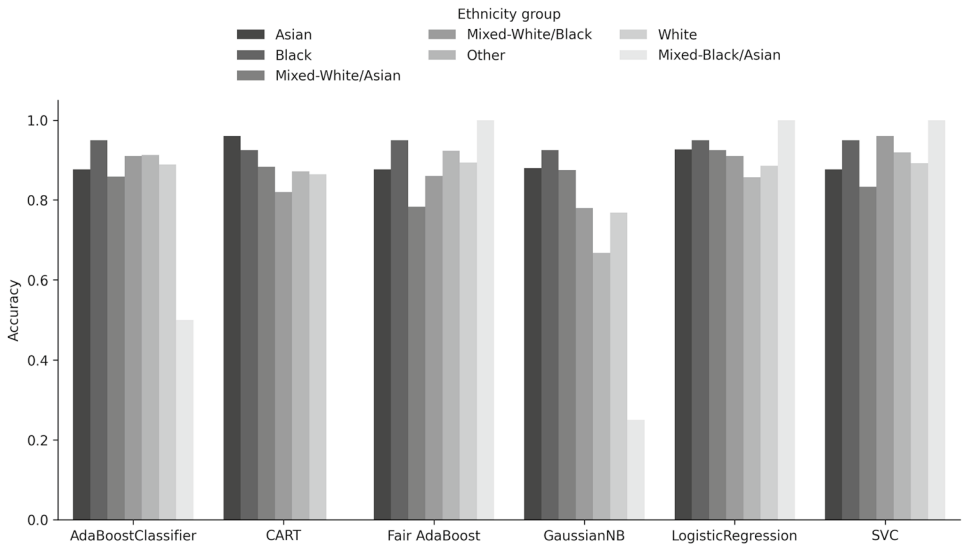

Fig. 6 Accuracy of each sensitive feature group achieved by algorithms for classification of Drug consumption dataset using ethnicity as a sensitive attribute

Table 3 View of a few instances incorrectly classified by AdaBoost and correctly classified by Fair AdaBoost when ethnicity as a sensitive feature

Instance	Sensitive feature group	Ground truth	AdaBoost prediction	Fair AdaBoost prediction
#134	Mixed-White/Asian	False	True	False
#1462	Mixed-Black/Asian	False	True	False
#217	White	True	False	True
#333	White	True	False	True
#908	Other	False	True	False

Table 4 View of a few instances incorrectly classified by AdaBoost and Fair AdaBoost when ethnicity as a sensitive feature

Instance	Sensitive feature group	Ground truth	AdaBoost prediction	Fair AdaBoost prediction
#23	White	True	False	False
#58	White	True	False	False
#1503	Mixed-White/Asian	False	True	True
#744	Mixed-White/Asian	True	False	False
#914	Mixed-White/Black	True	False	False

Table 5 View of a few instances incorrectly classified by AdaBoost and correctly classified by Fair AdaBoost when age as a sensitive feature

Instance	Sensitive feature group	Ground truth	AdaBoost prediction	Fair AdaBoost prediction
#645	18–24	False	True	False
#134	25–34	False	True	False
#439	35–44	False	True	False
#429	45–54	False	True	False
#360	55–64	True	False	True

Table 6 View of a few instances incorrectly classified by AdaBoost and Fair AdaBoost when age as a sensitive feature

Instance	Sensitive feature group	Ground truth	AdaBoost prediction	Fair AdaBoost prediction
#1845	18–24	True	False	False
#198	25–34	True	False	False
#410	35–44	False	True	True
#23	45–54	True	False	False
#81	55-64	True	False	False

Table 4 shows a few instances incorrectly classified by AdaBoost and Fair AdaBoost when the sensitive feature is ethnicity. In 4 out of 5 cases, we see that algorithms classify samples as false negatives. Those instances belong to the following groups: *White*, *Mixed-White/Asian*, and *Mixed-White/Black*. Interestingly, they both misclassify another instance of a *Mixed-White/Asian* group, but in this case, they incorrectly classify it as positive. Compared to previous instances that are misclassified only by AdaBoost, we can see that usually *White* race, even if it appears in some of the *Mixed* groups, can be incorrectly classified as non-user.

Next, we look at the instances incorrectly classified by AdaBoost and correctly classified by Fair AdaBoost when the sensitive feature is age, shown in Table 5. We observe that AdaBoost incorrectly classifies younger groups as positive, i.e., heroin users. On the other hand, it misclassifies instances of the *55–64* age group as negative. This means that AdaBoost can perceive younger people as heroin users while the elders as non-users.

Lastly, we show instances misclassified by both AdaBoost and Fair AdaBoost when the sensitive feature is age in Table 6. In 4 out of 5 cases, algorithms classify instances as false negatives. However, we cannot conclude which groups algorithms lean to, since out of four groups, it misclassifies the youngest two and the oldest two groups. The instance of the middle age group *35–44* is classified as false positive.

From instances misclassified only by AdaBoost, we could see a glimpse of bias and potential harm to specific groups. In neither case, the algorithm misclassifies an instance of the *65+* age group, achieving 100% accuracy in the oldest age group. Instances incorrectly classified by both AdaBoost and Fair AdaBoost can be considered more challenging than the others. This especially stands for instances that appear in misclassified instances of both sensitive features, such as instance *#23*.

Conclusion

In this work, we tackle the common unfairness problem in machine learning. Certain human traits used in decision-making can heavily influence the outcomes of ML models due to various reasons. These features, called sensitive features, should not be deal-breakers, and equal quality of service should be allowed regardless of the sensitive feature group. We address the unfairness in the UCI Drug consumption dataset, which uses the ethnicity and age of individuals as features. To this end, we propose Fair AdaBoost, based on AdaBoost, which considers fairness in the instance weights adaptation stage. While updating instance weights, we take into account fairness, presented as a difference between the maximum accuracy of any sensitive feature group and the accuracy of the group instances belongs to. We evaluate this proposed method with a binary classification task, determining if the person ever used heroin.

The results showed that Fair AdaBoost outperforms other algorithms regarding fairness and standard classification metrics as well. In both experiments with different sensitive features, some algorithms achieve similar results in some metrics, but we mostly focus on comparison with AdaBoost. The good results Fair AdaBoost achieves in standard classification metrics show that it is improving fairness and keeping the overall good quality AdaBoost already has.

We further analyze the results of algorithms by each sensitive feature group. We compare the accuracy of each sensitive group that algorithms achieve to find that Fair AdaBoost achieves high results in every group. In both experiments, it achieves 100% accuracy in at least one group, namely the *65+* age group and *Mixed-Black/Asian* ethnicity group. Then, we compare a few instances from both experiments that were incorrectly classified only by AdaBoost and misclassified by both AdaBoost and Fair AdaBoost. We discuss the difference between false positive and false negative cases, their sensitive feature group, and the ramifications of it. Interestingly, when ethnicity is used as a sensitive feature group, *White* people are usually classified as false negatives, whereas other groups are classified as false positives.

In the future, this approach should be evaluated on more than one dataset for more conceivable results. Different base estimators in the ensemble could be used since, in this work, we examine Fair AdaBoost with only the CART decision

tree as a base estimator. The results showed that fairness could be incorporated into the boosting technique, which is why it would be interesting to see it incorporated into other algorithms, such as XGBoost. Also, the proposed method should be evaluated along with other competing fair ensemble classifiers on multiple datasets.

Funding The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

Data availability The data used are public datasets and are available on the web.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Not applicable.

References

1. Buolamwini J. Opinion | When the Robot Doesn't See Dark Skin. The New York Times. Chap. Opinion. 2018. Accessed 18 Aug 2022.
2. Barocas S, Hardt M, Narayanan A. Fairness and Machine Learning. fairmlbook.org, 2019.
3. Verma S, Rubin J. Fairness definitions explained. In: Proceedings of the International Workshop on Software Fairness, 2018; pp. 1–7. ACM, Gothenburg Sweden. https://doi.org/10.1145/3194770.3194776.
4. Binns R. On the Apparent Conflict Between Individual and Group Fairness. arXiv:1912.06883 [cs, stat] 2019. arXiv: 1912.06883. Accessed 31 Mar 2022.
5. Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through Awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. ITCS '12, pp. 214–226. Association for Computing Machinery, New York, NY, USA 2012. https://doi.org/10.1145/2090236.2090255. event-place: Cambridge, Massachusetts.
6. Raff E, Sylvester J, Mills S. Fair Forests: Regularized Tree Induction to Minimize Model Bias. arXiv:1712.08197 [cs, stat] 2017. arXiv: 1712.08197. Accessed 22 Mar 2022.
7. Fish B, Kun J, Lelkes AD. Fair Boosting: a Case Study. In: Workshop on Fairness, Accountability, and Transparency in Machine Learning, 2015; p. 5.
8. Iosifidis V, Ntoutsi E. AdaFair: Cumulative Fairness Adaptive Boosting. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019; 781–790 https://doi.org/10.1145/3357384.3357974. arXiv: 1909.08982. Accessed 21 Mar 2022.
9. Huang X, Li Z, Jin Y, Zhang W. Fair-AdaBoost: Extending AdaBoost method to achieve fair classification. Expert Syst Appl. 2022;202: 117240. https://doi.org/10.1016/j.eswa.2022.117240.
10. Colakovic I, Karakatič S. Improved Boosted Classification to Mitigate the Ethnicity and Age Group Unfairness. In: Proceedings of the 11th International Conference on Data Science, Technology and Applications - DATA, 2022 pp. 432–437. SciTePress, New York. https://doi.org/10.5220/0011287400003269. Backup Publisher: INSTICC ISSN: 2184-285X.
11. Freund Y, Schapire RE. A desicion-theoretic generalization of online learning and an application to boosting. In: Vitányi P, editor. Computational Learning Theory. Berlin, Heidelberg: Springer; 1995. p. 23–37.
12. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting. Ann Stat. 2000;28(2):337–407. https://doi.org/10.1214/aos/1016218223.
13. Hastie T, Rosset S, Zhu J, Zou H. Multi-class AdaBoost. Statistics and Its. Interface. 2009;2(3):349–60. https://doi.org/10.4310/SII.2009.v2.n3.a8.
14. Fehrman E, Muhammad AK, Mirkes EM, Egan V, Gorban AN. The Five Factor Model of personality and evaluation of drug consumption risk. 2017. arXiv:1506.06297 [stat] arXiv: 1506.06297. Accessed 2022-04-06.
15. Bird S, Dudík M, Edgar R, Horn B, Lutz R, Milan V, Sameki M, Wallach H, Walker K. Fairlearn: A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft May 2020.
16. Mehrabi N, Naveed M, Morstatter F, Galstyan A. Exacerbating Algorithmic Bias through Fairness Attacks. arXiv:2012.08723 [cs] 2020. arXiv: 2012.08723. Accessed 5 Apr 2022.
17. Donini M, Oneto L, Ben-David S, Shawe-Taylor J, Pontil M. Empirical Risk Minimization under Fairness Constraints. arXiv: 1802.08626 [cs, stat] 2020. arXiv: 1802.08626. Accessed 6 Apr 2022.