Information Processing and Management 60 (2023) 103226

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Debiaser for Multiple Variables to enhance fairness in classification tasks

Giordano d'Aloisio *, Andrea D'Angelo, Antinisca Di Marco, Giovanni Stilo

University of L'Aquila, Department of Information Engineering and Information Sciences and Mathematics, L'Aquila, Italy

ARTICLE INFO

Keywords:
Machine learning
Bias and Fairness
Multi-class classification
Preprocessing algorithm
Equality

ABSTRACT

Nowadays assuring that *search* and *recommendation* systems are fair and do not apply discrimination among any kind of population has become of paramount importance. This is also highlighted by some of the sustainable development goals proposed by the United Nations. Those systems typically rely on machine learning algorithms that solve the classification task. Although the problem of fairness has been widely addressed in binary classification, unfortunately, the fairness of multi-class classification problem needs to be further investigated lacking well-established solutions. For the aforementioned reasons, in this paper, we present the *Debiaser for Multiple Variables (DEMV)*, an approach able to mitigate *unbalanced groups* bias (i.e., bias caused by an unequal distribution of instances in the population) in both binary and multi-class classification problems with multiple sensitive variables. The proposed method is compared, under several conditions, with a set of well-established baselines using different categories of classifiers. At first we conduct a specific study to understand which is the best generation strategies and their impact on DEMV's ability to improve fairness. Then, we evaluate our method on a heterogeneous set of datasets and we show how it overcomes the established algorithms of the literature in the multi-class classification setting and in the binary classification setting when more than two sensitive variables are involved. Finally, based on the conducted experiments, we discuss strengths and weaknesses of our method and of the other baselines.

1. Introduction

Bias impacts human beings as individuals or groups characterized by a set of legally-protected sensitive attributes (e.g., their race, gender, or religion) by under-representing them or by representing them in a wrong manner. If not managed, the inequalities reinforced by search and recommendation algorithms can lead to *severe societal consequences*, such as discrimination and unfairness (Hajian, Bonchi, & Castillo, 2016). Both *search* and *recommendation* algorithms provide users with ranked results that fit and match their needs and interests. Both tasks often convey and strengthen bias in terms of *imbalances* and *inequalities*, mainly if they rely on or encompass machine learning algorithms as those which solve classification problems. For this reason, assuring that search and recommendation systems are fair and do not apply discrimination among any kind of population has become of paramount importance, mainly because they are pervasive in several domains (Amigó, Deldjoo, Mizzaro, & Bellogín, 2023; Boratto & Marras, 2021) - e.g., justice (Redmond & Baveja, 2002), health care (Street, Wolberg, & Mangasarian, 1993), education (Austin, Christopher, & Dickerson, 2016), etc.

* Corresponding author.
E-mail addresses: giordano.daloisio@graduate.univaq.it (G. d'Aloisio), andrea.dangelo6@student.univaq.it (A. D'Angelo), antinisca.dimarco@univaq.it (A. Di Marco), giovanni.stilo@univaq.it (G. Stilo).

https://doi.org/10.1016/j.ipm.2022.103226
Received 28 April 2022; Received in revised form 16 November 2022; Accepted 5 December 2022
Available online 21 December 2022

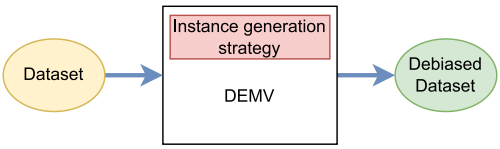


Fig. 1. Application of DEMV.

The importance of having *fair* and *egalitarian* systems must be a fundamental step to solving some of the 17 sustainable development goals proposed by the United Nations.[1] In particular, we will possibly rely on information systems to accomplish goals 5 (gender equality) and 10 (reduced inequalities) on a large scale. If those information systems include some AI or Machine learning techniques (such as classification and recommendation), it will be essential to identify and mitigate properly algorithmic *Bias and Fairness*.

Over the years, many recommendation systems have been proposed that rely on multi-class classification systems. For instance, consider the one proposed by Baskota and Ng (2018) to recommend admissions to a graduate school, the one proposed by Yanes, Mostafa, Ezz, and Almuayqil (2020) to improve the learning experience, the one proposed by Suchithra and Pai (2018) to recommend fertilizers, the one proposed by Meenachi, Ramakrishnan, Sivaprakash, Thangaraj, and Sethupathy (2022) for crop recommendation, or the one proposed by Zhang, Cao, Gross, and Zaiane (2013) to recommend physical therapy. These systems embed a multi-class classifier to solve multi-class classification tasks. Assuring *fairness* of these systems is also essential to achieving some of the aforementioned sustainable goals: i.e., goal 2 (zero huger) (Meenachi et al., 2022; Suchithra & Pai, 2018), goal 3 (good health and well-being) (Zhang et al., 2013), and goal 4 (quality education) (Baskota & Ng, 2018; Yanes et al., 2020). In Stitini, Kaloun, and Bencharef (2022) the authors even highlight how integrating multi-class classification into context-aware recommendation systems can improve the overall recommendation. Different methods have been proposed to mitigate bias at several levels of data processing for both classification and recommender systems (Caton & Haas, 2020; Mehrabi, Morstatter, Saxena, Lerman, & Galstyan, 2021). However, we notice that, despite the fact that it is widely adopted and constitutes a building block for personalization and search systems, the multi-class classification problem is still not effectively addressed (Jiang, Liu, Ding, Liang, & Duan, 2017). In addition to fairness, a system must also have high accuracy of the predictions in order to be usable in the real world. However, most of the time the mitigation of bias negatively influences the accuracy of the predictions (Caton & Haas, 2020). This trade-off must be managed, for example, by properly generating or modifying the instances of a dataset in a pre-processing context or by properly modifying the behavior of a method in an in-processing context.

For the aforementioned reasons, in this work, we focused, as a building block of other information access systems, on a bias mitigation method capable of (i) managing an arbitrary number of sensitive variables in the multi-class classification scenario (ii) preserving the accuracy of the predictions.

In the detail, to conduct our research and to better address the problem of bias mitigation in multi-class classification, we formulate the following four research questions (RQ) that will help to highlight the fundamental findings and novel contributions of this paper.

RQ1. What are the strengths and limitations of current existing approaches addressing bias mitigation in multi-class classification problems?
In this paper, we analyze three baselines designed to mitigate bias in multi-class classification problems, namely *Exponentiated Gradient* and *Grid Search* methods from Agarwal, Beygelzimer, Dudik, Langford, and Wallach (2018), and the *Blackbox* method from Putzel and Lee (2022). To the best of our knowledge, these are the only ones implemented for the multi-class classification task. To highlight their strengths and weaknesses, we apply each method to a heterogeneous set of binary and multi-class datasets that are widely used in research (extensively discussed in Section 4.2).

RQ2. How can we design a novel approach that goes beyond the existing baselines?
To overcome some of the limitations of the analyzed baseline, we present an improved version of *Debiaser for Multiple Variables (DEMV)* presented in d'Aloisio, Stilo, Di Marco, and D'Angelo (2022). DEMV is a generalization of the *Sampling* algorithm proposed by Kamiran and Calders (2012). DEMV is model- and data-agnostic, allowing its introduction in already existing systems without particular effort and without introducing structural changes.
DEMV is the first proposed pre-processing method to mitigate bias caused by an unequal distribution of instances in the population (i.e. *unbalanced groups* bias) in an agnostic way in both binary and multi-class classification considering multiple sensitive variables. As highlighted in Fig. 1, DEMV takes as input a generic dataset and returns in output the debiased dataset without considering the classifier involved in the task. We implement DEMV with a plug-in approach where the user can select different *Instance generation strategies*. The source code is provided on GitHub and on the Territori Aperti RI.

RQ3. How can DEMV keep a high level of accuracy while improving fairness?
Since DEMV is a pre-processing algorithm, the fairness and accuracy trade-off can be managed by better manipulating the instances of the dataset. Since our approach tackles the *unbalanced groups* bias, this means generating new instances that are

[1] https://sdgs.un.org/goals

coherent with the existing ones in terms of values and distribution. We plug-in in DEMV three different generating strategies, namely *Uniform*,[2] *SMOTE* (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) and *ADASYN* (He, Bai, Garcia, & Li, 2008). In order to evaluate the influence of each strategy on DEMV's ability to enhance the fairness and accuracy of the classifier, we extensively evaluate DEMV by employing nine datasets extensively used in literature (see Section 4.2). We perform this analysis both in binary and multi-class settings.

RQ4. In which conditions does DEMV goes beyond the existing baselines?
To answer this question, we run a set of experiments whose aim is to evaluate the performance of DEMV in improving fairness while keeping a high level of accuracy. In particular, we evaluate it in binary and multi-class classification problems and consider sensitive groups identified by up to three sensitive variables. To demonstrate the wide validity of DEMV, we employ in the experiments an heterogeneous set of ML classifiers, namely Logistic Regression, Multi-Layer Perceptrons, Gradient Boosting Classifier and SVC. As we show in Section 4.4, DEMV outperforms the baselines in the binary task in the case of more than two sensitive variables, while it remains competitive with one or two sensitive variables. In the case of the multi-class task, DEMV outperforms the baselines in every setting (i.e., number of sensitive variables) for all the considered datasets. Finally, DEMV improves the fairness of every of the analyzed classification methods without affecting their behavior and keeping a considerable high level of accuracy.

Note that in this paper, we extend our previous work presented in d'Aloisio et al. (2022) with the following main contributions:

1. We analyze strengths and weaknesses of currently established methods for bias mitigation that can be applied to both binary and multi-class classification with multiple sensitive variables;
2. We revise the original algorithm by generalizing the functions for generating and removing instances during the process. In particular, we rewrite the algorithm as a plug-in approach in which different functions for the creation and deletion of instances can be called;
3. We evaluate the impact of different *instances generating strategies* on the accuracy and fairness achieved by DEMV;
4. We extensively evaluate DEMV considering several settings:
    - employing a larger set of binary and multi-class datasets. The selected datasets are heterogeneous both in terms of data scope and size, enlarging the generality of our analysis;
    - considering a larger set of established baselines able to handle bias in binary and multi-class classification with any number of sensitive variables. We highlight the strengths and weaknesses of the baselines and show how DEMV is able to overcome them in all settings when the multi-class classification task is considered, and in the case of more than two sensitive variables, in case of binary classification task;
    - executing a deeper analysis that employs a different number of sensitive variables. In particular, we consider sensitive groups made by one, two, and three sensitive variables. We highlight how the number of sensitive variables (badly) influences the ability of baselines to improve fairness while DEMV performs consistently;
    - employing a more extensive set of classifiers and analyzing the impact of DEMV on their behavior. In particular, we consider the following categories of classification methods: Linear, Boosting, Support Vector Machines, and Neural Networks.

This paper is structured as follows: in Section 2, we recall some background knowledge used in our work and describe some bias mitigation methods in the context of multi-class classification problem; in Section 3, we describe in detail the proposed approach; Section 4 is dedicated to the experimental analysis that has been conducted to evaluate the best-generating strategy and to compare DEMV both in binary and multi-classification problems while discussing its current strengths and weaknesses, also, a description on how to reproduce the experiments is provided; in Section 5 we discuss the results, and we answer to the four research questions; finally, Section 6 reports possible points of improvements of DEMV and concludes the paper.

2. Background knowledge and related work

In the last ten years, the study of bias and fairness in machine learning acquired considerable relevance. Many definitions and metrics have been proposed to address different kinds of bias and fairness (Mehrabi et al., 2021). In this section, we recall the definitions of bias and fairness used in this paper. Then, we describe the related work in the context of bias mitigation in binary and multi-class classification problems.

2.1. Bias and fairness definitions

Bias (and relative unfairness) can arise from different sources and be defined in several ways. In Mehrabi et al. (2021) the authors highlighted that bias can be generated from:

- the **data** used to train the ML algorithms (e.g., *Measurement bias* (Suresh & Guttag, 2019), *Omitted Variable bias* (Busenbark, Yoon, Gamache, & Withers, 2022; Clarke, 2005), or *Representation bias* (Suresh & Guttag, 2019));

[2] Uniform strategy replicates the instances of the dataset with a uniform probability distribution.

- the **algorithm** which may introduce bias in the users' behavior (e.g., *Algorithmic bias* (Baeza-Yates, 2018));
- the **population** which generates the data used to train the models (e.g., *Historical bias* (Suresh & Guttag, 2019), *Population bias* (Olteanu, Castillo, Diaz, & Kıcıman, 2019), or *Social bias* (Baeza-Yates, 2018)).

The former definitions of bias, with the only exception of *Algorithmic bias*, which is strongly related to the ML algorithm, can be grouped into two macro-categories of bias:

- **Unbalanced Groups bias**: in which the bias is generated by an unequal distribution of instances in the population (e.g., *Representation bias*, *Historical bias*, *Social bias*, *Population bias*)
- **Confounding Variables bias**: in which the bias is generated by a wrong interpretation or representation of instances in the population (e.g., *Measurement bias*, *Omitted Variable bias*)

Our proposed approach addresses the first macro-definition of bias by mitigating the unequal distribution of instances through an optimal balancing of them inside the population.

Concerning fairness definitions, *Demographic (Statistical) Parity (DP)* (Dwork, Hardt, Pitassi, Reingold, & Zemel, 2012; Kusner, Loftus, Russell, & Silva, 2017) is one of the most used definitions of *group fairness* (Mehrabi et al., 2021), which assumes the independence among the predicted positive label y_p and the sensitive variables $S_1, \dots, S_n$. It is defined formally as follows:

Definition 1 (*Demographic Parity*)**.** Let $\hat{Y}$ be the predicted value, y_p the positive label and S a generic binary sensitive variable where $S = 1$ and $S = 0$ identify, respectively, the privileged and unprivileged groups. A predictor is *fair* under *Demographic Parity* if:

$$P(\hat{Y} = y_p | S = 1) = P(\hat{Y} = y_p | S = 0) \tag{1}$$

A different formulation for the DP is the *Disparate Impact (DI)* (Feldman, Friedler, Moeller, Scheidegger, & Venkatasubramanian, 2015), which considers the ratio among the two probabilities. In this case, following the *80% rule* (Feldman et al., 2015), the value must be between 0.8 and 1.2 in order to have *fairness*. DI is defined formally as follows:

Definition 2 (*Disparate Impact*)**.** Let $\hat{Y}$ be the predicted value, y_p the positive label and S a generic binary sensitive variable where $S = 1$ and $S = 0$ identify the privileged and unprivileged groups, respectively. A predictor is *fair* under *Disparate Impact* if:

$$0.8 \leq \frac{P(\hat{Y} = y_p | S = 1)}{P(\hat{Y} = y_p | S = 0)} \leq 1.2 \tag{2}$$

Equalized Odds (EO) (Hardt, Price, & Srebro, 2016) is the third definition of fairness we consider which overcomes the limitation of DP by not removing the correlation among the true and predicted outcomes (Hardt et al., 2016; Verma & Rubin, 2018). In fact, a classifier is considered fair under EO if the probability of an item to be positively classified is the same with respect to the sensitive variable and the ground truth. EO is formally defined as follows:

Definition 3 (*Equalized Odds*)**.** Let $\hat{Y}$ be the predicted value, Y the true value, y_p the positive label and S a generic binary sensitive variable where $S = 1$ and $S = 0$ identify the privileged and unprivileged groups, respectively. A predictor is fair under *Equalized Odds* if:

$$P(\hat{Y} = y_p | Y = y, S = 1) = P(\hat{Y} = y_p | Y = y, S = 0) \quad y \in \{y_1, \dots, y_n\} \tag{3}$$

Both DP and DI fall into the *We Are Equal* metrics family, which holds that all groups have similar abilities concerning the task (i.e., have the same probability of being classified in a certain way). On the contrary, EO resides in the *What You See Is What You Get* family, which states that the observations reflect the ability with respect to the task (i.e., an item should be classified in a certain way only if the other attributes imply it) (Friedler, Scheidegger, & Venkatasubramanian, 2016).

All these definitions were initially proposed for binary classification problems ($y_p = 1$), but they can be easily extended to the multi-class classification domain by identifying one positive label value among the possible ones ($y_p \in \{y_1, \dots, y_n\}$).

2.2. Related works

Over the years, many methods have been proposed to mitigate bias at different levels of data processing (Caton & Haas, 2020; Mehrabi et al., 2021). In particular, we distinguish among (d'Alessandro, O'Neil, & LaGatta, 2017):

- **Pre-processing** methods, which modify the data to remove the underlying bias, such as, (Feldman et al., 2015; Kamiran & Calders, 2012);
- **In-processing** methods, which change the learning algorithm to remove discrimination during the model training process, such as (Agarwal et al., 2018; Denis, Elie, Hebiri, & Hu, 2021);
- **Post-processing** methods, which re-calibrate an already trained model using a holdout set not used during the training phase, such as (Hardt et al., 2016; Putzel & Lee, 2022).

In general, the sooner a technique can be applied, the better because it can be chained with other bias mitigation methods in the later processing phases (*AI fairness 360 - Resources*, 2018; Wolpert, 1999).

Among the different machine learning problems (i.e. classification, regression, clustering, etc.), the classification task has been the most addressed in bias mitigation (Caton & Haas, 2020; Mehrabi et al., 2021). In the following, we will focus on stable methods[3] to improve fairness in the classification task.

Most of the methods available in the literature focus only on binary classification with one sensitive variable (Mehrabi et al., 2021). Among them, one widely adopted *pre-processing* method is the *Sampling* algorithm proposed by Kamiran and Calders (2012). This method balances both privileged and unprivileged users in the case of binary classification with a single sensitive variable. Formally, let be S the sensitive variable with $\{w, b\} \in S$ representing the privileged and unprivileged groups, respectively, and let be Y the target label with $\{+, -\} \in Y$ defining the positive and negative outcomes. The *Sampling* algorithm first splits the original dataset into four groups:

- Deprived group with Positive label (DP): all instances with $S = b \wedge Y = +$;
- Deprived group with Negative label (DN): all instances with $S = b \wedge Y = -$;
- Favored group with Positive label (FP): all instances with $S = w \wedge Y = +$;
- Favored group with Negative label (FN): all instances with $S = w \wedge Y = -$.

Then, for each group, the algorithm computes its *observed* and *expected* sizes. Finally, it balances the groups iteratively by randomly adding and removing instances until the *observed* sizes of the groups are equal to their *expected* ones.

The *Sampling* algorithm is the starting point for the definition of DEMV. In fact, we have extended this algorithm to the multi-class classification domain with multiple sensitive variables and we have employed different instance generation strategies during the balancing process. Very few methods are able to mitigate the bias in the multi-class classification problems (Agarwal et al., 2018; Putzel & Lee, 2022).

Among those, there is the *Blackbox post-processing* method proposed by Putzel and Lee (2022). The authors extend the method proposed by Hardt et al. (2016) to the multi-class setting. Their approach involves the construction of a linear program over the conditional probabilities of the adjusted predictor $P(Y^{adj} = y^{adj} | \hat{Y} = \hat{y}, A = a)$ such that the desired fairness criterion is satisfied by those probabilities. In order to build the linear program, the authors formulate both the loss and fairness criteria as linear constraints in terms of the protected attribute conditional probability matrices. Then, this linear program is used to find the label value, among the possible ones, that minimizes both the loss and the fairness constraints.

An *in-processing* method that solves unfairness in multiple classification settings is the one presented by Agarwal et al. (2018). The algorithm addresses two definitions of fairness at once: *Demographic Parity* and *Equalized Odds*. The authors formulate such definitions as linear constraints and then build an Exponentiated Gradient (EG) reduction algorithm (Kivinen & Warmuth, 1997) that yields a randomized classifier with the lowest error subject to the desired fairness constraints. The method follows a MinMax approach in which the players try to minimize the given constraint and maximize the classifier's score. The authors also propose a simplified Grid Search version of the algorithm (GRID), which generates a sequence of relabeling and reweightings, and trains a predictor for each one. The values yielding the best *accuracy* and *fairness* trade-off are selected and thus returned. Although the authors study their algorithms mainly in binary classification problems, they also show how their method can be applied to regression and multi-classification problems.

To the best of our knowledge, most of the methods in literature are primarily designed for binary classification problems, and few of them can be applied in the *pre-processing* phase. Moreover, only three stable approaches are realized to mitigate bias in multi-class classification problem, and none works in pre-processing phase.

Our proposed method goes beyond the state of art since it works in the pre-processing stage considering multiple sensitives variables, and both binary and multi-class classification task. Since it is a pre-processing method, it can be chained with other algorithms in later processing steps. Finally, it outperforms the state of art in most of the considered settings, as it is shown in the experimental Section 4.

3. Debiaser for Multiple Variables (DEMV)

In this section, we describe in detail the *Debiaser for Multiple Variables (DEMV)* approach, a pre-processing bias mitigation method for multiple sensitive variables in the classification context.

The main idea behind the proposed method is that to enhance effectively the classifier's fairness during pre-processing is necessary to consider all possible combinations of the values of the sensitive variables and the label's values for the definition of the so-called *sensitive groups*. Under the definition of bias considered in this paper (i.e., *unbalanced groups bias*), if a dataset is biased, we observe that the size of the sensitive group identified by the privileged value of the sensitive variable (e.g., *men*) and the positive label (e.g., *high income*) should be larger than expected. In comparison, the size of the sensitive group identified by the unprivileged value of the sensitive variable (e.g., *women*) and the positive label (e.g., *high income*) should be smaller than expected. In the same way, the size of the sensitive group identified by the unprivileged value of the sensitive variable and the negative label should be larger than expected, and the group size determined by the positive value of the sensitive variable and the negative label

[3] Stable methods are the ones having an available and stable implementation.

should be smaller than expected. For this reason, to enhance the fairness of the classifier, we have to perfectly balance the size of these groups by adding or removing items to remove disparity.

We approach the problem by recursively identifying all the possible groups given by combining all the values of the sensible variables with the belonging label (class). Next, for each group, we compute its expected (W_{exp}) and observed (W_{obs}) sizes[4] and look at the ratio among these two values. If $W_{exp} \backslash W_{obs} = 1$, it implies that the group is fully balanced. Otherwise, if the ratio is less than one, the group size is larger than expected, so we must remove an element from the considered group accordingly to a chosen deletion strategy. Finally, if the ratio is greater than one, the group is smaller than expected, so we have to add another item accordingly to a generation strategy. For each group, we recursively repeat this balancing operation until $W_{exp} \backslash W_{obs}$ converge to one. It is worth noting that, in order to keep a high level of accuracy, the new items added to a group should be coherent in their values and distribution with the already existing ones.

Hence, DEMV can be defined as an algorithm made of two separate procedures: identification of the sensitive groups and balancing of them. In the following, we first illustrate the procedure used to identify the sensitive groups; then, we describe the balancing step.

The full implementation of DEMV, comprising of all the code to reproduce the experiments, is available at the Territori Aperti RI[5] and on GitHub[6] as well (see also Section 4.5).

3.1. Sensitive groups identification

Algorithm 1: Pseudo-code of DEMV

Input: (Dataset D, Sensitive variables $S_1, S_2, \dots, S_n$, Label L, $i = 0$, $G = []$, condition=*true*)
Output: Sampled dataset D_S

$n = length(\{S_1, S_2, \dots, S_n\})$
/* base condition: check if all the sensitive variables have been explored for a given condition */
if $i == n$ **then**
  foreach $l \in L$ **do**
    $g = \{X \in D | \text{ condition} \wedge L == l\}$
    $W_{exp} = \frac{|\{X \in D | \text{condition}\}|}{|D|} * \frac{|\{X \in D | L == l\}|}{|D|}$
    $W_{obs} = \frac{|g|}{|D|}$
    $g_b = \text{BALANCE}(g, W_{exp}, W_{obs})$
    add g_b to G
  return G
else
  /* recursion point: select a new sensitive variable and call DEMV for each possible value of the variable */
  $i = i + 1$
  foreach $s \in S_i$ **do**
    $G' = \text{DEMV}(D, S_1, \dots, S_n, i, G, \text{condition} = condition \wedge S_i == s)$
    add G' to G
  /* end condition: check if the number of explored sensitive groups is equal to the number of all possible combinations among values of the sensitive variables and values of the label */
  if $length(G) == |L| * \left(\prod_{i=1}^{n} |S_i|\right)$ **then**
    D_S = merge all $g \in G$
    return D_S
  else
    /* if the end condition is not satisfied, simply return the set of explored sensitive groups */
    return G

[4] The formal definition of these values is given in Section 3.1.
[5] https://bit.ly/3scwtaB
[6] https://github.com/giordanoDaloisio/demv2022

The identification and management of the sensitive groups are performed by the *DEMV* recursive function, whose pseudo-code is shown in listing Algorithm 1.[7]

The main scope of this function is to identify and manage all the possible sensitive groups of a dataset. To this aim, this function takes as input the dataset D, the categorical sensitive variables $S_1, \dots, S_n$, the label L and other parameters useful for the recursion: a counter i initially set to 0 (used to count the number of explored sensitive variables), an array G initially empty (used to collect the balanced, sensitive groups), and a boolean *condition* initially set to *true* (used to define the condition needed to identify the different sensitive groups). Lines from 2 to 9 define the base condition of the function. This condition checks if all the sensitive variables needed to identify a sensitive group have been explored (i.e., the counter i equals the number of sensitive variables). If so, the algorithm iterates the possible values of the label and creates, for each of them, a corresponding sensitive group g is defined as $\{X \in D | S_1 == s_1 \wedge S_2 == s_2 \wedge \dots \wedge S_n == s_n \wedge L == l\}$, where $s_1, \dots, s_n$ are possible values of the sensitive variables and l is a value of the label.

Then, for each group, the algorithm computes expected and observed sizes. These two values are defined respectively as:

$$W_{exp} = \frac{|\{X \in D | S = s\}|}{|D|} * \frac{|\{X \in D | L = l\}|}{|D|} \tag{4}$$

$$W_{obs} = \frac{|\{X \in D | S = s \wedge L = l\}|}{|D|} \tag{5}$$

where $S = s$ is a generic condition on the value of the sensitive variables[8] (i.e., *condition* variable in algorithm 1) and $L = l$ is a condition on the label's value. It is worth noting that $|\{X \in D | S = s \wedge L = l\}|$ is equal to the size of the sensitive group g identified by the conditions $S = s$ and $L = l$.

Next, the algorithm balances the group by invoking the BALANCE function (listing Algorithm 2). This function implements the balancing strategies described in Section 3.2. Finally, the approach adds the balanced group g_b to the array G (used to collect all the balanced groups) and returns it.

Lines from 11 to 14 identify the recursion point of the function. The purpose of the recursion is to build the condition needed to determine the sensitive groups dynamically. In particular, if the algorithm has not explored all the sensitive variables (i.e., the value of i is not equal to the number of sensitive variables), the algorithm starts exploring a new one (variable S_i in the code). The exploration is done by iterating all the possible values of the current sensitive variable S_i. Each value of the sensitive variable corresponds to a new sensitive group that must be identified and balanced. Each identified sensitive group is collected inside a temporary set G'. The returning set of sensitive groups G' partially identified by the given sub-condition is then merged with the given set of sensitive groups G.

Finally, lines from 15 to 19 define the end condition of the function. In particular, the total number of sensitive groups obtainable from a dataset with n sensitive variables and a label L is equal to the product of all the possible values of the sensitive variable and the values of the label, that is:

$$|L| * \left(\prod_{i=1}^{n} |S_i| \right)$$

If the length of G is equal to this value, then the function has considered and balanced all the groups and returns the final sampled dataset D_S. Otherwise, the function, being in the middle of the recursion returns G, will be again merged with the result of the previous recursive calls. This procedure shown in Algorithm 1 can also be applied to binary classification problems; in that case, the number of sensitive groups will be equal to

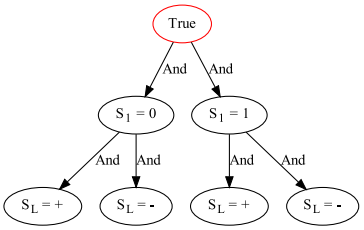
$$2 * \left(\prod_{i=1}^{n} |S_i| \right)$$

We like to note that, even if the number of sensitive groups grows exponentially with respect to the number of sensitive variables, this number is still manageable in the real-world case scenario where a small amount of sensitive variables are typically considered (e.g., solely 32 groups are present in a multi-class classification task where four classes and three binary sensitive variables are considered).

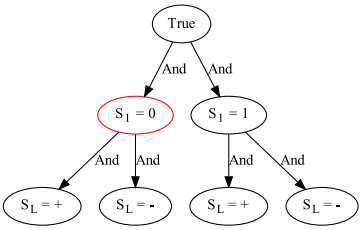
To better clarify the behavior of DEMV, Fig. 2 shows an example execution of the first steps of the algorithm on a dataset with one binary sensitive variable and a binary label. For the sake of simplicity in this example we are using binary variables, but as highlighted in Section 4, DEMV can also be applied to multi-class labels and categorical sensitive variables. Fig. 2(a) represents step 0 of the algorithm, in which the counter is set to 0, and the condition is set to *true*. Next, the algorithm starts a depth-first exploration of the depicted tree. In Fig. 2(b), the algorithm adds the condition $S_1 == 0$ to the initial *true* condition, and in Fig. 2(c) the condition $S_L == +$ is also added. Fig. 2(d) depicts the identification of the first sensitive group defined as $\{X \in D | S_1 == 0 \wedge S_L == +\}$. Finally, Figs. 2(e) and 2(f) show the identification of the second sensitive group, this time defined as $\{X \in D | S_1 == 0 \wedge S_L == -\}$. The algorithm then proceeds to balance the other sensitive groups. When all the groups have been balanced, they are merged to return a fully balanced dataset.

[7] We recall that a recursive function is generally made of three main sections: a *base condition*, which defines the main return statement of the function, a *recursion point* in which the function calls itself, and an *end condition* representing the end of the process.

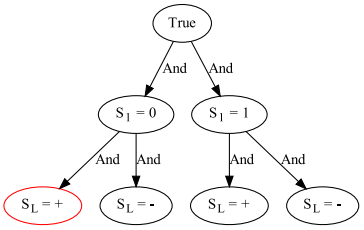
[8] The variables can be binary, discrete or categorical ones



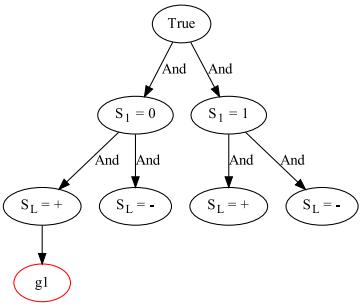
(a) Step 0: $i = 0$ and $condition = true$



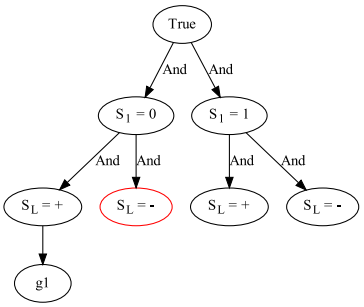
(b) Step 1: $i = 1$ and $condition = (true \wedge S_1 == 0)$



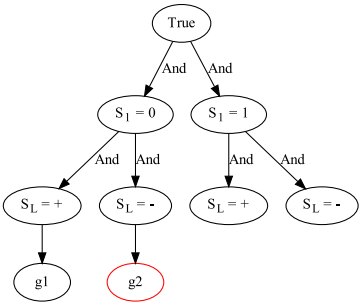
(c) Step 2: $i = 1$ and $condition = (true \wedge S_1 == 0 \wedge S_L == +)$



(d) Step 3: first sensitive group identified



(e) Step 4: $i = 1$ and $condition = (true \wedge S_1 == 0 \wedge S_L == -)$



(f) Step 5: second sensitive group identified

Fig. 2. Example execution of the first steps of DEMV algorithm.

3.2. Balancing strategies

The group-balancing operation is implemented by the BALANCE function, whose pseudo-code is depicted in listing Algorithm 2. This function takes as input the group g and the expected (W_{exp}) and observed size (W_{obs}). The core of this algorithm is a loop that checks if the value of $W_{exp} \backslash W_{obs}$ is different from 1. If the ratio is < 1, then it means that the size of the group is higher than expected. In this case the algorithm selects an index in the range of $(0, size(g) - 1)$ accordingly to the deletion strategy REMOVE to remove the corresponding item from the group. Otherwise, if the ratio is > 1, then the size of the observed group is lower than expected. In this case, the algorithm generates a new sample by using the generative strategy GENERATE, then it adds the new generated sensitive group. Finally, the algorithm returns the balanced group when the while condition becomes true.

To better understand the overall process we need to also discuss the two underlying removal and generative strategies. The simplest of the two strategies is the removal one, where the removal candidate must be selected among the samples already present in the group. The removal strategy implemented in REMOVE function is typically based on a sampling function that follows a given distribution (e.g. uniform).

Conversely, the generative strategy implemented in the GENERATE function might be the most tricky since it is responsible for providing new samples used by the subsequent learning task. For instance, a simple approach might be to duplicate one of the

Algorithm 2: Pseudo-code of BALANCE

```
Input: (Group g, Expected size W_exp, Observed size W_obs)
Output: Balanced group g
while W_exp \ W_obs != 1 do
    /* the group is not balanced */
    if W_exp \ W_obs < 1 then
        /* the size of the group is higher than expected, so we must remove an item from the
           group */
        i = REMOVE(0, ..., size(g) − 1)
        remove item i from g
    else if W_exp \ W_obs > 1 then
        /* the size of the group is lower than expected, so we must add a new item to the group */
        i = GENERATE()
        add item i to g
    recompute W_obs
return g
```

samples already present in the group according to a sampling function that follows a certain distribution (e.g. uniform). It might be also possible to adopt other well-known generative approaches in the literature. In this work, we adopt a uniform sampling for the removal step while we will test and discuss three generative approaches (i.e. Uniform Sampling, SMOTE and ADASYN) in the experimental Section 4.

4. Experimental analysis

This section describes the experiments we have conducted to evaluate DEMV: Section 4.1 reports the used experimental setting comprising the selected metrics and baselines; Section 4.2 describes the employed datasets and their characteristics; Section 4.3 reports on the analysis we have performed to select the best instance generation strategy to be plugged-in DEMV; Section 4.4 shows DEMV's evaluation results both in multi-class and binary classification; and finally, Section 4.5 reports a description on how to reproduce the performed experiments using the available code. We like to remark that the full implementation of DEMV and the code to reproduce all the performed experiments is available at the Territori Aperti RI[9] and on GitHub[10] as well.

4.1. Experimental setting

We evaluate DEMV under heterogeneous conditions by applying a set of binary and multi-class datasets. As a base classifier, we used a Logistic Regression model (Menard, 2002) since it is very efficient from a computational point of view, natively supports multi-class classification, and being a white-box method, it is comprehensible and promotes transparency. In addition, we also performed some specific experiments involving more sophisticated classifiers to analyze the impact of DEMV on these methods. The involved classifiers are: Gradient Boosting (Friedman, 2002), Support Vector Machine (SVM) (Noble, 2006), and Neural Network with *ReLU* activation function (Hagan, Demuth, & Beale, 1997). For all the experiments, we adopt the implementation from the `scikit-learn` library (Pedregosa et al., 2011) with the default hyper-parameters.

For all the experiments, we compute the following metrics on the testing set:

- *Absolute Statistical Parity (SP)*, defined as the absolute value of the original Statistical Parity from (Dwork et al., 2012). We normalized this metric to reduce his variability and better evaluate each method's performance (i.e., avoid situations in which we measure values like 0.2 and −0.2 in two different runs, resulting in a mean of zero with a high standard deviation). The optimal value is **zero**.
- *Disparate Impact (DI)* (Feldman et al., 2015), where the optimal value is **one**. To avoid the occurrence of reverse bias (i.e., metric value firmly higher than one), we adopt the formulation proposed by Radovanović, Petrović, Delibašić, and Suknović (2021):

$$\mathrm{DI} = min\left(\frac{p(\hat{y}=1|s=1)}{p(\hat{y}=1|s=0)}, \frac{p(\hat{y}=1|s=0)}{p(\hat{y}=1|s=1)}\right) \tag{6}$$

  This metric computes the minimum among two formulations of DI wherein one, the unprivileged group ($s = 0$) is at the numerator, and the other is at the denominator. The metric value is between zero and one, where one means complete fairness.
- *Absolute Equalized Odds (EO)*, defined as the absolute value of the original Equalized Odds from (Hardt et al., 2016). We normalized this metric for the same reasons as SP. The optimal value is **zero**.

[9] https://bit.ly/3scwtaB
[10] https://github.com/giordanoDaloisio/demv2022

Table 1
Description of the performed experiments.

Experiment	Reference section	Scope	Task	Involved classifier	Number of sensitive vars	Involved debiaser methods
1	4.3	Comparison of different implementations of DEMV embedding diverse generative strategies both in binary and multi-class classification	Binary and multi-class	Logistic Regression	2	DEMV Uniform DEMV Smote DEMV Adasyn
2	4.4.1	Analyze the behavior exposed by debiaser methods with sensitive groups identified by a different number of sensitive variables	Binary	Logistic Regression	1	No one EG Grid Blackbox DEMV
					2	No one EG Grid DEMV
					3	No one EG Grid DEMV
3	4.4.2	Analyze the behavior exposed by debiaser methods with sensitive groups identified by a different number of sensitive variables	Multi-class	Logistic Regression	1	No one EG Grid Blackbox DEMV
					2	No one EG Grid DEMV
					3	No one EG Grid DEMV
4	4.4.3	Analyze the behavior exposed by debiaser methods involving more sophisticated classifiers both in binary and multi-class classification	Binary and multi-class	Logistic Regression[a] Gradient Boosting Support Vector Machine Neural Network	2	No one EG Grid DEMV

[a]This classifier has not been directly employed in this experiment, but for clearness we report the results obtained in the previous experiments.

- *Zero–one Loss (ZO Loss)* (Domingos & Pazzani, 1997), where the optimal value is **zero**.
- *Accuracy (Acc)* (Rosenfield & Fitzpatrick-Lins, 1986), where the optimal value is **one**.
- *Harmonic Mean (H-Mean)* (Ferger, 1931) of the above metrics. In particular, concerning the metrics whose optimal value is zero (i.e., SP, EO, and ZO Loss), we beforehand perform a value's permutation to have the optimal value equal to one, then we compute the H-Mean using these new values. Formally, H-Mean is computed as follows:

$$\text{H-Mean} = \frac{5}{\frac{1}{(1-|SP|)} + \frac{1}{(1-|EO|)} + \frac{1}{(1-|ZOLoss|)} + \frac{1}{DI} + \frac{1}{Acc}} \tag{7}$$

Table 1 shows the list of performed experiments. Specifically, we performed four main sets of experiments.

The first is the comparison of different implementations of DEMV embedding diverse generative strategies (see Section 4.3). We consider the following three strategies:

- *Random sampling on Uniform Distribution (UNIFORM)*, where the algorithm duplicates an item present in the group with a uniform probability distribution;
- *Synthetic Minority Oversampling Technique (SMOTE)* from (Chawla et al., 2002);
- *Adaptive Synthetic Sampling Approach (ADASYN)* from (He et al., 2008).

This analysis has been performed in binary and multi-class classification tasks on all the considered datasets considering two sensitive variables and using the Logistic Regression as a classifier.

After identifying and settling on the best generation strategy, we compare DEMV with the selected baselines by performing three main sets of experiments (please refer to Section 4.4). Experiments two and three in Table 1 are focused on analyzing the behavior exposed by debiaser methods with sensitive groups identified by one, two and three sensitive variables. Experiment two focuses on binary classification task (see Section 4.4.1), while experiment three focuses on multi-class classification task (see Section 4.4.2). In both these experiments we employed a Logistic Regression model as a classifier. To have a more concrete representation of the

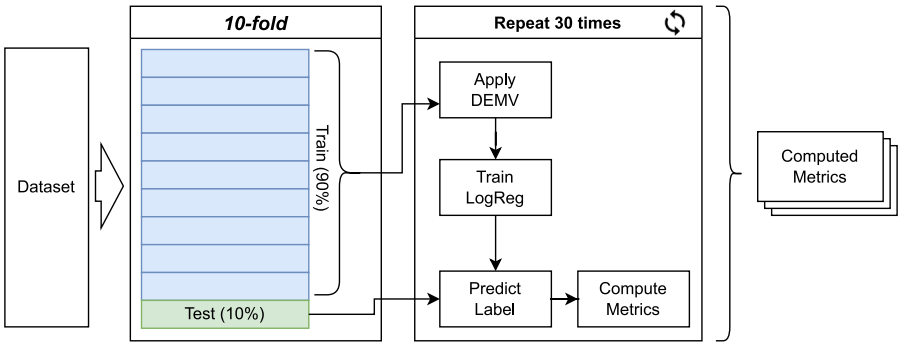


Fig. 3. Evaluation procedure of DEMV for each train–test fold.

behavior of DEMV and the other baselines, at the end of Section 4.4.2 we also report a comparison of normalized confusion matrices for the privileged and unprivileged groups of a particular dataset.

Finally, experiment four in Table 1, is devoted to analyze the behavior exposed by debiaser methods involving more sophisticated classifiers: Gradient Boosting (Friedman, 2002), Support Vector Machine (SVM) (Noble, 2006), and Neural Network with *ReLU* activation function (Hagan et al., 1997). Since these models are more complex from a computational point of view, this last experiment has been performed in binary and multi-class classification tasks on a reduced but heterogeneous dataset considering the two established sensitive variables (see Section 4.4.3).

In all the experiments (with the exception of experiment one) we compare with the following baselines:

- a biased classifier, where no debiasing method is applied, identified in the following by ***No one***;
- the *Exponentiated Gradient* (***EG***) and *Grid Search* (***Grid***) in-processing methods from (Agarwal et al., 2018)[11];
- the ***Blackbox*** post-processing method from (Putzel & Lee, 2022).[12] This method has been employed only in the analyses with one sensitive variable since, by the time of this paper, it does not support multiple sensitive variables.

Concerning Exponentiated Gradient and Grid Search, in agreement with the documentation available online (Fairlearn, 2022), we used the Absolute Statistical Parity and the Zero–one Loss as constraints for binary and multi-class problems, respectively. Instead, Blackbox does not require a specific configuration of the hyperparameters.

For all the experiments showed in Table 1 we follow a *10-fold* cross-validation (Refaeilzadeh, Tang, & Liu, 2016), repeated 30 times for those methods that expose a stochastic behavior (namely DEMV) as depicted in Fig. 3. In particular, to better reproduce a production scenario, we apply DEMV only on the training set and train the Logistic Regression classifier using the balanced dataset. Then, we predict the labels using the original biased testing set and compute the metrics described above. In addition, since the balancing of the groups has a stochastic behavior, for each train–test fold, we repeat the aforementioned process 30 times so that we can investigate how the removal or duplication of different items can influence the *accuracy* and the *fairness* of the classifier.

In all the performed experiments, we report the mean and standard deviation of all the metrics calculated over all the involved datasets. In the representation of such metrics we use bar plots where larger bars depict the mean of the metrics and thin bars show their standard deviation. In representing plots, we distinguish between metrics whose optimal value is 0 (showed on the left side of the figures) and metrics whose optimal value is 1 (reported on the right side of the figures).

4.2. Employed datasets

The experiments are conducted by employing nine well-known datasets (3 for the binary classification and 6 for the multi-class task) from the Bias and Fairness literature. For each dataset, we consider sensitive groups identified by three sensitive variables: two variables are the ones established as sensitive variables by the literature, while the third one is selected, for each dataset, among the variables that could create discrimination, like age, education and so on. Note that PARK dataset has been excluded by the analysis with three sensitive variables because it does not have a third variable suitable for discrimination.

Table 2 depicts the descriptive statistics for the employed datasets. Concerning the sensitive variables, we highlight in bold the two ones established as sensitive by the literature. In the following, it is provided a brief description of the 9 considered datasets.

1. **Adult Income (ADULT)** (Kohavi et al., 1996): This binary dataset comprises 30,940 items by 102 features (one-hot encoded). The goal is to predict if a person has an income higher than 50k a year. This information is represented by the `income` variable. The protected attributes are `sex`, and `race` and the unprivileged group is *black women* (items with `sex` and `race` equal to zero). In the analysis with three sensitive variables, we also introduced the `bachelor` variable, indicating if a

[11] The adopted implementation of Exponentiated Gradient and Grid Search methods are available on the Fairlearn library (Bird et al., 2020)

[12] The considered Blackbox implementation is available at the following link: https://github.com/scotthlee/fairness

person has a bachelor's degree or not. In this case, the sensitive group is *black women with no bachelor's degree*. The positive label is *high income*.

2. **ProPublica Recidivism (COMPAS)** (Angwin, Larson, Mattu, & Kirchner, 2016): This binary dataset is made of 6,167 samples by 399 attributes. The sensitive variables are `sex` and `race`. The goal is to predict if a person will recidivate in the next two years. The favorable label, in this case, is *no*, and the unprivileged group is *Non-Caucasian men* (items with `sex` and `race` equal to zero). In the test with three sensitive variables, we also introduced the `age` attribute. In this case, the sensitive group is `Non-Caucasian men with less than 50 years`.
3. **German Credit (GERMAN)** (Ratanamahatana & Gunopulos, 2002): This binary dataset classifies people described by a set of attributes as good or bad credit risks (`credit` variable). The dataset consists of 1,000 instances by 59 features (one-hot encoded). The sensitive variables are `sex`, and `age` and the unprivileged group is *women with less than 25 years*. The positive label is *low credit risk*. In the experiment with three sensitive variables, we also introduced the `investment_as_income` variable, meaning if a person has more than the 30% of his income invested. In this case, the sensitive group is *women with less than 25 years and with less than 30% of their income invested*.
4. **Contraceptive Method Choice (CMC)** (Lim, Loh, & Shih, 2000): This multi-class dataset comprises 1,473 instances and ten columns about women's contraceptive method choice (*not-use*, *short-use*, and *long-use*). The sensitive variables are `religion` and `work`. The unprivileged group is *Islamic women who do not work* (both values equal one), and the positive label is *long-term use*. In the analysis with three sensitive variables, we introduced the education (`edu`) variable. The sensitive group, in this case, is *Islamic women who do not work and with no education*.
5. **Communities and Crime (CRIME)** (Redmond & Baveja, 2002): This multi-class dataset is made of 1,994 instances by 100 attributes and contains information about the per-capita violent crimes in a community (variable `ViolentCrimesPerPop`). Since the label is continuous, we transformed it by grouping the values in 6 classes using equidistant quantiles. Following (Calders, Karim, Kamiran, Ali, & Zhang, 2013) the sensitive attribute is the percentage of the black population, but we also considered the ratio of the Hispanic population to have two sensitive variables. The unprivileged group is communities with a *high percentage of both black and Hispanic people* (both variables equal to 1), and the positive label is `100` (class of *low rate of crimes*). In the experiment with three sensitive variables, we also introduced the `MedRent` variable, showing the average price of rents in a community. In this case, the unprivileged group is *communities with a high percentage of black and Hispanic people and a low cost of the rent*.
6. **Drug Usage (DRUG)** (Fehrman, Muhammad, Mirkes, Egan, & Gorban, 2017): This multi-class dataset has 1,885 instances and 15 attributes about the frequency of drugs consumption (variable `y`). The classes are *never used*, *not used last year*, and *used last year*. The sensitive variables are `race` and `gender` and the unprivileged group are *white women* (`race` equal to one and `gender` equal to zero). The positive label is *never used*. In the test with three sensitive variables, we also used the `age` variable. In this case, the sensitive group is *white women less than 50 years*.
7. **Law School Admission (LAW)** (Austin et al., 2016): This multi-class dataset comprises 20,694 samples by 14 attributes and contains information about the bar passage data of Law School students. We grouped the continuous label (`GPA`) in 3 groups using equidistant quantiles. The sensitive variables are `race` and `gender` and the unprivileged group are *black women* (both variables equal to one), and the positive label is 2 (class of *high scores*). In the analyses with three sensitive variables, we also introduced the `age` variable. In the experiments with three sensitive variables, we also used the `age` variable. In this case, the unprivileged group is *black women with less than 61 years*.
8. **Parkinson's Telemonitoring (PARK)** (Tsanas, Little, McSharry, & Ramig, 2009): This multi-class dataset comprises 5875 items and 19 features about Unified Parkinson's Disease Rating Scale (UPDRS) score classification (variable `score_cut`). The classes are *Mild*, *Moderate* and *Severe*. The sensitive variables are `sex` and `age` and the unprivileged group are *males with more than 65 years* (`age` equal to one and `sex` equal to zero). Since this dataset does not have a third variable suited for identifying sensitive groups, we used it only in the experiments with one and two sensitive variables.
9. **Wine Quality (WINE)** (Cortez, Cerdeira, Almeida, Matos, & Reis, 2009): This multi-class dataset comprises 6,438 instances and 13 attributes about wine quality (variable `quality`). The classes are four increasing values indicating quality (the higher, the better). The sensitive attributes are the wine's color (`type` variable) and the alcohol percentage lower or higher than 10 (`alcohol` variable). The unprivileged group is *white wine with an alcohol percentage* ≤ 10, and the positive label is 6 (*high quality*). In the experiment with three sensitive variables, we also introduced the `density` variable. In this case, the unprivileged group is *white wine with an alcohol percentage* ≤ 10 *and with a density less than 1.1%*.

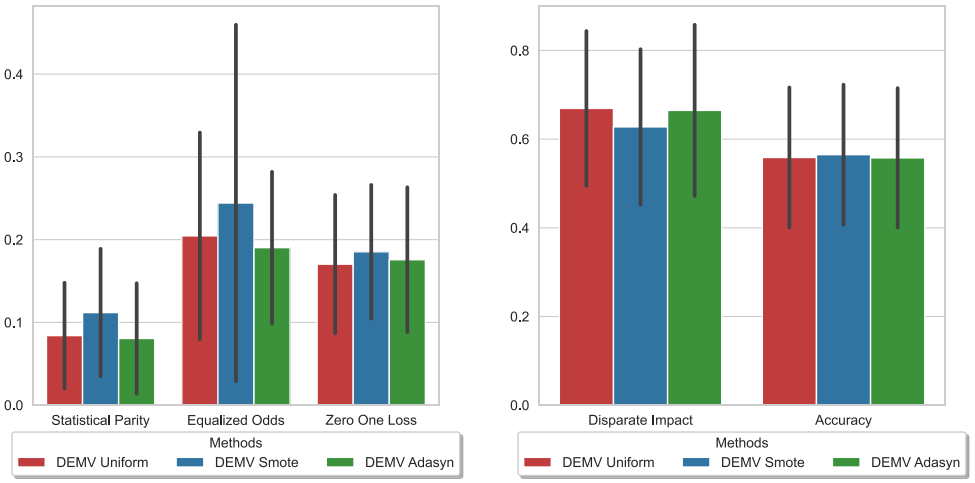
4.3. Selection of the best generative strategy

In this section, we show the experiments made to select the best instance generation strategy to plug-in in DEMV. As described in Section 4.1, we consider the following generation strategies: Uniform sampling, SMOTE, and ADASYN. We perform the comparison with both binary and multi-class datasets using, for each dataset, sensitive groups identified by the two sensitive variables specified in literature. For the considered metrics (i.e., the ones introduced in Section 4.1), we report in Appendix A the tables showing the detailed values calculated. While in this section we show their mean and standard deviation calculated over all the datasets.

The aggregated metrics for multi-class datasets are shown in Fig. 4. From this first analysis, *Uniform* sampling and *ADASYN* give similar results in fairness and accuracy, while *SMOTE* behaves worse.

Table 2
Descriptive statistics for the employed Datasets (boldface are highlighted the protected variables established in the original dataset).

	Adult	Compas	German	CMC	Crime	Drug	Law	Park	Wine
Scope	Social	Justice	Social	Social	Justice	Social	Education	Health	Food
Instances	30,940	6,167	1,000	1473	1,994	1,885	20,427	5,875	6,438
Features	102	399	59	10	100	15	14	19	13
Classes	2	2	2	3	6	3	3	3	4
Positive label	*high income*	*no*	*low-credit risk*	*long-term use*	*100 (low percentage class)*	*never used*	*2 (high scores class)*	*mild*	*high quality*
Sensitive variables	**sex** **race** bachelors	**sex** **race** age	**sex** **age** investment	**religion** **work** edu	**black** **hisp** Medium Rent	**race** **gender** age	**gender** **race** age	**age** **sex**	**type** **alcohol** density
Percentage of sensitive group with two sensitive vars	5.02%	54.71%	10.50%	64.83%	23.62%	45.78%	8.42%	39.45%	11.40%



(a) Metrics whose best value is zero (b) Metrics whose best value is one

Fig. 4. Comparison of generation strategies of DEMV for multi-class classification with two sensitive variables.

Fig. 5 confirms that, also in the case of binary classification, the *Uniform* sampling and *ADASYN* have comparable performances concerning accuracy and fairness.

Since both the *Uniform* sampling and *ADASYN* generative strategies expose similar performance in terms of classifier's fairness and accuracy, we decide to analyze their computational performances in order to select the best and efficient strategy to embed in DEMV. In particular, we focused on their execution time (expressed in seconds) that we report in Fig. 6.

This experiment has been conducted on a *MacBook Air M1 2020* with 16 GB of RAM.

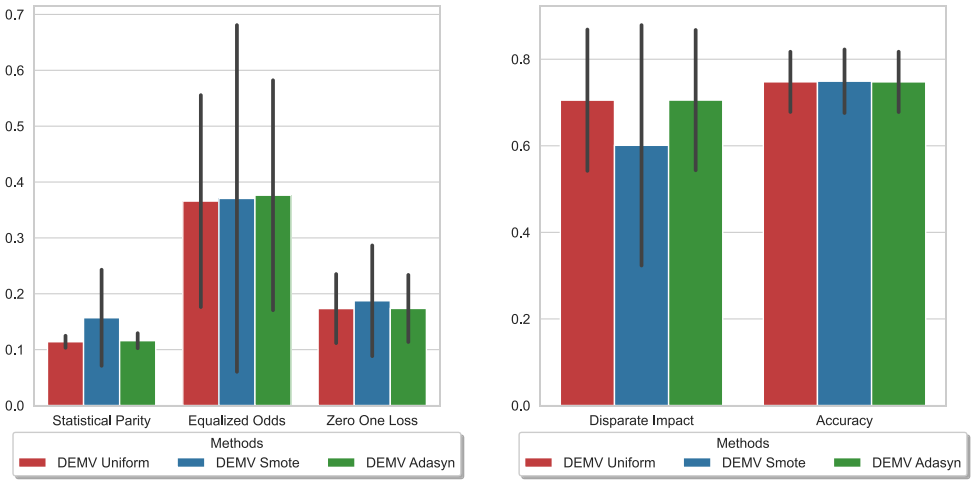
The results show that DEMV implementing *ADASYN* takes much more time for completion, especially in larger datasets, while DEMV with *Uniform* always takes a reasonable execution time.

Considering all the analysis made, we adopt the *Uniform* sampling as the generation strategy to compare against the baselines because the obtained metrics are comparable to ADASYN and its execution time is lower.

4.4. DEMV evaluation in classification tasks

This section presents the quantitative results of the DEMV's evaluation. We compare the performance of DEMV with the selected baselines shown in Section 4.1.

Even if DEMV is a debiaser for the multi-class classification problem, we decided to evaluate it also in binary classification problems to identify its potentialities in this scenario (see Section 4.4.1). However, since the binary classification task is not the primary scope of our work, we decide for readability to show, for each method, the variation of H-Means at the increasing of sensitive variables. For interested readers, detailed results are reported in Appendix B.



(a) Metrics whose optimal value is zero (b) Metrics whose optimal value is one

Fig. 5. Comparison of generation strategies of DEMV for binary classification.

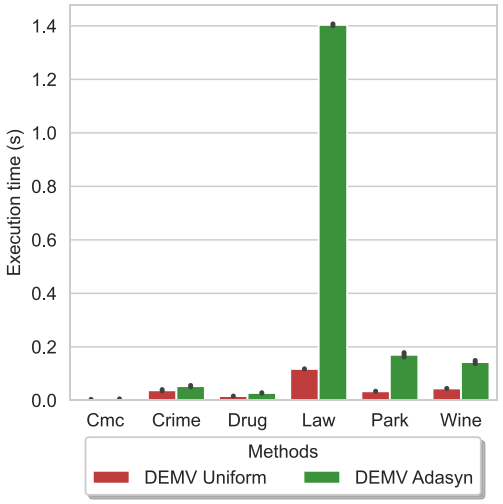


Fig. 6. Execution time in seconds of DEMV Uniform and DEMV Adasyn in multi-class classifications tasks.

In Section 4.4.2, we present the DEMV's evaluation with multi-class classification tasks. In this case, we report the mean and standard deviation of each measure described in Section 4 using the bar plots. Then, as an overall view, we show the variation of each method's H-Mean at increasing sensitive variables. Detailed metrics for each dataset are provided through tables in Appendix C.

Finally, in both binary and multi-class classification scenarios:

- for each dataset, the variation of H-Means, at the increasing number of sensitive variables is reported using line plots in which each line identifies one method. We recall that since the *Blackbox* algorithm does not support multiple sensitive variables, it has been applied only in the experiments involving sensitive groups identified by one sensitive variable, so it is represented as a point in such plots;
- as already said, each dataset has two sensitive variables. To run the experiment with one sensitive variable, we averaged the results of two independent experiments, one for each sensitive variable;
- it is reported the statistical significance of all experiments computed using the ANOVA test in each analysis (McDonald, 2009). This test checks for the null hypothesis that all groups have the same mean; if the probability value (*p-value*) is less than 0.05, the test rejects the null hypothesis, which means that the groups have a different mean value. The ANOVA tables showing the test results are shown in Appendix D as well.

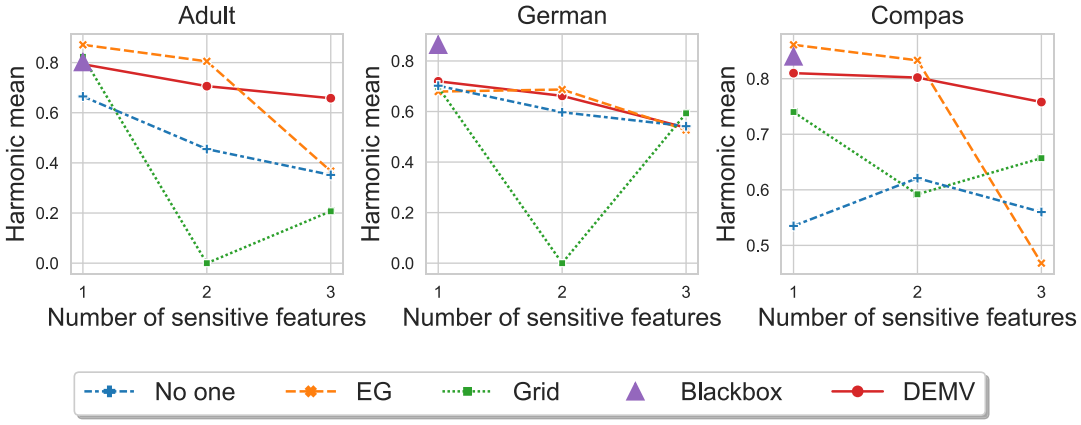


Fig. 7. Comparison of overall H-Mean at different number of sensitive variables for binary classification datasets.

Table 3
Overall H-Mean of all methods with different sensitive variables in the binary classification context.

Sensitive variables	Methods				
	No one	Blackbox	EG	Grid	DEMV
1	0.648 ± 0.034	**0.835 ± 0.031**	**0.835 ± 0.048**	0.761 ± 0.056	0.777 ± 0.036
2	0.558 ± 0.09	–	**0.775 ± 0.077**	0.197 ± 0.342	0.723 ± 0.072
3	0.485 ± 0.115	–	0.454 ± 0.081	0.486 ± 0.243	**0.651 ± 0.111**

4.4.1. Comparison in the binary classification task

In this section, we compare DEMV with the other baselines in a binary classification context. It is worth noting that, in the context of binary classification with one sensitive variable, DEMV coincides with the original *Sampling* method (Kamiran & Calders, 2012) it derives from.

The results of the experiments are reported in Fig. 7. As reported in the figure, DEMV (represented with the red line) better mitigates the bias with an arbitrary number of sensitive variables, producing results that are generally competitive and even better when more than two variables are considered. A closer analysis lets us notice that EG outperforms the other methods when the number of sensitive variables is one or two. At the same time, it dramatically fails when three sensitive variables are considered. Blackbox method (reported by a single triangle in correspondence to one variable) is a good performer only when one sensitive variable is needed. In contrast, its adoption will not be applicable in cases where more sensitive variables must be considered.

The conducted ANOVA test, whose detailed results are reported in Appendix D, confirms the statistical significance of all the experiments made in case of the binary classification task.

To give an overall view of the performances of each method, we provide a synthetic version of the above results in Table 3 where, for each method, we report the average of the H-Mean computed overall for the considered datasets. This summary confirms what we observed in the details above; that is, in binary classification with one sensitive variable, Blackbox and EG perform similarly. EG also behaves well in case of two variables. Finally, DEMV produces competitive results with one or two sensitive variables while outperforming the other baselines when three variables are needed. However, no clear winner can be picked out of the shelf, and more evaluation should be provided to determine which method to apply in different settings, including dataset characteristics. In addition, a quality that can be decisive in selecting the best method is the computational complexity, which we will consider in the future for a better evaluation of DEMV.

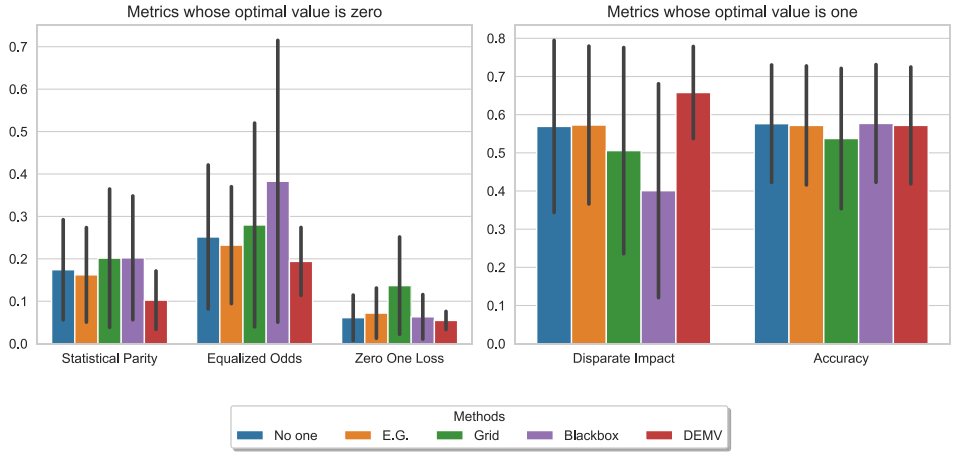
4.4.2. Comparison in the multi-class classification task

In this subsection, we report the results of the experiments conducted in the context of multi-class classification.

The experiment's results are reported in Figs. 8 and 9. The former reports the mean and the standard deviation of the metrics computed by each method on overall datasets, distinguishing among the usage of one (a), two (b), and three (c) sensitive variables. The latter instead, provides a different view, and for each dataset, it reports the values of H-Mean for each method at the increasing of sensitive variables.

In particular, Fig. 8(a) focuses on the experiments involving one sensitive variable. The high standard deviation of all metrics is explained by the fact that the metrics are here calculated putting together the results of two separate experiments. From the figure we can see that, in average, DEMV overcomes all the baselines. In addition, we observe that DEMV is the method performing in a more stable and coherent way. This is highlighted by an overall lower standard deviation for all metrics.

The performances of DEMV in case of one variable are confirmed by Fig. 9, where it can be seen that DEMV overcomes the baselines in all dataset with the only exception of CMC (in which the best method is Grid), and Wine (in which the best method is Blackbox).



(a) Application with one sensitive variable

(b) Application with two sensitive variables

(c) Application with three sensitive variables

Fig. 8. Comparison of DEMV with the baselines in multi-class classification.

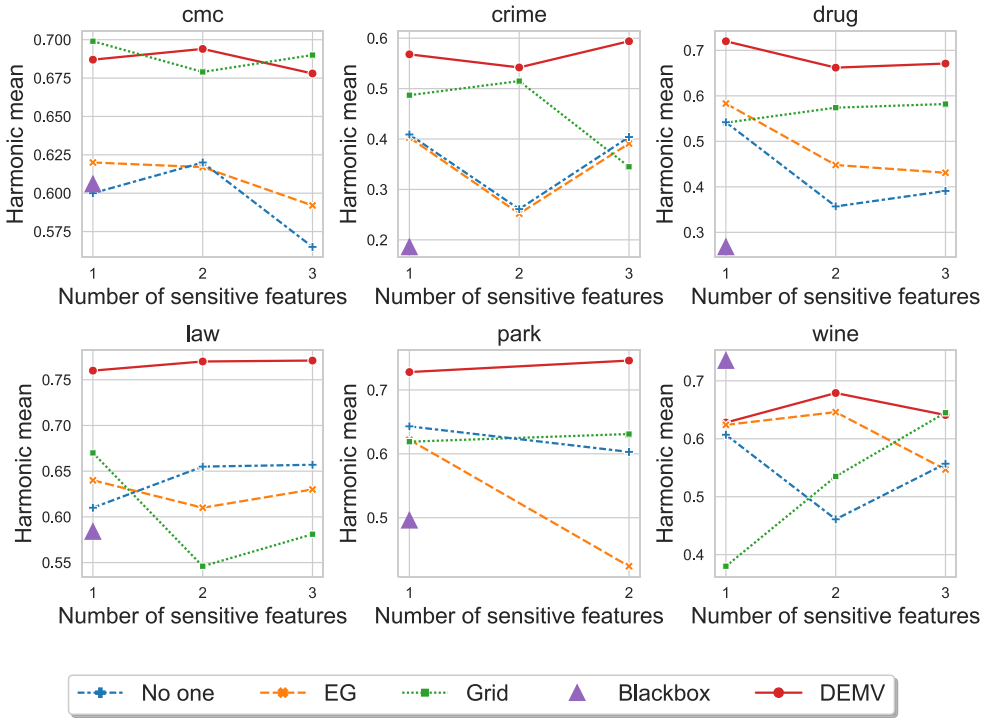


Fig. 9. Comparison of overall H-Mean at different number of sensitive variables for multi-class classification datasets.

Table 4
Overall H-Mean of all methods with different sensitive variables in the multi-class classification context.

Sensitive variables	Methods				
	No one	Blackbox	EG	Grid	DEMV
1	0.568 ± 0.085	0.479 ± 0.211	0.582 ± 0.09	0.566 ± 0.121	**0.682 ± 0.072**
2	0.493 ± 0.16	–	0.505 ± 0.16	0.58 ± 0.063	**0.677 ± 0.081**
3	0.486 ± 0.135	–	0.49 ± 0.128	0.529 ± 0.182	**0.646 ± 0.08**

More detailed results are reported in Table 15 in Appendix C.

Finally, the ANOVA test (whose detailed results are reported in Table 22.a of Appendix D) confirms the statistical significance of the experiments with the exception of the metrics EO, which has a p-value of 0.262. The fact that the observations of EO are not statistically significant can be explained by the high standard deviation of such metric in Grid and especially in Blackbox.

Fig. 8(b) reports the results of the experiments with sensitive groups identified by two sensitive variables. As before, DEMV overcomes all the other baselines in all the involved datasets, and its stability is confirmed by an overall lower standard deviation. Fig. 9 shows that, also in this context, DEMV outperforms the baselines in all datasets. Detailed results in the case of two sensitive variables are reported in Table 16 of Appendix C.

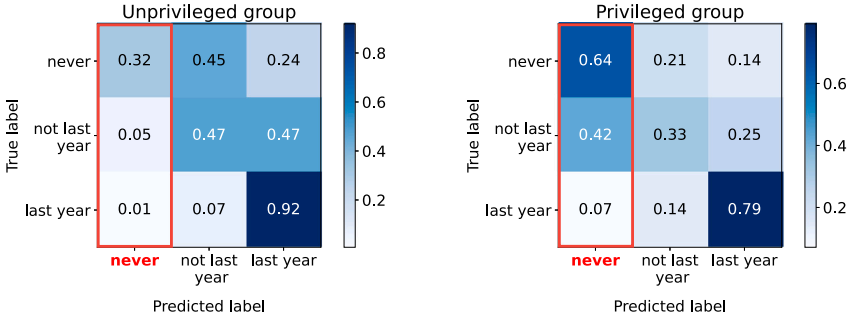
The ANOVA test confirms the statistical relevance of all the results (see Table 22.b in Table 22).

The above considerations are also confirmed in the case of three involved sensitive variables. The results are reported in Fig. 8(c) and in Table 17 of Appendix C. We recall that the Park dataset has not been used in this experiment since it does not have a third variable suitable to be treated as sensitive.

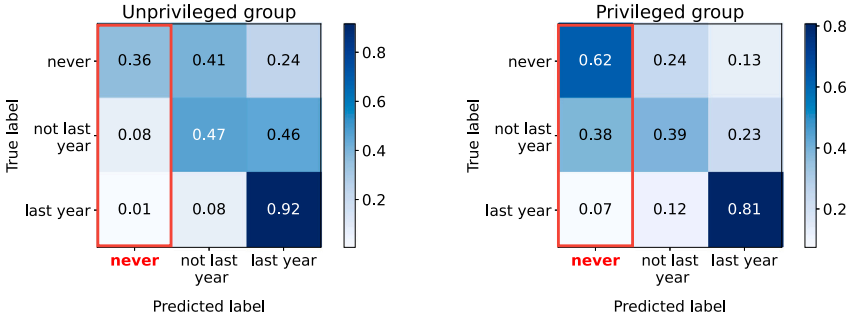
In this case, from Fig. 9 it can be seen that Grid performs slightly better in CMC (with a delta of H-Mean of about 0.01 points) and Wine (with a delta of 0.005). As for the other two experiments, DEMV performs more consistently with an overall standard deviation lower than the different baselines. Again, the ANOVA test confirms the statistical significance of the experiments (see Table 22.c in Table 22), with the only exception of EO metrics (with a p-value of 0.27) which has a high variability especially with EG and with the biased classifier (indentified by *No one* label) in the figure.

As for the experiments involving binary classification datasets, in order to have a complete concise overview, in Table 4 we report the overall H-Mean of all the methods in the three performed experiments overall the datasets. Note that, DEMV generally overcomes the other baselines in all the explored contexts increasing the H-Mean by up to 0.2 points with respect to the biased classifier (i.e., *No one* in the table) in the experiments with two and three sensitive variables.

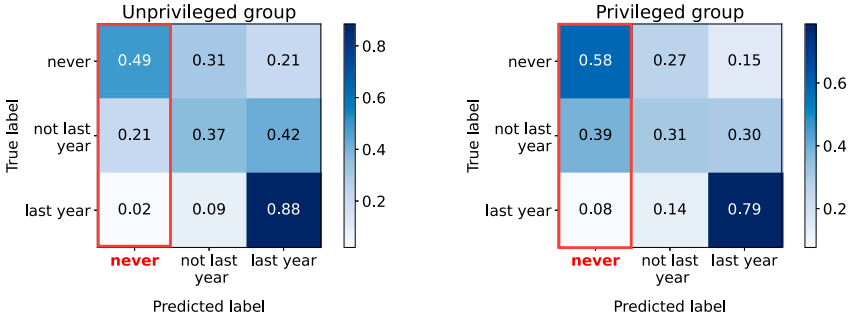
Finally, to have a more concrete representation of the behavior of all the analyzed methods, in Fig. 10 we report a comparison of the normalized confusion matrices (Krstinić, Braović, Šerić, & Božić-Štulić, 2020) for the privileged and unprivileged (i.e., biased)



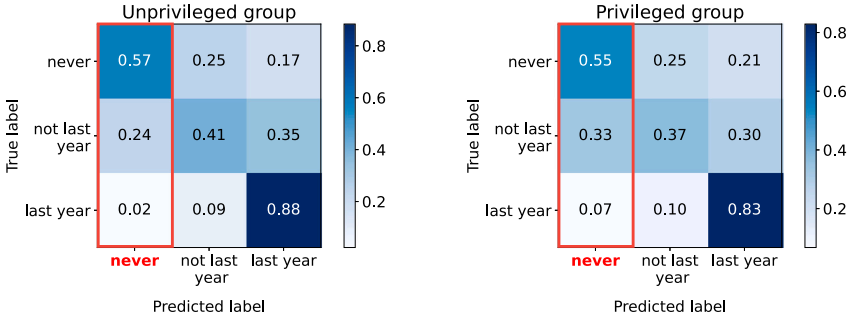
(a) Biased classifier



(b) EG



(c) Grid



(d) DEMV

Fig. 10. Normalized confusion matrices of privileged and unprivileged groups for each baseline on Drug dataset.

Table 5
Overall H-Mean of all methods with different classifiers in the binary classification context.

Classifier	Methods			
	No one	EG	Grid	DEMV
Logistic Regression	0.558 ± 0.09	**0.775 ± 0.077**	0.197 ± 0.342	0.723 ± 0.072
Gradient Boosting	0.588 ± 0.19	0.476 ± 0.383	0.582 ± 0.228	**0.724 ± 0.057**
SVM	0.57 ± 0.201	0.554 ± 0.238	0.59 ± 0.205	**0.721 ± 0.066**
Neural Network	0.584 ± 0.202	–	–	**0.69 ± 0.127**

groups of the Drug dataset with two sensitive variables. We decided to choose the Drug dataset for this experiment since it is among the ones having a high bias and showing better the inequality among the privileged and unprivileged groups (confirmed also by the values of the fairness metrics for the biased classifier showed in Table 16 of Appendix C). In all the matrices, we highlight in red and in boldface the predicted positive label (i.e., *never*), which identifies the column of the matrix affected by bias (highlighted in red as well). In particular, Fig. 10(a) shows the confusion matrices of the biased classifier. From the picture, it can be seen how the probability of the privileged group having a predicted positive label (i.e., column corresponding to *never*) is much higher than the unprivileged group. The confusion matrices related to EG and Grid (Figs. 10(b) and 10(c) respectively) do not differ much from the ones of the biased classifier, meaning that these two methods are not able to improve the fairness of the classifier. Instead, in Fig. 10(d), it can be seen how DEMV is able to balance these two matrices, and the probability of having the positive label predicted is almost the same for the two groups, meaning that the fairness of the classifier has increased.

4.4.3. Comparison using more sophisticated classifiers

In this subsection, we report the results of the experiments conducted in binary and multi-class classification context using more complex classifiers. As already described in Section 4.1, the employed classifiers are: Gradient Boosting, Support Vector Machine (SVM), and Neural Network with *ReLU* activation function. Since these models are more complex from a computational point of view, we performed these experiments on a reduced, but heterogeneous set of data using sensitive groups identified by two sensitive variables. The selected datasets are: Adult (binary large dataset), COMPAS (binary small dataset), CMC (multi-class small dataset), and Law (multi-class large dataset).

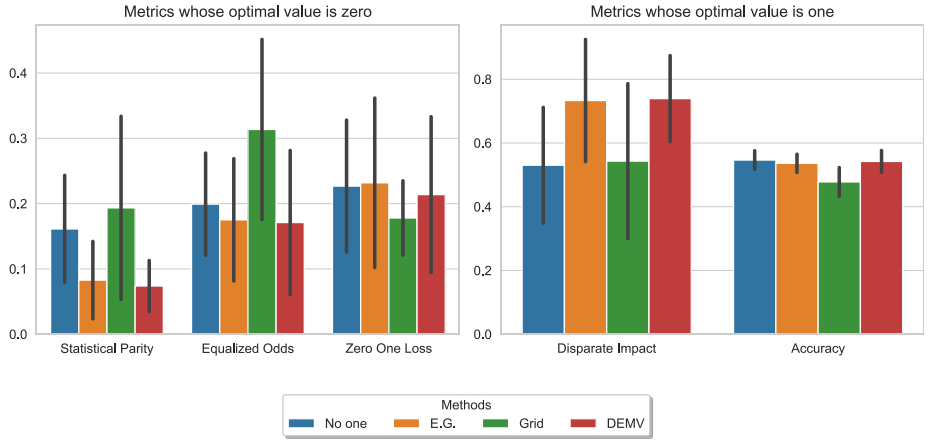
Finally, considering the debaiser approaches, it is worth noting that EG and Grid cannot be applied when a Neural Network model is used as classifier. In fact, EG and Grid apply arbitrary weights to the instances in order to remove bias, but Neural Networks by their nature do not allow the specification of weights to the instances. For this reason, in the experiments involving Neural Networks, we only compared the performance of the original classifier with the performance of the classifiers after the application of DEMV.

Concerning binary classification, Table 5 reports the overall H-Mean of all the baselines for each involved classifier overall the involved datasets,[13] detailed results are reported in Appendix B as well. Note how, differently from the experiments with a Logistic Regression classifier, DEMV overcomes the other baselines in all of the performed analyses, with a delta up to around 0.2 points in case of EG with a Gradient Boosting classifier. The Anova test, whose results are reported in Appendix D, confirms the statistical significance of the results.

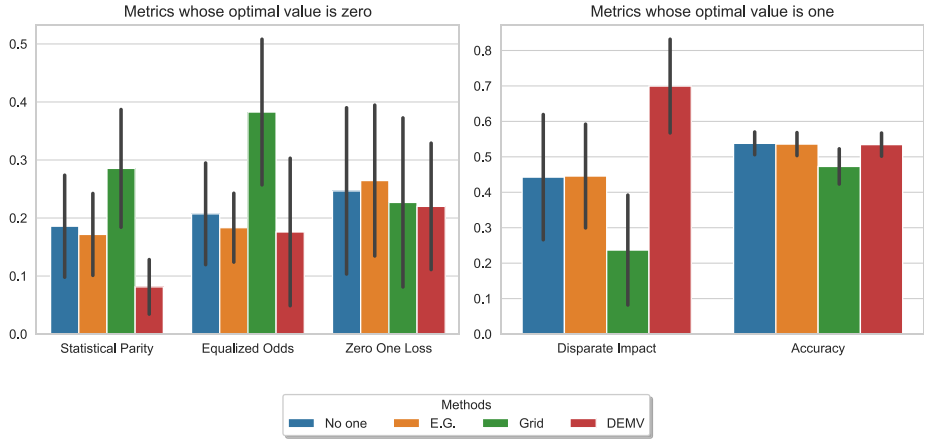
Concerning multi-class classification, Fig. 11 reports the mean and the standard deviation of all the metrics computed overall datasets. In particular, Fig. 11(a) shows the results of the experiments involving the Gradient Boosting classifier. In this context, DEMV outperforms the baselines under the SP and EO definitions of fairness, while it almost equals EG under the DI definition of fairness. More detailed results are reported in the Table 18 of Appendix C. The ANOVA test confirms the statistical significance of this experiment, with the only exception of Zero One Loss which has a p-value of 0.732 (see Table 24.a of Appendix D). Fig. 11(b), reports instead the results of the experiments involving Support Vector Machines. In this context, it can be seen how DEMV overcomes the baselines under all the considered definitions of fairness, keeping an accuracy level almost equal to the original biased classifier. Detailed results are reported in Table 19 of Appendix C as well. Also in this case, the ANOVA test confirms the statistical significance of the results (see Table 24.b of Appendix D). Finally, Fig. 11(c) details the results of the experiments performed with Neural Networks. We recall that in this case EG and Grid cannot be applied, hence we compared DEMV only with the biased classifier. Also in this case DEMV is able to improve the fairness of the classifiers keeping an almost unchanged level of accuracy and more detailed results are reported in Table 20 of Appendix C. The ANOVA test confirms the statistical significance of the results as well (see Table 24.c of Appendix D).

Table 6 reports the overall H-Means of all the methods for each classifier overall the selected datasets. It can be seen how DEMV generally overcomes the other baselines with all the selected classifiers with a delta up to around 0.3 points concerning SVM with Grid method.

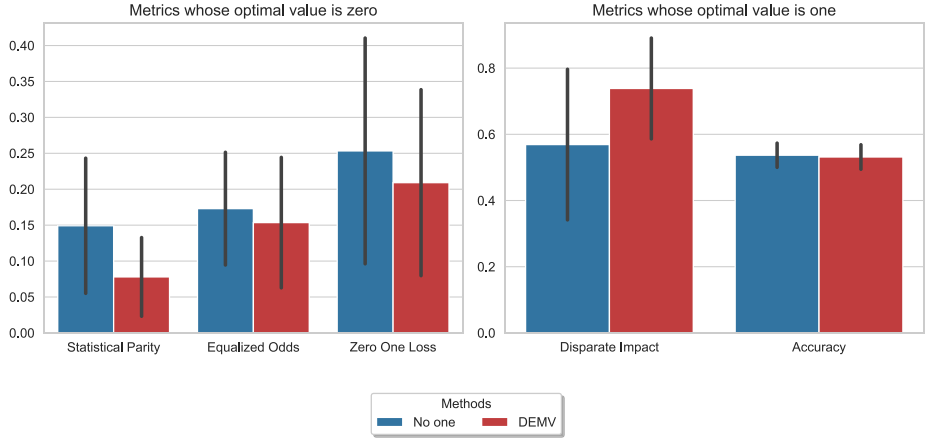
[13] To have a better overall view, we also reported the measurements computed with a Logistic Regression classifiers, which corresponds to the measures of Table 3 with two sensitive variables.



(a) Application with Gradient Boosting



(b) Application with Support Vector Machines



(c) Application with Neural Networks

Fig. 11. Comparison of DEMV with the baselines in multi-class classification using other classifiers.

4.5. Reproducibility of the experiments

Nowadays, ensuring that the proposed methods and their corresponding results are sound and reliable is one of the challenges for research in machine learning. To ensure that the findings are valid, it is essential for the experiments to be repeatable and to

yield results and conclusions comparable or identical to the originally reported ones (Pineau et al., 2021). For this reason, we choose to release the full code of the DEMV algorithm along with the replication package of all the performed experiments. This section is dedicated to describing how to use such code in order to reproduce the experiments described in the previous sections. We recall again that the full implementation is available in the Territori Aperti RI and on GitHub as well. The repository also includes a specification of all the python dependencies required for a correct execution of the code, which can be installed using `anaconda`.[14] or `pip`[15]

The program that must be called to replicate the experiments is `generatemetrics.py`, which is responsible to generate the measures computed and aggregated in all the experiments of Section 4. Noticing that the code to reproduce the plots presented in this paper has not been included in the replication package, but can be easily implemented using the metrics generated from the given program. The code `generatemetrics.py` can be invoked from the command line through the python interpreter in the following way:

```
$ python generatemetrics.py <DATASET> <METHOD> <NUMBER_OF_FEATURES> --sensitivefeature <
    SENSITIVE FEATURE?> --classifier <CLASSIFIER?> --cm <CM?>
```

and accepts the following parameters (please refer also to the `README` of the GitHub repository for a more precise description of these parameters):

- **DATASET**: the dataset on which apply the experiments. Can be one of the datasets described in Section 4.2.
- **METHOD**: debiaser method to use. Can be one of the debiaser methods employed in this paper or `biased` in case of no methods.
- **NUMBER OF FEATURES**: number of sensitive variables to identify the sensitive groups. Can be an integer up to 3.
- **SENSITIVE FEATURE**: optional parameter to specify the sensitive variables for the identification of the sensitive groups in case `NUMBER OF FEATURES` is equal to one or two. To ensure that the selected variables are truly sensitive, they must be among the three sensitive variables defined for each dataset in Section 4.2.
- **CLASSIFIER**: optional parameter to specify a classification method. Can be one of the classifiers employed in the experiments of this paper. The default is the Logistic Regression classifier.
- **CM**: optional boolean value to plot the confusion matrices of the two sensitive groups. Default is `false`.

The following command can also be executed to receive help and information on the requested parameters:

```
$ python generatemetrics.py -h
```

The execution of this script will produce a `.csv` file containing all the measures described in Section 4.1 for each train–test fold. We like to remark that, in case of DEMV, the number of measures will be equal to 30 times the number of train–test folds (see the description of the experiment setting in Section 4.1).

To reproduce the experiments of Section 4.3, the script can be called passing as input any of the involved dataset, a number of sensitive variables equal to 2, and `UNIFORM`, `SMOTE` or `ADASYN` as debiaser method. For instance:

```
$ python generatemetrics.py cmc uniform 2 --sensitivevariable religion,work
```

will produce the metrics of the CMC dataset using the DEMV Uniform debiaser strategy. Aggregating the results of the execution of this script for all of the involved datasets and all of the analyzed generation strategies makes possible to reproduce the experiments and charts shown in Section 4.3.

Similarly, it is possible to reproduce the experiments of Section 4.4 by running `generatemetrics.py` on all the combinations of datasets, methods, number of sensitive variables and classification methods and then aggregating the results. For instance, the following command:

```
$ python generatemetrics.py adult eg 3 --classifier gradient
```

will generate the metrics for the Adult dataset with three sensitive variables, using EG debiaser method and the Gradient Boosting classifier.

Finally, confusion matrices for the privileged and unprivileged groups can also be created using this script. For instance, the command:

```
$ python generatemetrics.py crime uniform 3 --cm
```

will generate the confusion matrices of the Crime dataset for the privileged and unprivileged groups.

It is also possible to refer to the `README` file on the GitHub repository for a more complete description of the method and the parameters.

[14] https://www.anaconda.com/
[15] https://pypi.org/project/pip/

5. Discussion

In this section, we discuss the results of the experiments conducted in Section 4, by referring to the research questions highlighted in Section 1. Hence, we can draw the following considerations:

- RQ1. From the experiments conducted in Section 4, we have seen that almost all the baselines can improve fairness in the binary classification context and classification problems involving one sensitive variable. However, no baselines can consistently handle bias in the multi-class classification domain with multiple sensitive variables either because they do not support it (like in the case of *Blackbox*) or because they perform very poorly (like in the case of *EG* and *Grid*). More specifically, the strengths and weaknesses we have observed for each baseline in the performed experiments are as follows:
  - **EG.** It can improve the fairness in the context of binary classification with very relevant results. It has much more difficulty in improving fairness in multi-class classification. This weakness might be imputed to the constraint metric suggested by the authors to be used in the case of multi-class classification, i.e., ZO Loss. In addition, this method is highly influenced by the involved classification algorithm and cannot be applied if the employed classifier is a Neural Network.
  - **Grid.** His performances are strictly related to the dataset and the search space size. Since, in our experiments, we always used a grid size of 20 (the default value of the adopted implementation), this method performed well with some datasets and worse with others in which a larger search space was needed. This results in very high variability of the overall obtained metrics. In particular, our experiments observed that Grid performs well with the CMC and Wine multi-class datasets. At the same time, in the binary classification task, the Grid method exposes a higher variability even in the same dataset but among a different number of sensitive variables. Finally, as for EG, this method is strongly influenced by the classification algorithm and cannot used if the employed classifier is a Neural Network.
  - **Blackbox.** This method performs well in mitigating bias in binary and multi-class classification. However, it does not support multiple sensitive variables. In addition, we observed high variability in the overall metrics that let the method be considered unstable.
- RQ2. In this work, we have presented the *Debiaser for Multiple Variables (DEMV)*, which is, to the best of our knowledge, the first *pre-processing* approach able to improve fairness both in binary and multi-class classification with multiple sensitive variables. DEMV generally overcomes the other baselines in binary and multi-class classification tasks with one, two, and three sensitive variables. In addition, being a pre-processing method, DEMV can be applied to a heterogeneous set of classification methods without impacting or being influenced by their behavior. DEMV is also the method that performs more consistently in all the experiments, resulting in less variability of the overall metrics.
- RQ3. The generative strategy that must be adopted to rebalance groups in DEMV is the Uniform sampling. The Uniform generating strategy is preferable from two points of view: (i) is the best performer (among the other generative strategies) in terms of fairness and accuracy; (ii) its computational complexity is negligible. This approach has been adopted in DEMV and compared with the other baselines obtaining excellent results.
- RQ4. The performed experiments showed that DEMV can improve the fairness in binary and multi-class classification contexts with any number of involved sensitive variables keeping a high level of accuracy. In particular, our method overcomes the other baselines in multi-class classification problems with any number of sensitive variables. In contrast, as expected, other specifically designed methods may perform better in binary classification with one or two sensitive variables. Instead, we have noticed that when the sensitive variables are more than two, DEMV overcomes the baselines also in the case of binary classification. In addition, we have shown how DEMV can consistently improve the fairness of several classification methods without impacting their behavior, while other debiaser methods behave differently according to the involved classification method or cannot be applied at all (like EG and Grid with a Neural Network). Finally, we observed that when the size of the sensitive group is tiny, DEMV has more difficulty improving fairness and finding the optimal group size. This issue can be explained by the fact that when the group size is small, the addition or removal of a single item impacts more on the expected and observed size ratio, so the optimal balancing is more complex or impossible to be achieved.

6. Conclusion and future work

In this paper, we addressed the problem of bias mitigation in the multi-class classification context by proposing the *Debiaser for Multiple Variables*, a novel approach extending the work of Kamiran and Calders (2012) to the multi-class classification domain with multiple sensitive variables. To the best of our knowledge, DEMV is the first pre-processing method able to handle bias in both binary and multi-class classification problems with any number of sensitive variables.

We have exhaustively evaluated our algorithm by comparing it with three established baselines using a heterogeneous set of binary and multi-class datasets, with a different number of sensitive variables, and by employing a heterogeneous set of classification methods. In addition, we have also evaluated how different instances generation strategies can influence the ability of DEMV in improving fairness. The conducted experiments show that our method is the better choice to adopt in the multi-class classification context with one, two, or three sensitive variables. Instead, we noticed how other specifically designed methods might perform better in binary classification with one and two sensitive variables. However, our method is still the better solution in binary classification

Table 6
Overall H-Mean of all methods with different classifiers in the multi-class classification context.

Classifier	Methods			
	No one	EG	Grid	DEMV
Logistic Regression	0.493 ± 0.16	0.505 ± 0.16	0.58 ± 0.063	**0.677 ± 0.081**
Gradient Boosting	0.653 ± 0.061	0.72 ± 0.049	0.607 ± 0.121	**0.729 ± 0.018**
SVM	0.603 ± 0.069	0.613 ± 0.054	0.392 ± 0.127	**0.716 ± 0.012**
Neural Network	0.656 ± 0.038	–	–	**0.728 ± 0.016**

Table 7
Evaluation results of generative strategies for binary datasets.

Data	Strategy	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	DEMV Uniform	**0.126 ± 0.03**	0.26 ± 0.124	**0.144 ± 0.015**	0.373 ± 0.137	**0.834 ± 0.005**	0.635 ± 0.117
	DEMV SMOTE	0.138 ± 0.014	**0.22 ± 0.149**	0.153 ± 0.011	0.242 ± 0.075	**0.834 ± 0.005**	0.543 ± 0.078
	DEMV ADASYN	**0.126 ± 0.03**	0.259 ± 0.123	0.145 ± 0.015	**0.374 ± 0.137**	**0.834 ± 0.005**	**0.636 ± 0.117**
Compas	DEMV Uniform	0.161 ± 0.063	0.29 ± 0.202	**0.124 ± 0.043**	0.773 ± 0.08	0.664 ± 0.016	0.75 ± 0.099
	DEMV SMOTE	**0.15 ± 0.043**	**0.266 ± 0.154**	0.133 ± 0.051	**0.79 ± 0.056**	**0.665 ± 0.016**	**0.767 ± 0.061**
	DEMV ADASYN	0.16 ± 0.063	0.288 ± 0.203	**0.124 ± 0.043**	0.774 ± 0.081	0.664 ± 0.016	0.75 ± 0.099
German	DEMV Uniform	**0.18 ± 0.134**	0.644 ± 0.342	0.278 ± 0.119	**0.772 ± 0.163**	**0.748 ± 0.038**	0.616 ± 0.157
	DEMV SMOTE	0.183 ± 0.138	**0.625 ± 0.381**	**0.276 ± 0.121**	0.771 ± 0.172	**0.748 ± 0.039**	**0.636 ± 0.162**
	DEMV ADASYN	0.181 ± 0.134	0.649 ± 0.353	**0.276 ± 0.12**	0.771 ± 0.163	0.747 ± 0.039	0.623 ± 0.146
Mean	DEMV Uniform	**0.156 ± 0.027**	0.398 ± 0.214	**0.182 ± 0.084**	0.639 ± 0.231	**0.749 ± 0.085**	0.667 ± 0.073
	DEMV SMOTE	0.157 ± 0.023	0.37 ± 0.222	0.187 ± 0.077	0.601 ± 0.311	**0.749 ± 0.085**	0.649 ± 0.113
	DEMV ADASYN	**0.156 ± 0.028**	**0.399 ± 0.217**	**0.182 ± 0.082**	**0.64 ± 0.23**	0.748 ± 0.085	**0.67 ± 0.07**

Table 8
Evaluation results of generative strategies for multi-class datasets.

Data	Strategy	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	DEMV Uniform	0.056 ± 0.029	0.206 ± 0.191	**0.233 ± 0.102**	0.663 ± 0.157	0.512 ± 0.038	0.694 ± 0.074
	DEMV SMOTE	**0.048 ± 0.033**	**0.195 ± 0.138**	0.273 ± 0.098	**0.722 ± 0.138**	0.51 ± 0.038	**0.704 ± 0.051**
	DEMV ADASYN	0.054 ± 0.03	0.213 ± 0.172	0.255 ± 0.101	0.68 ± 0.168	**0.514 ± 0.038**	0.693 ± 0.07
Crime	DEMV Uniform	**0.202 ± 0.049**	0.32 ± 0.138	**0.164 ± 0.044**	**0.365 ± 0.143**	0.441 ± 0.028	**0.542 ± 0.076**
	DEMV SMOTE	0.242 ± 0.048	**0.309 ± 0.146**	0.181 ± 0.039	0.292 ± 0.124	**0.456 ± 0.03**	0.501 ± 0.08
	DEMV ADASYN	0.215 ± 0.051	0.316 ± 0.162	0.175 ± 0.054	0.271 ± 0.151	0.454 ± 0.033	0.476 ± 0.122
Drug	DEMV Uniform	**0.148 ± 0.047**	**0.17 ± 0.072**	0.337 ± 0.109	0.486 ± 0.13	0.675 ± 0.025	0.662 ± 0.062
	DEMV SMOTE	0.185 ± 0.056	0.184 ± 0.058	**0.328 ± 0.108**	0.41 ± 0.121	**0.68 ± 0.029**	0.624 ± 0.069
	DEMV ADASYN	0.134 ± 0.054	0.198 ± 0.085	0.345 ± 0.106	**0.519 ± 0.121**	0.671 ± 0.025	**0.67 ± 0.059**
Law	DEMV Uniform	**0.041 ± 0.028**	0.145 ± 0.063	0.159 ± 0.022	**0.887 ± 0.077**	0.512 ± 0.011	**0.77 ± 0.019**
	DEMV SMOTE	0.095 ± 0.031	0.144 ± 0.058	0.172 ± 0.021	0.741 ± 0.087	**0.515 ± 0.01**	0.737 ± 0.023
	DEMV ADASYN	0.044 ± 0.031	**0.14 ± 0.055**	**0.153 ± 0.014**	0.883 ± 0.08	0.511 ± 0.012	**0.77 ± 0.02**
Park	DEMV Uniform	0.062 ± 0.048	0.073 ± 0.046	**0.211 ± 0.047**	0.809 ± 0.136	0.493 ± 0.024	**0.746 ± 0.042**
	DEMV SMOTE	0.067 ± 0.049	0.085 ± 0.044	0.22 ± 0.048	0.796 ± 0.136	**0.496 ± 0.024**	0.742 ± 0.048
	DEMV ADASYN	**0.057 ± 0.032**	**0.071 ± 0.043**	**0.211 ± 0.052**	**0.812 ± 0.1**	0.478 ± 0.023	0.742 ± 0.036
Wine	DEMV Uniform	0.106 ± 0.038	0.478 ± 0.264	**0.078 ± 0.03**	0.858 ± 0.047	0.519 ± 0.018	0.646 ± 0.177
	DEMV SMOTE	0.174 ± 0.052	0.787 ± 0.369	0.138 ± 0.046	0.772 ± 0.062	**0.538 ± 0.016**	0.566 ± 0.154
	DEMV ADASYN	**0.096 ± 0.039**	**0.34 ± 0.215**	0.083 ± 0.028	**0.864 ± 0.053**	0.515 ± 0.018	**0.722 ± 0.073**
Mean	DEMV Uniform	0.102 ± 0.063	0.232 ± 0.145	**0.197 ± 0.087**	**0.678 ± 0.214**	**0.525 ± 0.079**	0.677 ± 0.081
	DEMV SMOTE	0.135 ± 0.077	0.284 ± 0.257	0.219 ± 0.071	0.622 ± 0.215	0.533 ± 0.077	0.646 ± 0.099
	DEMV ADASYN	**0.1 ± 0.066**	**0.213 ± 0.102**	0.204 ± 0.09	0.672 ± 0.239	0.524 ± 0.076	**0.679 ± 0.105**

problems with three sensitive variables and, being a pre-processing method, it can be successfully applied even with classifiers not supported by other baselines (e.g., Neural Networks).

Finally, we have seen that the Uniform sampling of existing instances is the best strategy to manipulate groups in terms of accuracy and fairness.

In the future, we want to overcome the current main weakness of DEMV, highlighted when the sensitive groups are very small. We will address this issue by investigating if there are situations that lead to optimal fairness before a complete balance within the groups. If so, we want to identify further which are the characteristics that leads to these situations. In addition, we want to improve our analysis by studying the impact of the size of the dataset and the number of their attributes that are not the sensitive variables on DEMV and his behavior. Next, we also want to assess the computational complexity respect to the other baselines and we want to study the impact that different removal strategies may have during the balancing procedure and thus on the capacity of DEMV to mitigate bias and improving fairness. Finally, we will study the impact that DEMV have on the fairness of the full pipeline of recommender systems that embed a multi-class classifier.

Table 9
Evaluation results for all binary datasets and methods with one sensitive variables.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	No one	0.139 ± 0.017	**0.104 ± 0.063**	0.094 ± 0.033	0.34 ± 0.062	0.835 ± 0.007	0.659 ± 0.054
	Blackbox	0.061 ± 0.021	0.253 ± 0.065	0.094 ± 0.033	0.659 ± 0.045	0.835 ± 0.007	0.802 ± 0.032
	EG	**0.02 ± 0.015**	0.238 ± 0.072	0.078 ± 0.035	**0.892 ± 0.079**	0.827 ± 0.008	**0.868 ± 0.026**
	Grid	0.066 ± 0.021	0.164 ± 0.074	0.09 ± 0.033	0.662 ± 0.11	0.833 ± 0.005	0.818 ± 0.039
	DEMV	0.094 ± 0.023	0.135 ± 0.062	0.093 ± 0.03	0.57 ± 0.098	0.834 ± 0.006	0.787 ± 0.044
Compas	No one	0.217 ± 0.067	0.759 ± 0.499	0.041 ± 0.033	0.727 ± 0.066	0.67 ± 0.019	0.61 ± 0.216
	Blackbox	0.064 ± 0.048	**0.112 ± 0.05**	0.041 ± 0.033	0.831 ± 0.114	0.67 ± 0.019	0.84 ± 0.037
	EG	**0.035 ± 0.027**	0.158 ± 0.08	0.035 ± 0.028	**0.947 ± 0.039**	0.662 ± 0.015	**0.857 ± 0.022**
	Grid	0.183 ± 0.044	0.442 ± 0.172	0.051 ± 0.036	0.731 ± 0.07	0.657 ± 0.023	0.706 ± 0.081
	DEMV	0.117 ± 0.053	0.212 ± 0.154	0.037 ± 0.028	0.832 ± 0.068	0.665 ± 0.018	0.807 ± 0.066
German	No one	0.166 ± 0.105	0.549 ± 0.359	0.112 ± 0.084	0.798 ± 0.125	0.741 ± 0.045	0.676 ± 0.19
	Blackbox	**0.025 ± 0.026**	**0.191 ± 0.108**	0.099 ± 0.069	**0.963 ± 0.038**	0.741 ± 0.028	**0.864 ± 0.025**
	EG	0.084 ± 0.056	0.641 ± 1.016	0.085 ± 0.091	0.897 ± 0.068	0.746 ± 0.042	0.78 ± 0.139
	Grid	0.133 ± 0.072	1.395 ± 1.629	0.1 ± 0.096	0.843 ± 0.083	0.746 ± 0.038	0.76 ± 0.175
	DEMV	0.119 ± 0.088	0.563 ± 0.419	0.098 ± 0.077	0.851 ± 0.102	0.748 ± 0.039	0.737 ± 0.112
Mean	No one	0.174 ± 0.04	0.471 ± 0.334	0.082 ± 0.037	0.622 ± 0.247	0.749 ± 0.083	0.648 ± 0.034
	Blackbox	0.05 ± 0.022	**0.185 ± 0.071**	0.078 ± 0.032	0.818 ± 0.152	0.749 ± 0.083	**0.835 ± 0.031**
	EG	**0.046 ± 0.033**	0.346 ± 0.259	0.066 ± 0.027	**0.912 ± 0.03**	0.745 ± 0.083	**0.835 ± 0.048**
	Grid	0.127 ± 0.059	0.667 ± 0.646	0.08 ± 0.026	0.745 ± 0.091	0.745 ± 0.088	0.761 ± 0.056
	DEMV	0.11 ± 0.014	0.303 ± 0.228	0.076 ± 0.034	0.751 ± 0.157	0.749 ± 0.085	0.777 ± 0.036

Table 10
Evaluation results for all binary datasets and methods with two sensitive variables.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	No one	0.17 ± 0.017	**0.17 ± 0.136**	0.156 ± 0.01	0.174 ± 0.071	**0.835 ± 0.007**	0.455 ± 0.107
	EG	**0.021 ± 0.012**	0.396 ± 0.101	**0.117 ± 0.019**	**0.871 ± 0.08**	0.82 ± 0.005	**0.805 ± 0.046**
	Grid	0.366 ± 0.007	0.52 ± 0.014	0.237 ± 0.012	0.0 ± 0.0	0.771 ± 0.005	0.0 ± 0.0
	DEMV	0.1 ± 0.021	0.284 ± 0.112	0.141 ± 0.015	0.475 ± 0.109	0.832 ± 0.004	0.706 ± 0.072
Compas	No one	0.241 ± 0.038	0.55 ± 0.212	0.127 ± 0.047	0.678 ± 0.045	**0.67 ± 0.019**	0.621 ± 0.13
	EG	**0.044 ± 0.027**	**0.161 ± 0.085**	**0.114 ± 0.048**	**0.932 ± 0.039**	0.644 ± 0.025	**0.833 ± 0.029**
	Grid	0.294 ± 0.235	0.396 ± 0.148	0.276 ± 0.064	0.593 ± 0.198	0.584 ± 0.016	0.592 ± 0.174
	DEMV	0.116 ± 0.048	0.185 ± 0.119	0.119 ± 0.042	0.831 ± 0.067	0.662 ± 0.016	0.802 ± 0.046
German	No one	0.206 ± 0.139	0.647 ± 0.41	0.317 ± 0.123	0.743 ± 0.173	0.741 ± 0.046	0.597 ± 0.187
	EG	0.116 ± 0.093	0.833 ± 0.764	0.264 ± 0.121	**0.86 ± 0.117**	**0.749 ± 0.039**	**0.687 ± 0.198**
	Grid	**0.691 ± 0.06**	0.811 ± 0.049	0.44 ± 0.108	0.0 ± 0.0	0.67 ± 0.024	0.0 ± 0.0
	DEMV	0.148 ± 0.131	**0.628 ± 0.373**	**0.26 ± 0.126**	0.81 ± 0.157	**0.749 ± 0.036**	0.662 ± 0.105
Mean	No one	0.206 ± 0.036	0.456 ± 0.252	0.2 ± 0.102	0.532 ± 0.311	**0.749 ± 0.083**	0.558 ± 0.09
	EG	**0.06 ± 0.05**	0.463 ± 0.341	**0.165 ± 0.086**	**0.888 ± 0.039**	0.738 ± 0.089	**0.775 ± 0.077**
	Grid	0.45 ± 0.212	0.576 ± 0.213	0.318 ± 0.108	0.198 ± 0.342	0.675 ± 0.094	0.197 ± 0.342
	DEMV	0.121 ± 0.024	**0.366 ± 0.233**	0.173 ± 0.076	0.705 ± 0.2	0.748 ± 0.085	0.723 ± 0.072

Table 11
Evaluation results for all binary datasets and methods with three sensitive variables.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	No one	0.179 ± 0.016	**0.248 ± 0.189**	0.286 ± 0.077	0.124 ± 0.073	0.835 ± 0.007	0.352 ± 0.138
	EG	0.179 ± 0.012	0.259 ± 0.21	0.271 ± 0.063	0.128 ± 0.06	0.829 ± 0.007	0.367 ± 0.115
	Grid	0.35 ± 0.112	0.434 ± 0.099	0.312 ± 0.031	0.119 ± 0.251	0.757 ± 0.022	0.207 ± 0.221
	DEMV	**0.096 ± 0.023**	0.325 ± 0.15	0.31 ± 0.06	**0.465 ± 0.136**	0.821 ± 0.005	**0.658 ± 0.105**
Compas	No one	0.244 ± 0.045	0.612 ± 0.204	0.351 ± 0.092	0.683 ± 0.055	0.653 ± 0.02	0.56 ± 0.142
	EG	0.262 ± 0.038	0.666 ± 0.254	0.361 ± 0.082	0.661 ± 0.046	0.652 ± 0.02	0.468 ± 0.243
	Grid	0.215 ± 0.055	0.434 ± 0.141	0.339 ± 0.102	0.707 ± 0.073	0.641 ± 0.015	0.657 ± 0.069
	DEMV	**0.1 ± 0.045**	**0.215 ± 0.127**	0.323 ± 0.1	**0.861 ± 0.06**	0.648 ± 0.023	**0.758 ± 0.061**
German	No one	0.191 ± 0.073	0.851 ± 0.504	0.552 ± 0.135	**0.786 ± 0.08**	0.744 ± 0.042	0.542 ± 0.138
	EG	0.236 ± 0.118	0.554 ± 0.281	0.631 ± 0.131	0.717 ± 0.152	0.729 ± 0.033	0.527 ± 0.13
	Grid	**0.181 ± 0.154**	**0.404 ± 0.171**	0.565 ± 0.179	0.775 ± 0.191	0.694 ± 0.034	**0.593 ± 0.124**
	DEMV	0.188 ± 0.07	0.831 ± 0.476	0.512 ± 0.157	**0.786 ± 0.076**	0.747 ± 0.034	0.536 ± 0.18
Mean	No one	0.205 ± 0.035	0.57 ± 0.304	0.396 ± 0.139	0.531 ± 0.356	0.744 ± 0.091	0.485 ± 0.115
	EG	0.226 ± 0.042	0.493 ± 0.21	0.421 ± 0.187	0.502 ± 0.325	0.737 ± 0.089	0.454 ± 0.081
	Grid	0.249 ± 0.089	**0.424 ± 0.017**	0.405 ± 0.139	0.534 ± 0.361	0.697 ± 0.058	0.486 ± 0.243
	DEMV	**0.128 ± 0.052**	0.457 ± 0.329	0.382 ± 0.113	**0.704 ± 0.21**	0.739 ± 0.087	**0.651 ± 0.111**

Table 12
Evaluation results for binary datasets using Gradient Boosting classifier.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	No one	0.154 ± 0.017	0.225 ± 0.123	0.158 ± 0.009	0.175 ± 0.073	0.833 ± 0.006	0.454 ± 0.099
	EG	0.166 ± 0.013	0.438 ± 0.298	0.155 ± 0.012	0.062 ± 0.059	0.829 ± 0.006	0.205 ± 0.175
	Grid	0.156 ± 0.013	**0.16 ± 0.13**	0.16 ± 0.01	0.151 ± 0.067	0.832 ± 0.007	0.421 ± 0.098
	DEMV	**0.099 ± 0.022**	0.299 ± 0.144	0.145 ± 0.014	**0.443 ± 0.118**	0.831 ± 0.005	**0.684 ± 0.084**
Compas	No one	0.234 ± 0.039	0.185 ± 0.047	0.092 ± 0.035	0.546 ± 0.059	0.689 ± 0.019	0.722 ± 0.035
	EG	0.207 ± 0.042	0.164 ± 0.047	0.08 ± 0.033	0.594 ± 0.072	0.686 ± 0.018	0.746 ± 0.04
	Grid	0.208 ± 0.041	0.166 ± 0.046	0.083 ± 0.042	0.591 ± 0.072	0.686 ± 0.019	0.744 ± 0.038
	DEMV	**0.179 ± 0.04**	**0.136 ± 0.043**	0.098 ± 0.039	**0.632 ± 0.069**	0.687 ± 0.017	**0.765 ± 0.035**
Mean	No one	0.194 ± 0.057	0.205 ± 0.028	0.125 ± 0.047	0.361 ± 0.262	0.761 ± 0.102	0.588 ± 0.19
	EG	0.186 ± 0.029	0.301 ± 0.194	0.118 ± 0.053	0.328 ± 0.376	0.758 ± 0.101	0.476 ± 0.383
	Grid	0.182 ± 0.037	**0.163 ± 0.004**	0.122 ± 0.054	0.371 ± 0.311	0.759 ± 0.103	0.582 ± 0.228
	DEMV	**0.139 ± 0.057**	0.218 ± 0.115	0.122 ± 0.033	**0.538 ± 0.134**	0.759 ± 0.102	**0.724 ± 0.057**

Table 13
Evaluation results for binary datasets using Support Vector Machines classifier.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	No one	0.165 ± 0.013	0.185 ± 0.133	0.167 ± 0.009	0.151 ± 0.042	0.831 ± 0.005	0.428 ± 0.074
	EG	0.162 ± 0.014	0.175 ± 0.126	0.163 ± 0.013	0.132 ± 0.057	0.828 ± 0.006	0.386 ± 0.124
	Grid	0.162 ± 0.021	0.208 ± 0.146	0.162 ± 0.017	0.178 ± 0.092	0.828 ± 0.006	0.445 ± 0.155
	DEMV	**0.096 ± 0.021**	0.298 ± 0.155	0.151 ± 0.017	**0.431 ± 0.116**	0.825 ± 0.006	**0.674 ± 0.083**
Compas	No one	0.191 ± 0.039	0.129 ± 0.029	0.129 ± 0.066	0.523 ± 0.074	0.645 ± 0.021	0.712 ± 0.037
	EG	0.196 ± 0.039	0.14 ± 0.032	0.132 ± 0.048	0.559 ± 0.073	0.643 ± 0.02	0.722 ± 0.033
	Grid	0.179 ± 0.037	0.134 ± 0.044	0.139 ± 0.06	0.588 ± 0.072	0.646 ± 0.019	0.735 ± 0.036
	DEMV	**0.121 ± 0.039**	0.123 ± 0.049	0.122 ± 0.053	**0.67 ± 0.094**	0.632 ± 0.019	**0.768 ± 0.038**
Mean	No one	0.178 ± 0.018	0.157 ± 0.04	0.148 ± 0.027	0.337 ± 0.263	0.738 ± 0.132	0.57 ± 0.201
	EG	0.179 ± 0.024	0.158 ± 0.025	0.148 ± 0.022	0.346 ± 0.302	0.736 ± 0.131	0.554 ± 0.238
	Grid	0.17 ± 0.012	0.171 ± 0.052	0.151 ± 0.016	0.383 ± 0.29	0.737 ± 0.129	0.59 ± 0.205
	DEMV	**0.108 ± 0.018**	0.21 ± 0.124	0.136 ± 0.021	**0.55 ± 0.169**	0.728 ± 0.136	**0.721 ± 0.066**

CRediT authorship contribution statement

Giordano d'Aloisio: Methodology, Software, Validation, Investigation, Visualization, Writing – original draft, Writing – review & editing. **Andrea D'Angelo:** Software, Validation, Investigation, Visualization, Writing – review & editing. **Antinisca Di Marco:** Methodology, Writing – review & editing, Supervision, Project administration. **Giovanni Stilo:** Conceptualization, Methodology, Validation, Investigation, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Data availability

Links to code and data are available in the paper.

Acknowledgments

This work was supported in part by Territori Aperti (a project funded by Fondo Territori, Lavoro e Conoscenza CGIL CISL UIL), and in part by Ruropean Union – Horizon 2020 Program under the scheme "INFRAIA-01-2018-2019 – Integrating Activities for Advanced Communities", Grant Agreement n. 871042, "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (http://www.sobigdata.eu), European Union - NextGenerationEU - National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) - Project: "SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013 - Avviso n. 3264 del 28/12/2021 and in part by "FAIR-EDU: Promote FAIRness in EDUcation institutions" a project founded by the University of L'Aquila, 2022, and by EMELIOT national research project, which has been funded by the MUR under the PRIN 2020 program (Contract 2020W3A5FY). All the numerical simulations have been realized on the Linux HPC cluster Caliban of the High-Performance Computing Laboratory of the Department of Information Engineering, Computer Science and Mathematics (DISIM) at the University of L'Aquila.

Appendix A. Detailed results of generative strategies' comparison

In the following, we report the detailed results of the evaluation of DEMV's generative strategies. For each dataset and for each method, we report the mean and standard deviation of all metrics. In addition, we report the mean and standard deviation of the H-Mean computed from the obtained values. Finally, we also report the overall means and standard deviations of all the values obtained by each method in each experiment. For each dataset, we highlight in boldface the best value of each metric

In particular, Table 7 shows the results for binary datasets, while Table 8 describes the results for multi-class datasets.

Table 14
Evaluation results for binary datasets using Neural Network classifier.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
Adult	No one	0.185 ± 0.031	0.23 ± 0.174	0.17 ± 0.013	0.17 ± 0.074	0.819 ± 0.008	0.441 ± 0.133
	EG	Not applicable					
	Grid	Not applicable					
	DEMV	**0.144 ± 0.028**	0.24 ± 0.136	0.16 ± 0.017	**0.315 ± 0.101**	0.815 ± 0.006	**0.6 ± 0.096**
Compas	No one	0.214 ± 0.053	0.193 ± 0.075	0.089 ± 0.061	0.583 ± 0.068	0.652 ± 0.017	0.727 ± 0.045
	EG	Not applicable					
	Grid	Not applicable					
	DEMV	**0.136 ± 0.046**	0.15 ± 0.053	0.12 ± 0.059	**0.719 ± 0.077**	0.651 ± 0.017	**0.779 ± 0.035**
Mean	No one	0.2 ± 0.021	0.212 ± 0.026	0.13 ± 0.057	0.376 ± 0.292	0.736 ± 0.118	0.584 ± 0.202
	EG	Not applicable					
	Grid	Not applicable					
	DEMV	**0.14 ± 0.006**	0.195 ± 0.064	0.14 ± 0.028	**0.517 ± 0.286**	0.733 ± 0.116	**0.69 ± 0.127**

Table 15
Evaluation results for all multi-class datasets and methods using one sensitive variables.

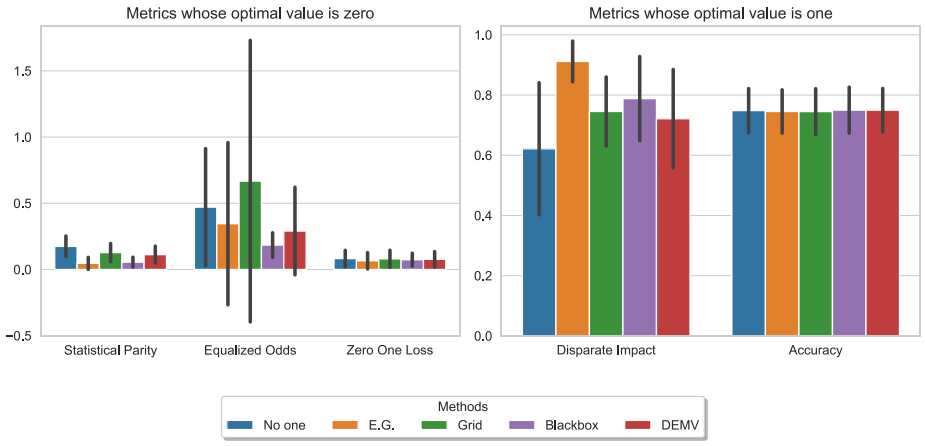
Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	No one	0.188 ± 0.15	0.305 ± 0.231	0.122 ± 0.081	0.51 ± 0.282	0.521 ± 0.039	0.6 ± 0.166
	Blackbox	0.125 ± 0.1	0.275 ± 0.169	0.098 ± 0.08	0.539 ± 0.308	0.515 ± 0.042	0.606 ± 0.185
	EG	0.162 ± 0.128	0.292 ± 0.193	0.111 ± 0.074	0.536 ± 0.279	0.505 ± 0.038	0.62 ± 0.138
	Grid	0.089 ± 0.086	0.278 ± 0.231	0.105 ± 0.057	**0.748 ± 0.154**	0.501 ± 0.04	**0.699 ± 0.131**
	DEMV	**0.088 ± 0.072**	0.254 ± 0.146	**0.092 ± 0.072**	0.641 ± 0.224	0.516 ± 0.038	0.687 ± 0.107
Crime	No one	0.389 ± 0.082	0.329 ± 0.113	0.069 ± 0.049	0.182 ± 0.076	0.497 ± 0.028	0.409 ± 0.109
	Blackbox	0.425 ± 0.074	0.884 ± 0.314	0.069 ± 0.049	0.097 ± 0.082	0.497 ± 0.028	0.186 ± 0.118
	EG	0.39 ± 0.084	0.332 ± 0.112	0.066 ± 0.05	0.179 ± 0.077	0.496 ± 0.03	0.403 ± 0.118
	Grid	0.3 ± 0.111	0.399 ± 0.135	0.117 ± 0.06	0.336 ± 0.182	0.433 ± 0.039	0.487 ± 0.124
	DEMV	**0.253 ± 0.064**	**0.317 ± 0.109**	**0.062 ± 0.034**	0.377 ± 0.106	0.47 ± 0.029	**0.568 ± 0.063**
Drug	No one	0.264 ± 0.121	0.308 ± 0.236	0.142 ± 0.076	0.343 ± 0.216	0.68 ± 0.025	0.542 ± 0.183
	Blackbox	0.441 ± 0.144	0.806 ± 0.58	0.145 ± 0.073	0.095 ± 0.047	0.683 ± 0.025	0.268 ± 0.087
	EG	0.246 ± 0.107	0.267 ± 0.135	0.137 ± 0.093	0.371 ± 0.193	0.68 ± 0.026	0.583 ± 0.153
	Grid	0.26 ± 0.117	0.298 ± 0.245	0.134 ± 0.091	0.336 ± 0.201	0.683 ± 0.025	0.541 ± 0.189
	DEMV	**0.128 ± 0.083**	0.218 ± 0.112	**0.126 ± 0.058**	**0.585 ± 0.199**	0.678 ± 0.026	**0.72 ± 0.091**
Law	No one	0.26 ± 0.04	0.31 ± 0.038	0.072 ± 0.04	0.441 ± 0.128	0.521 ± 0.01	0.61 ± 0.062
	Blackbox	0.179 ± 0.035	0.231 ± 0.093	0.072 ± 0.04	0.408 ± 0.206	0.521 ± 0.01	0.584 ± 0.119
	EG	0.231 ± 0.048	0.264 ± 0.044	0.072 ± 0.037	0.487 ± 0.134	0.521 ± 0.009	0.64 ± 0.064
	Grid	0.176 ± 0.133	0.228 ± 0.149	0.104 ± 0.093	0.626 ± 0.289	0.503 ± 0.013	0.67 ± 0.135
	DEMV	**0.103 ± 0.024**	0.126 ± 0.039	**0.06 ± 0.03**	**0.757 ± 0.056**	0.518 ± 0.011	**0.76 ± 0.02**
Park	No one	0.221 ± 0.042	0.207 ± 0.055	**0.084 ± 0.064**	0.473 ± 0.075	0.503 ± 0.029	0.643 ± 0.046
	Blackbox	0.21 ± 0.092	0.381 ± 0.2	0.087 ± 0.057	0.334 ± 0.164	0.504 ± 0.024	0.496 ± 0.148
	EG	0.216 ± 0.051	0.23 ± 0.072	0.171 ± 0.079	0.461 ± 0.091	0.49 ± 0.022	0.622 ± 0.056
	Grid	0.221 ± 0.056	0.228 ± 0.05	0.172 ± 0.077	0.454 ± 0.11	0.493 ± 0.024	0.619 ± 0.064
	DEMV	**0.111 ± 0.054**	0.157 ± 0.043	0.085 ± 0.05	**0.697 ± 0.121**	0.502 ± 0.022	**0.728 ± 0.037**
Wine	No one	0.342 ± 0.165	1.067 ± 0.734	0.043 ± 0.033	0.544 ± 0.172	0.56 ± 0.02	0.607 ± 0.133
	Blackbox	**0.056 ± 0.043**	0.192 ± 0.132	0.048 ± 0.029	0.675 ± 0.223	0.561 ± 0.02	**0.735 ± 0.123**
	EG	0.338 ± 0.17	1.075 ± 0.756	0.043 ± 0.036	0.552 ± 0.179	0.56 ± 0.019	0.624 ± 0.104
	Grid	0.363 ± 0.206	0.761 ± 0.233	0.204 ± 0.184	0.453 ± 0.346	0.498 ± 0.043	0.38 ± 0.196
	DEMV	0.195 ± 0.078	0.84 ± 0.642	**0.033 ± 0.025**	**0.737 ± 0.077**	0.545 ± 0.022	0.628 ± 0.186
Mean	No one	0.277 ± 0.075	0.421 ± 0.319	0.089 ± 0.037	0.416 ± 0.134	0.547 ± 0.069	0.568 ± 0.085
	Blackbox	0.239 ± 0.159	0.462 ± 0.305	0.087 ± 0.033	0.358 ± 0.234	0.547 ± 0.07	0.479 ± 0.211
	EG	0.264 ± 0.084	0.41 ± 0.328	0.1 ± 0.048	0.431 ± 0.139	0.542 ± 0.072	0.582 ± 0.09
	Grid	0.235 ± 0.096	0.365 ± 0.204	0.139 ± 0.041	0.492 ± 0.164	0.518 ± 0.085	0.566 ± 0.121
	DEMV	**0.146 ± 0.064**	0.319 ± 0.264	**0.076 ± 0.032**	**0.632 ± 0.14**	0.538 ± 0.073	**0.682 ± 0.072**

Appendix B. Detailed results for binary classification

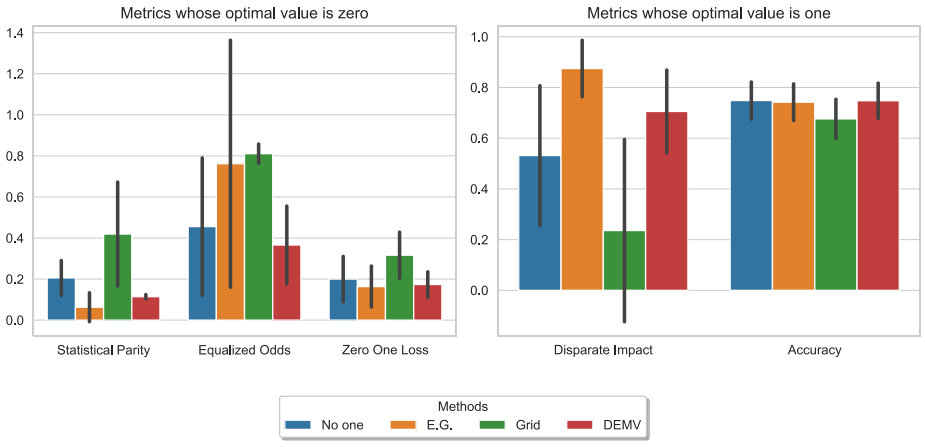
In the following, we report the charts and the detailed results for binary classification. Concerning the experiment with one sensitive variable we report the mean of the measures of both experiments taking each sensitive variable singularly.

Fig. 12 reports the means and standard deviations obtained in all three experiments. As noticed above, EG is the method performing better when one or two sensitive variables are involved, while it is not able to manage groups identified by three sensitive variables.

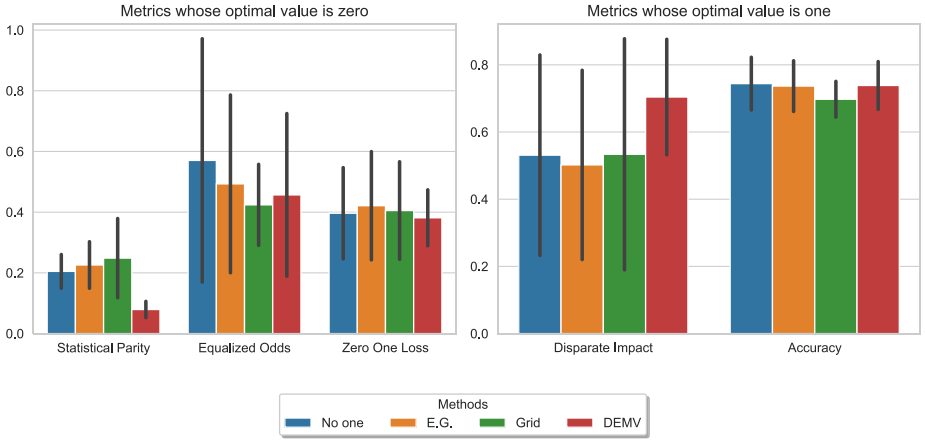
Tables 9, 10, and 11 reports the detailed results for each dataset. For each dataset, we highlight in boldface the best value of each metric whose differences are statistically significant. As mentioned above, we like to remark that when dealing with a binary dataset with one single sensitive variable, DEMV coincides with the *Sampling* method of Kamiran and Calders (2012).



(a) Application with one sensitive variable



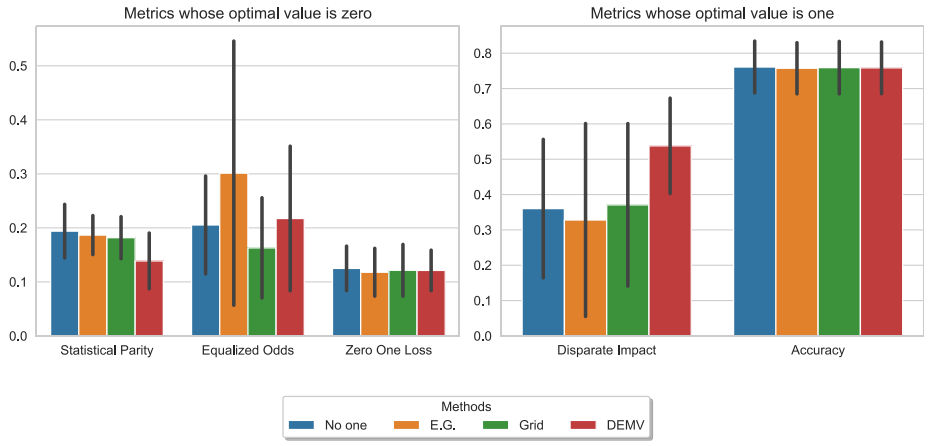
(b) Application with two sensitive variables



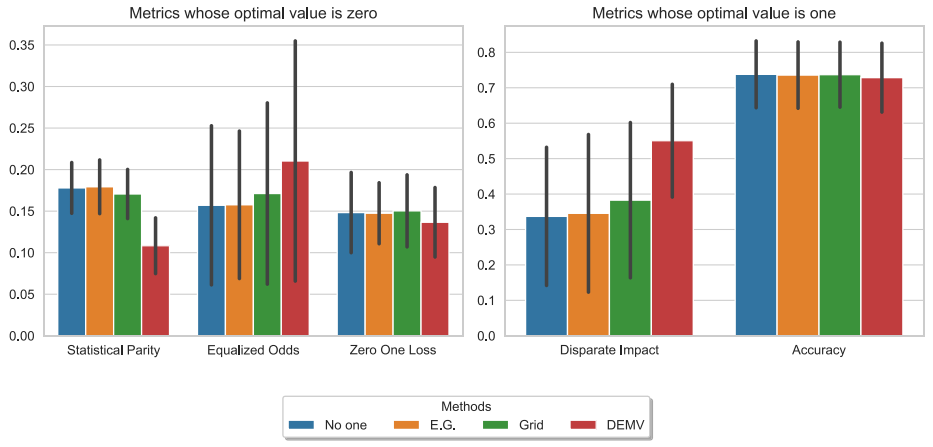
(c) Application with three sensitive variables

Fig. 12. Comparison of DEMV with the baselines in binary classification.

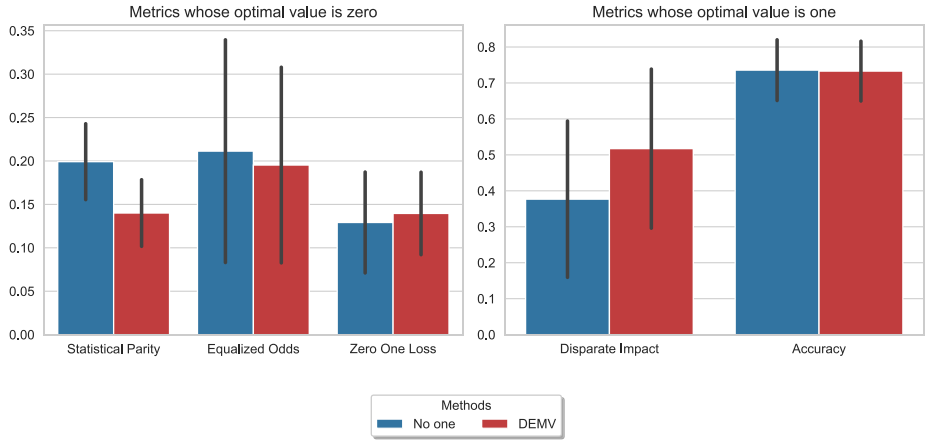
Fig. 13 reports instead the overall mean and standard deviation of all the metrics computed in the experiments involving more complex classifiers. It can be seen how, differently from the experiments involving a Logistic Regression model, DEMV overcomes



(a) Application with Gradient Boosting



(b) Application with Support Vector Machines



(c) Application with Neural Networks

Fig. 13. Comparison of DEMV with the baselines in binary classification using other classifiers.

the other baselines in all the experiments, with the only exception of EO with SVM in which the best method is EG. As already said, we like to remark that EG and Grid cannot be applied with a Neural Network model.

Table 16
Evaluation results for all multi-class datasets and methods using two sensitive variables.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	No one	0.126 ± 0.034	0.219 ± 0.118	0.33 ± 0.155	0.494 ± 0.128	**0.521 ± 0.04**	0.62 ± 0.058
	EG	0.107 ± 0.045	0.218 ± 0.15	0.35 ± 0.171	0.543 ± 0.173	0.509 ± 0.035	0.617 ± 0.081
	Grid	0.079 ± 0.049	0.241 ± 0.109	0.26 ± 0.176	0.815 ± 0.115	0.445 ± 0.049	0.679 ± 0.062
	DEMV	**0.056 ± 0.029**	**0.206 ± 0.191**	**0.233 ± 0.102**	**0.663 ± 0.157**	0.512 ± 0.038	**0.694 ± 0.074**
Crime	No one	0.339 ± 0.051	0.442 ± 0.139	0.209 ± 0.07	0.09 ± 0.066	**0.497 ± 0.029**	0.261 ± 0.108
	EG	0.332 ± 0.052	0.458 ± 0.166	0.212 ± 0.084	0.091 ± 0.074	0.493 ± 0.029	0.252 ± 0.139
	Grid	0.217 ± 0.077	0.335 ± 0.091	0.318 ± 0.077	**0.445 ± 0.136**	0.34 ± 0.042	0.515 ± 0.039
	DEMV	**0.202 ± 0.049**	**0.32 ± 0.138**	**0.164 ± 0.044**	0.365 ± 0.143	0.441 ± 0.028	**0.542 ± 0.076**
Drug	No one	0.299 ± 0.055	0.319 ± 0.15	0.335 ± 0.103	0.142 ± 0.086	0.68 ± 0.026	0.357 ± 0.144
	EG	0.272 ± 0.047	0.23 ± 0.118	0.375 ± 0.087	0.198 ± 0.068	**0.681 ± 0.032**	0.448 ± 0.073
	Grid	0.198 ± 0.057	0.193 ± 0.073	**0.331 ± 0.101**	0.356 ± 0.182	0.653 ± 0.025	0.574 ± 0.104
	DEMV	**0.148 ± 0.047**	**0.17 ± 0.072**	0.337 ± 0.109	**0.486 ± 0.13**	0.675 ± 0.025	**0.662 ± 0.062**
Law	No one	0.2 ± 0.027	0.2 ± 0.029	0.164 ± 0.03	0.502 ± 0.072	**0.521 ± 0.01**	0.655 ± 0.033
	EG	0.248 ± 0.031	0.308 ± 0.043	0.184 ± 0.027	0.456 ± 0.076	0.509 ± 0.013	0.61 ± 0.033
	Grid	0.3 ± 0.057	0.359 ± 0.09	0.193 ± 0.026	0.351 ± 0.086	0.508 ± 0.013	0.546 ± 0.067
	DEMV	**0.041 ± 0.028**	**0.145 ± 0.063**	**0.159 ± 0.022**	**0.887 ± 0.077**	0.512 ± 0.011	**0.77 ± 0.019**
Park	No one	0.208 ± 0.041	0.218 ± 0.072	0.246 ± 0.067	0.424 ± 0.089	**0.503 ± 0.026**	0.603 ± 0.05
	EG	0.272 ± 0.03	0.216 ± 0.041	0.324 ± 0.107	0.185 ± 0.044	0.481 ± 0.012	0.424 ± 0.053
	Grid	0.125 ± 0.031	0.127 ± 0.038	0.446 ± 0.063	0.617 ± 0.081	0.463 ± 0.019	0.631 ± 0.025
	DEMV	**0.062 ± 0.048**	**0.073 ± 0.046**	**0.211 ± 0.047**	**0.809 ± 0.136**	0.493 ± 0.024	**0.746 ± 0.042**
Wine	No one	0.322 ± 0.039	0.768 ± 0.185	0.146 ± 0.053	0.534 ± 0.048	**0.56 ± 0.021**	0.461 ± 0.117
	EG	0.189 ± 0.066	**0.35 ± 0.163**	0.178 ± 0.08	0.692 ± 0.096	0.541 ± 0.021	0.649 ± 0.066
	Grid	0.153 ± 0.042	0.616 ± 0.234	0.425 ± 0.055	0.77 ± 0.049	0.435 ± 0.021	0.535 ± 0.085
	DEMV	**0.106 ± 0.038**	0.478 ± 0.264	**0.078 ± 0.03**	**0.858 ± 0.047**	0.519 ± 0.018	**0.676 ± 0.177**
Mean	No one	0.249 ± 0.084	0.361 ± 0.219	0.238 ± 0.081	0.364 ± 0.196	**0.547 ± 0.069**	0.493 ± 0.16
	EG	0.237 ± 0.078	0.297 ± 0.096	0.27 ± 0.089	0.361 ± 0.238	0.536 ± 0.074	0.505 ± 0.16
	Grid	0.179 ± 0.078	0.312 ± 0.172	0.329 ± 0.096	0.559 ± 0.205	0.474 ± 0.104	0.58 ± 0.063
	DEMV	**0.102 ± 0.063**	**0.232 ± 0.145**	**0.197 ± 0.087**	**0.678 ± 0.214**	0.525 ± 0.079	**0.677 ± 0.081**

Table 17
Evaluation results for all multi-class datasets and methods using three sensitive variables.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	No one	0.148 ± 0.039	0.283 ± 0.141	0.305 ± 0.143	0.353 ± 0.12	**0.497 ± 0.042**	0.54 ± 0.095
	EG	0.134 ± 0.047	0.27 ± 0.108	**0.346 ± 0.114**	0.427 ± 0.141	0.489 ± 0.038	0.574 ± 0.073
	Grid	0.065 ± 0.057	0.237 ± 0.116	0.277 ± 0.196	0.854 ± 0.128	0.432 ± 0.043	**0.673 ± 0.066**
	DEMV	**0.031 ± 0.019**	0.326 ± 0.235	0.274 ± 0.122	0.695 ± 0.189	0.489 ± 0.036	0.656 ± 0.112
Crime	No one	0.267 ± 0.066	0.435 ± 0.152	0.371 ± 0.097	0.176 ± 0.161	**0.504 ± 0.035**	0.336 ± 0.203
	EG	0.258 ± 0.072	0.459 ± 0.215	0.413 ± 0.128	0.171 ± 0.17	0.493 ± 0.04	0.307 ± 0.242
	Grid	**0.141 ± 0.058**	0.539 ± 0.229	0.381 ± 0.074	0.159 ± 0.178	0.309 ± 0.028	0.218 ± 0.233
	DEMV	0.149 ± 0.074	0.291 ± 0.133	**0.349 ± 0.048**	0.498 ± 0.226	0.437 ± 0.03	**0.571 ± 0.096**
Drug	No one	0.299 ± 0.045	0.293 ± 0.152	0.331 ± 0.099	0.144 ± 0.086	0.67 ± 0.029	0.36 ± 0.142
	EG	0.286 ± 0.056	0.236 ± 0.124	0.366 ± 0.067	0.172 ± 0.054	**0.671 ± 0.042**	0.419 ± 0.065
	Grid	0.207 ± 0.038	0.278 ± 0.14	**0.295 ± 0.092**	0.338 ± 0.155	0.64 ± 0.025	0.546 ± 0.147
	DEMV	**0.142 ± 0.055**	0.178 ± 0.068	0.362 ± 0.093	0.504 ± 0.169	0.66 ± 0.033	**0.659 ± 0.072**
Law	No one	0.2 ± 0.027	0.201 ± 0.028	0.165 ± 0.03	0.502 ± 0.071	**0.52 ± 0.01**	0.655 ± 0.032
	EG	0.225 ± 0.03	0.238 ± 0.032	0.172 ± 0.031	0.457 ± 0.072	0.517 ± 0.012	0.628 ± 0.034
	Grid	0.278 ± 0.091	0.359 ± 0.116	0.189 ± 0.028	0.408 ± 0.189	0.505 ± 0.016	0.566 ± 0.089
	DEMV	**0.042 ± 0.029**	0.144 ± 0.064	**0.159 ± 0.022**	0.885 ± 0.078	0.512 ± 0.011	**0.769 ± 0.019**
Wine	No one	0.434 ± 0.049	1.513 ± 0.308	0.163 ± 0.07	0.448 ± 0.049	**0.546 ± 0.019**	0.538 ± 0.063
	EG	0.419 ± 0.049	1.453 ± 0.294	0.169 ± 0.07	0.463 ± 0.051	0.541 ± 0.018	0.524 ± 0.071
	Grid	**0.057 ± 0.043**	0.101 ± 0.045	0.429 ± 0.063	0.76 ± 0.125	0.398 ± 0.022	**0.642 ± 0.041**
	DEMV	0.097 ± 0.04	0.593 ± 0.287	**0.109 ± 0.048**	0.877 ± 0.051	0.508 ± 0.02	0.577 ± 0.192
Mean	No one	0.27 ± 0.109	0.545 ± 0.548	0.267 ± 0.097	0.325 ± 0.16	**0.547 ± 0.071**	0.486 ± 0.135
	EG	0.264 ± 0.104	0.531 ± 0.524	0.293 ± 0.115	0.338 ± 0.153	0.542 ± 0.075	0.49 ± 0.128
	Grid	0.15 ± 0.094	0.303 ± 0.162	0.314 ± 0.094	0.504 ± 0.293	0.457 ± 0.124	0.529 ± 0.182
	DEMV	**0.092 ± 0.055**	0.306 ± 0.177	**0.251 ± 0.113**	0.692 ± 0.19	0.521 ± 0.083	**0.646 ± 0.08**

Finally, Tables 12, 13, and 14 reports the detailed results for each dataset in the experiments involving, respectively Gradient Boosting, SVM and Neural Network. For each dataset, we highlight in boldface the best value of each metric whose differences are statistically significant.

Table 18
Evaluation results for multi-class datasets using Gradient Boosting classifier.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	No one	0.09 ± 0.053	0.178 ± 0.107	0.279 ± 0.127	0.656 ± 0.177	0.557 ± 0.04	0.696 ± 0.062
	EG	0.095 ± 0.068	**0.183 ± 0.119**	0.309 ± 0.153	0.658 ± 0.221	0.546 ± 0.039	0.685 ± 0.086
	Grid	0.065 ± 0.043	0.194 ± 0.091	0.195 ± 0.077	**0.742 ± 0.138**	0.443 ± 0.042	0.693 ± 0.047
	DEMV	**0.056 ± 0.04**	0.192 ± 0.139	0.272 ± 0.146	0.74 ± 0.17	**0.559 ± 0.042**	**0.716 ± 0.061**
Law	No one	0.232 ± 0.03	0.221 ± 0.035	0.175 ± 0.025	0.405 ± 0.082	**0.536 ± 0.01**	0.61 ± 0.045
	EG	**0.071 ± 0.053**	0.167 ± 0.072	0.154 ± 0.026	**0.809 ± 0.142**	0.527 ± 0.008	**0.754 ± 0.039**
	Grid	0.322 ± 0.071	0.433 ± 0.049	0.161 ± 0.025	0.344 ± 0.157	0.512 ± 0.01	0.522 ± 0.063
	DEMV	0.091 ± 0.029	**0.15 ± 0.065**	0.156 ± 0.02	0.739 ± 0.088	0.526 ± 0.008	0.742 ± 0.025
Mean	No one	0.161 ± 0.1	0.2 ± 0.03	0.227 ± 0.074	0.53 ± 0.177	**0.546 ± 0.015**	0.653 ± 0.061
	EG	0.083 ± 0.017	0.175 ± 0.011	0.231 ± 0.11	0.734 ± 0.107	0.536 ± 0.013	0.72 ± 0.049
	Grid	0.194 ± 0.182	0.314 ± 0.169	0.178 ± 0.024	0.543 ± 0.281	0.478 ± 0.049	0.607 ± 0.121
	DEMV	**0.074 ± 0.025**	**0.171 ± 0.03**	0.214 ± 0.082	**0.74 ± 0.001**	0.542 ± 0.023	**0.729 ± 0.018**

Table 19
Evaluation results for multi-class datasets using Support Vector Machines classifier.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	No one	0.105 ± 0.046	0.174 ± 0.119	0.321 ± 0.18	0.574 ± 0.17	0.543 ± 0.046	0.652 ± 0.077
	EG	0.109 ± 0.044	**0.16 ± 0.071**	0.337 ± 0.158	0.549 ± 0.142	**0.546 ± 0.045**	0.652 ± 0.067
	Grid	0.197 ± 0.068	0.273 ± 0.083	0.295 ± 0.191	0.197 ± 0.22	0.435 ± 0.045	0.302 ± 0.261
	DEMV	**0.047 ± 0.03**	0.218 ± 0.164	0.279 ± 0.128	**0.73 ± 0.153**	**0.546 ± 0.042**	**0.707 ± 0.062**
Law	No one	0.267 ± 0.022	0.241 ± 0.019	0.173 ± 0.031	0.311 ± 0.048	**0.533 ± 0.011**	0.554 ± 0.035
	EG	0.234 ± 0.022	0.207 ± 0.04	0.192 ± 0.03	0.343 ± 0.063	0.525 ± 0.01	0.575 ± 0.044
	Grid	0.375 ± 0.024	0.492 ± 0.039	0.159 ± 0.019	0.277 ± 0.04	0.511 ± 0.013	0.482 ± 0.033
	DEMV	**0.116 ± 0.034**	**0.134 ± 0.042**	0.161 ± 0.02	**0.67 ± 0.099**	0.523 ± 0.01	**0.724 ± 0.031**
Mean	No one	0.186 ± 0.115	0.208 ± 0.047	0.247 ± 0.105	0.442 ± 0.186	**0.538 ± 0.007**	0.603 ± 0.069
	EG	0.172 ± 0.088	0.184 ± 0.033	0.264 ± 0.103	0.446 ± 0.146	0.536 ± 0.015	0.613 ± 0.054
	Grid	0.286 ± 0.126	0.382 ± 0.155	0.227 ± 0.096	0.237 ± 0.057	0.473 ± 0.054	0.392 ± 0.127
	DEMV	**0.082 ± 0.049**	**0.176 ± 0.059**	0.22 ± 0.083	**0.7 ± 0.042**	0.534 ± 0.016	**0.716 ± 0.012**

Table 20
Evaluation results for multi-class datasets using Neural Network classifier.

Data	Method	SP	EO	ZO Loss	DI	Acc	H-Mean
CMC	No one	0.081 ± 0.087	0.149 ± 0.104	0.338 ± 0.195	0.702 ± 0.261	0.542 ± 0.053	0.683 ± 0.098
	EG	Not applicable					
	Grid	Not applicable					
	DEMV	**0.06 ± 0.059**	0.17 ± 0.111	0.293 ± 0.135	**0.756 ± 0.171**	**0.544 ± 0.048**	**0.717 ± 0.071**
Law	No one	0.218 ± 0.04	0.197 ± 0.039	0.168 ± 0.03	0.436 ± 0.085	**0.531 ± 0.01**	0.629 ± 0.047
	EG	Not applicable					
	Grid	Not applicable					
	DEMV	**0.096 ± 0.044**	0.138 ± 0.06	0.125 ± 0.032	**0.721 ± 0.129**	0.519 ± 0.01	**0.739 ± 0.04**
Mean	No one	0.15 ± 0.097	0.173 ± 0.034	0.253 ± 0.12	0.569 ± 0.188	**0.536 ± 0.008**	0.656 ± 0.038
	EG	Not applicable					
	Grid	Not applicable					
	DEMV	**0.078 ± 0.025**	0.154 ± 0.023	0.209 ± 0.119	**0.738 ± 0.025**	0.532 ± 0.018	**0.728 ± 0.016**

Appendix C. Detailed results for multi-class classification

In the following we report the tables describing the detailed results of experiments involving multi-class datasets. For each dataset and for each method, we report the mean and standard deviation of all metrics. In addition, we report the mean and standard deviation of the H-Mean computed from the obtained values. Finally, we also report the overall means and standard deviations of all the values obtained by each method in each experiment. We split the results among experiments involving one, two, and three sensitive variables and experiments with more complex classifiers. For each dataset, we highlight in boldface the best value of each metric whose differences are statistically significant.

In particular, Table 15 reports the results of experiments involving one sensitive variable, Table 16 reports the results of experiments with two sensitive variables, and Table 17 shows the results of experiments with three sensitive variables.

Finally, Tables 18, 19, and 20 reports the detailed results for each dataset of the experiments involving respectively Gradient Boosting, SVM, and Neural Networks.

Table 21
ANOVA tables for binary datasets.
(a) One sensitive variable

	DF	SS	MS	F	p-value
Statistical Parity					
C(method)	4.0	153.413	38.353	40.894	0.0
Residual	2405.0	2255.587	0.938		
Equalized Odds					
C(method)	4.0	57.828	14.457	12.001	0.0
Residual	670.0	807.154	1.205		
Zero–one Loss					
C(method)	4.0	2.999	0.75	0.749	0.558
Residual	2405.0	2406.001	1.00		
Disparate Impact					
C(method)	4.0	108.786	27.197	28.436	0.0
Residual	2405.0	2300.214	0.956		
Accuracy					
C(method)	4.0	0.453	0.113	0.113	0.978
Residual	2405.0	2408.547	1.001		
H-Mean					
C(method)	3.0	1.378	0.459	24.349	0.0
Residual	366.0	6.906	0.019		

(b) Two sensitive variables

	DF	SS	MS	F	p-value
Statistical Parity					
C(method)	3.0	42.582	14.194	25.563	0.0
Residual	89.0	49.418	0.555		
Equalized Odds					
C(method)	3.0	8.884	2.961	3.365	0.026
Residual	49.0	43.116	0.880		
Zero One Loss					
C(method)	3.0	24.807	8.269	10.953	0.0
Residual	89.0	67.193	0.755		
Disparate Impact					
C(method)	3.0	44.572	14.857	27.881	0.0
Residual	89.0	47.428	0.533		
Accuracy					
C(method)	3.0	14.831	4.944	5.702	0.001
Residual	89.0	77.169	0.867		
H-Mean					
C(method)	3.0	6.423	2.141	77.032	0.0
Residual	276.0	7.671	0.028		

(c) Three sensitive variables

	DF	SS	MS	F	p-value
Statistical Parity					
C(method)	3.0	9.643	3.214	3.474	0.019
Residual	89.0	82.357	0.925		
Equalized Odds					
C(method)	3.0	19.087	6.362	4.388	0.01
Residual	38.0	55.102	1.450		
Zero One Loss					
C(method)	3.0	0.432	0.144	0.14	0.936
Residual	89.0	91.568	1.029		
Disparate Impact					
C(method)	3.0	1.196	0.399	0.391	0.76
Residual	89.0	90.804	1.020		

(*continued on next page*)

Table 21 (*continued*).

Accuracy					
C(method)	3.0	7.084	2.361	2.475	0.067
Residual	89.0	84.916	0.954		
H-Mean					
C(method)	3.0	1.038	0.346	12.147	0.0
Residual	276.0	7.858	0.028		

Table 22
ANOVA tables for multi-class datasets.
(a) One sensitive variable

	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	4.0	7.402	1.850	1.86	0.016
Residual	651.0	647.598	0.995		
Equalized Odds					
C(method)	4.0	13.725	3.431	1.326	0.262
Residual	184.0	476.038	2.587		
Zero–one Loss					
C(method)	4.0	50.71	12.678	13.657	0.0
Residual	651.0	604.29	0.928		
Disparate Impact					
C(method)	4.0	19.447	4.862	4.98	0.001
Residual	651.0	635.553	0.976		
Accuracy					
C(method)	4.0	4.338	1.084	1.085	0.363
Residual	651.0	650.662	0.999		
H-Mean					
C(method)	3.0	0.628	0.209	7.547	0.0
Residual	926.0	25.670	0.028		
(b) Two sensitive variables					
	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	4.0	104.788	26.197	39.255	0.0
Residual	303.0	202.212	0.667		
Equalized Odds					
C(method)	4.0	19.262	4.816	3.127	0.018
Residual	112.0	172.494	1.540		
Zero One Loss					
C(method)	4.0	29.399	7.350	8.022	0.0
Residual	303.0	277.601	0.916		
Disparate Impact					
C(method)	4.0	18.98	4.745	4.992	0.001
Residual	303.0	288.02	0.951		

(*continued on next page*)

Appendix D. ANOVA tables

In the following, we report the ANOVA tables of our experiments. In particular, Table 21 shows the results for binary, and Table 22 reports the results for multi-class experiments involving sensitive groups identified by a different number of sensitive variables. Tables 23 and 24 reports instead the results of the ANOVA tests involving more complex classifiers for respectively binary and multi-class classification. We recall that, in order to be statistically significant the probability value (*p-value*) most be lower than 0.05. In this case, the test rejects the null hypothesis of equal mean for all groups.

Table 22 (*continued*).

Accuracy					
C(method)	4.0	17.356	4.339	4.539	0.001
Residual	303.0	289.644	0.956		
H-Mean					
C(method)	3.0	1.243	0.414	16.59	0.0
Residual	686.0	17.130	0.025		
(c) Three sensitive variables					
	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	3.0	7.6	2.533	2.582	0.054
Residual	243.0	238.4	0.981		
Equalized Odds					
C(method)	3.0	9.151	3.050	1.333	0.27
Residual	73.0	167.103	2.289		
Zero One Loss					
C(method)	3.0	24.773	8.258	9.07	0.0
Residual	243.0	221.227	0.910		
Disparate Impact					
C(method)	3.0	7.054	2.351	2.391	0.069
Residual	243.0	238.946	0.983		
Accuracy					
C(method)	3.0	21.399	7.133	7.717	0.0
Residual	243.0	224.601	0.924		
H-Mean					
C(method)	3.0	0.572	0.191	6.921	0.0
Residual	586.0	16.132	0.028		

Table 23
ANOVA tables of binary experiments with other classifiers.
(a) Gradient Boosting

	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	3.0	46.935	15.645	16.768	0.0
Residual	656.0	612.065	0.933		
Equalized Odds					
C(method)	3.0	10.709	3.570	3.612	0.013
Residual	656.0	648.291	0.988		
Zero–one Loss					
C(method)	3.0	3074.488	1024.829	2.16	0.092
Residual	385.0	182633.494	474.373		
Disparate Impact					
C(method)	3.0	77.001	25.667	28.931	0.0
Residual	656.0	581.999	0.887		
Accuracy					
C(method)	3.0	0.026	0.009	0.009	0.999
Residual	656.0	658.974	1.005		
H-Mean					
C(method)	3.0	0.385	0.128	46.927	0.0
Residual	656.0	1.793	0.003		
(b) Support Vector Machines					
	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	3.0	167.808	55.936	74.704	0.0
Residual	656.0	491.192	0.749		

(*continued on next page*)

Table 23 (*continued*).

Equalized Odds					
C(method)	3.0	6.486	2.162	2.173	0.09
Residual	656.0	652.514	0.995		
Zero One Loss					
C(method)	3.0	0.008	0.003	1.523	0.207
Residual	656.0	1.159	0.002		
Disparate Impact					
C(method)	3.0	69.386	23.129	25.733	0.0
Residual	656.0	589.614	0.899		
Accuracy					
C(method)	3.0	0.404	0.135	0.134	0.94
Residual	656.0	658.596	1.004		
H-Mean					
C(method)	1.0	0.101	0.101	28.775	0.0
Residual	618.0	2.164	0.004		
(c) Neural Networks					
	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	1.0	42.363	42.363	45.402	0.0
Residual	618.0	576.637	0.933		
Equalized Odds					
C(method)	1.0	0.387	0.387	0.386	0.534
Residual	618.0	618.613	1.001		
Zero One Loss					
C(method)	1.0	349.329	349.329	0.826	0.364
Residual	373.0	157719.633	422.841		
Disparate Impact					
C(method)	1.0	7.721	7.721	7.806	0.005
Residual	618.0	611.279	0.989		
Accuracy					
C(method)	1.0	0.02	0.020	0.02	0.886
Residual	618.0	618.98	1.002		
H-Mean					
C(method)	1.0	0.101	0.101	28.775	0.0
Residual	618.0	2.164	0.004		

Table 24
ANOVA tables of multi-class experiments with other classifiers.
(a) Gradient Boosting

	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	3.0	141.392	47.131	59.732	0.0
Residual	656.0	517.608	0.789		
Equalized Odds					
C(method)	3.0	31.746	10.582	11.067	0.0
Residual	656.0	627.254	0.956		
Zero–one Loss					
C(method)	3.0	164.678	54.893	0.429	0.732
Residual	203.0	25949.535	127.830		
Disparate Impact					
C(method)	3.0	67.483	22.494	24.947	0.0
Residual	656.0	591.517	0.902		

(*continued on next page*)

Table 24 (*continued*).

Accuracy					
C(method)	3.0	62.187	20.729	22.785	0.0
Residual	656.0	596.813	0.910		
H-Mean					
C(method)	3.0	0.385	0.128	46.927	0.0
Residual	656.0	1.793	0.003		
(b) Support Vector Machines					
	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	3.0	1.119	0.373	137.255	0.0
Residual	656.0	1.783	0.003		
Equalized Odds					
C(method)	3.0	0.839	0.280	17.93	0.0
Residual	656.0	10.227	0.016		
Zero One Loss					
C(method)	3.0	0.051	0.017	1.346	0.258
Residual	656.0	8.290	0.013		
Disparate Impact					
C(method)	3.0	6.316	2.105	115.033	0.0
Residual	656.0	12.005	0.018		
Accuracy					
C(method)	3.0	0.074	0.025	22.195	0.0
Residual	656.0	0.731	0.001		
H-Mean					
C(method)	3.0	2.383	0.794	211.341	0.0
Residual	656.0	2.465	0.004		
(c) Neural Networks					
	DF	**SS**	**MS**	**F**	**p-value**
Statistical Parity					
C(method)	1.0	0.098	0.098	30.711	0.0
Residual	618.0	1.975	0.003		
Equalized Odds					
C(method)	1.0	0.007	0.007	0.892	0.345
Residual	618.0	5.064	0.008		
Zero One Loss					
C(method)	1.0	0.038	0.038	2.216	0.137
Residual	618.0	10.539	0.017		
Disparate Impact					
C(method)	1.0	0.556	0.556	23.01	0.0
Residual	618.0	14.938	0.024		
Accuracy					
C(method)	1.0	0.001	0.001	0.403	0.526
Residual	618.0	0.844	0.001		
H-Mean					
C(method)	1.0	0.101	0.101	28.775	0.0
Residual	618.0	2.164	0.004		

References

Agarwal, A., Beygelzimer, A., Dudik, M., Langford, J., & Wallach, H. (2018). A reductions approach to fair classification. In J. Dy, & A. Krause (Eds.), *Proceedings of machine learning research*: *vol. 80*, *Proceedings of the 35th international conference on machine learning* (pp. 60–69). PMLR, URL: https://proceedings.mlr.press/v80/agarwal18a.html.

AI fairness 360 - Resources. (2018). URL: https://aif360.mybluemix.net/resources#guidance.

Amigó, E., Deldjoo, Y., Mizzaro, S., & Bellogín, A. (2023). A unifying and general account of fairness measurement in recommender systems. *Information Processing & Management*, *60*(1), Article 103115.

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias. *ProPublica*, *23*(2016), 139–159.

Austin, K. A., Christopher, C. M., & Dickerson, D. (2016). Will I pass the bar exam: Predicting student success using LSAT scores and law school performance. *Hofstra Law Review*, *45*, 753.

Baeza-Yates, R. (2018). Bias on the web. *Communications of the ACM*, *61*(6), 54–61.

Baskota, A., & Ng, Y.-K. (2018). A graduate school recommendation system using the multi-class support vector machine and KNN approaches. In *2018 IEEE international conference on information reuse and integration* (pp. 277–284). IEEE.

Bird, S., Dudík, M., Edgar, R., Horn, B., Lutz, R., Milan, V., et al. (2020). *Fairlearn: A toolkit for assessing and improving fairness in AI*: *Technical report MSR-TR-2020-32*, Microsoft, URL: https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/.

Boratto, L., & Marras, M. (2021). Advances in bias-aware recommendation on the web. In *Proceedings of the 14th ACM international conference on web search and data mining* (pp. 1147–1149). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3437963.3441665.

Busenbark, J. R., Yoon, H., Gamache, D. L., & Withers, M. C. (2022). Omitted variable bias: Examining management research with the impact threshold of a confounding variable (ITCV). *Journal of Management*, *48*(1), 17–48.

Calders, T., Karim, A., Kamiran, F., Ali, W., & Zhang, X. (2013). Controlling attribute effect in linear regression. In *2013 IEEE 13th international conference on data mining* (pp. 71–80). [ISSN: 2374-8486] http://dx.doi.org/10.1109/ICDM.2013.114.

Caton, S., & Haas, C. (2020). Fairness in machine learning: A survey. arXiv:2010.04053 [cs, stat].

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Clarke, K. A. (2005). The phantom menace: Omitted variable bias in econometric research. *Conflict Management and Peace Science*, *22*(4), 341–352.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, *47*(4), 547–553.

d'Alessandro, B., O'Neil, C., & LaGatta, T. (2017). Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big Data*, *5*(2), 120–134.

d'Aloisio, G., Stilo, G., Di Marco, A., & D'Angelo, A. (2022). Enhancing fairness in classification tasks with multiple variables: A data- and model-agnostic approach. In *Proceedings of third international workshop on algorithmic bias in search and recommendation* (to be published).

Denis, C., Elie, R., Hebiri, M., & Hu, F. (2021). Fairness guarantee in multi-class classification. arXiv:2109.13642 [math, stat].

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*(2), 103–130.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference* (pp. 214–226). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2090236.2090255.

Fairlearn (2022). Fairlearn documentation. URL: https://fairlearn.org/main/faq.html.

Fehrman, E., Muhammad, A. K., Mirkes, E. M., Egan, V., & Gorban, A. N. (2017). The five factor model of personality and evaluation of drug consumption risk. In F. Palumbo, A. Montanari, & M. Vichi (Eds.), *Data science, studies in classification, data analysis, and knowledge organization* (pp. 231–242). Cham: Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-55723-6_18.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 259–268). Sydney NSW Australia: ACM, http://dx.doi.org/10.1145/2783258.2783311.

Ferger, W. F. (1931). The nature and use of the harmonic mean. *Journal of the American Statistical Association*, *26*(173), 36–40.

Friedler, S. A., Scheidegger, C., & Venkatasubramanian, S. (2016). On the (im) possibility of fairness. arXiv preprint arXiv:1609.07236.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Hagan, M. T., Demuth, H. B., & Beale, M. (1997). *Neural network design*. PWS Publishing Co.

Hajian, S., Bonchi, F., & Castillo, C. (2016). Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2125–2126). ACM.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, *29*, 3315–3323.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE world congress on computational intelligence*, *2008 IEEE international joint conference on neural networks* (pp. 1322–1328). IEEE.

Jiang, C., Liu, Y., Ding, Y., Liang, K., & Duan, R. (2017). Capturing helpful reviews from social media for product quality improvement: A multi-class classification approach. *International Journal of Production Research*, *55*(12), 3528–3541.

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, *33*(1), 1–33.

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, *132*(1), 1–63.

Kohavi, R., et al. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd, vol. 96* (pp. 202–207).

Krstinić, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). Multi-label classifier performance evaluation with confusion matrix. *Computer Science Information Technology*, *10*, 1–14.

Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. In *Advances in neural information processing systems, vol. 30*. Curran Associates, Inc.

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40*(3), 203–228.

McDonald, J. H. (2009). One-way ANOVA. In *Handbook of biological statistics, vol. 2*. MD: Sparky House Publishing Baltimore.

Meenachi, L., Ramakrishnan, S., Sivaprakash, M., Thangaraj, C., & Sethupathy, S. (2022). Multi class ensemble classification for crop recommendation. In *2022 International conference on inventive computation technologies* (pp. 1319–1324). [ISSN: 2767-7788] http://dx.doi.org/10.1109/ICICT54344.2022.9850561.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*(6), 1–35.

Menard, S. (2002). *Applied logistic regression analysis, vol. 106*. Sage.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, *24*(12), 1565–1567.

Olteanu, A., Castillo, C., Diaz, F., & Kıcıman, E. (2019). Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, *2*, 13.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché Buc, F., et al. (2021). Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program. *Journal of Machine Learning Research*, *22*.

Putzel, P., & Lee, S. (2022). Blackbox post-processing for multiclass fairness. arXiv:2201.04461 [cs].

Radovanović, S., Petrović, A., Delibašić, B., & Suknović, M. (2021). A fair classifier chain for multi-label bank marketing strategy classification. *International Transactions in Operational Research*, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.13059.

Ratanamahatana, C. A., & Gunopulos, D. (2002). Scaling up the naive Bayesian classifier: Using decision trees for feature selection.

Redmond, M., & Baveja, A. (2002). A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, *141*(3), 660–678.

Refaeilzadeh, P., Tang, L., & Liu, H. (2016). Cross-validation. In *Encyclopedia of database systems* (pp. 1–7). New York, NY: Springer New York, http://dx.doi.org/10.1007/978-1-4899-7993-3_565-2.

Rosenfield, G., & Fitzpatrick-Lins, K. (1986). A coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering and Remote Sensing*, *52*(2), 223–227.

Stitini, O., Kaloun, S., & Bencharef, O. (2022). Integrating contextual information into multi-class classification to improve the context-aware recommendation. *Procedia Computer Science*, *198*, 311–316.

Street, W. N., Wolberg, W. H., & Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization, vol. 1905* (pp. 861–870). International Society for Optics and Photonics.

Suchithra, M. S., & Pai, M. L. (2018). Improving the performance of Sigmoid Kernels in multiclass SVM using optimization techniques for agricultural fertilizer recommendation system. In I. Zelinka, R. Senkerik, G. Panda, & P. S. Lekshmi Kanthan (Eds.), *Communications in computer and information science*, *Soft computing systems* (pp. 857–868). Singapore: Springer, http://dx.doi.org/10.1007/978-981-13-1936-5_87.

Suresh, H., & Guttag, J. V. (2019). A framework for understanding unintended consequences of machine learning, *2* (p. 8). arXiv preprint arXiv:1901.10002.

Tsanas, A., Little, M., McSharry, P., & Ramig, L. (2009). Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests. *Nature Precedings*, 1.

Verma, S., & Rubin, J. (2018). Fairness definitions explained. In *2018 IEEE/ACM international workshop on software fairness* (pp. 1–7). IEEE.

Wolpert, D. H. (1999). What does dinner cost? (p. 34). URL: http://www.no-free-lunch.org/coev.pdf.

Yanes, N., Mostafa, A. M., Ezz, M., & Almuayqil, S. N. (2020). A machine learning-based recommender system for improving students learning experiences. *IEEE Access*, *8*, 201218–201235.

Zhang, J., Cao, P., Gross, D. P., & Zaiane, O. R. (2013). On the application of multi-class classification in physical therapy recommendation. *Health Information Science and Systems*, *1*(1), 15. http://dx.doi.org/10.1186/2047-2501-1-15.

Giordano d'Aloisio is a Ph.D. student on an institutional fellowship in Information and Communication Technology at the University of L'Aquila, where he also holds a Master's degree in Computer Science. His research activity mainly focuses on Data Science and Software Engineering techniques for the quality assurance of machine learning systems, in particular Bias and Fairness. He is responsible for the Data Accumulation task of the Territori Aperti project (funded by Fondo territori Lavoro Conoscenza - CGIL CISL UIL), and he is a member of Emeliot, FAIR-EDU, and PinKamP projects. He has been a co-reviewer of different conferences and journals, including IPM, FASE, ICPE, and ICDM. He has been a member of the 2022 and 2023 editions of the BIAS workshop's program committee.

Andrea D'Angelo is a Ph.D. student on an institutional fellowship at the University of L'Aquila, where he previously earned his Bachelor's and Master's degrees in Computer Science. He formerly held a research scholarship of the Territori Aperti project (funded by Fondo territori Lavoro Conoscenza - CGIL CISL UIL), in which he is still currently involved. He is part of the program committee of the BIAS 2023 workshop. His research interests revolve around Information Retrieval, Formal Language Theory and Machine Learning models for Natural Language Processing, such as Transformer Neural Networks.

Antinisca Di Marco is associate professor in Computer Science at University of L'Aquila. Her main research topics are Software Quality Engineering, Quality (such as fairness, privacy and explainability) Engineering in Learning Systems, Data Science, and Bioinformatics. She is involved in several national and international projects on such topics. She is the responsible of the research infrastructure of Territori Aperti project (funded by Fondo territori Lavoro Conoscenza - CGIL CISL UIL), co-PI of SoBigData.it (a project funded by European Union - NextGenerationEU - National Recovery and Resilience Plan - Piano Nazionale di Ripresa e Resilienza - PNRR), and the director of the INFOLIFE CINI Laboratory node in L'Aquila. Since 2018, she is involved in serveral actions and projects aiming at improving equal opportunities in STEM (Science, Technology, Engineering and Mathematics). In particolar, she is member of the cost action EUGAIN (https://eugain.eu/), and co-ideator and co-coordinator of PinKamP (www.pinkamp.disim.univaq.it).

Giovanni Stilo is a Computer Science and Data Science associate professor at the University of L'Aquila. He received his Ph.D. in Computer Science in 2013, and in 2014 he was a visiting researcher at Yahoo! Labs in Barcelona. His research interests are machine learning, data mining, and artificial intelligence, with a special interest in (but not limited to) trustworthiness aspects such as Bias, Fairness, and Explainability. He has co-organized a long series (2020–2023) of top-tier International workshops and Journal Special Issues focused on Bias and Fairness in Search and Recommendation. He serves on the editorial boards of IEEE, ACM, Springer, and Elsevier Journals such as TITS, TKDE, DMKD, AI, KAIS, and AIIM. He is responsible for New technologies for data collection, preparation, and analysis of the Territori Aperti project (funded by Fondo territori Lavoro Conoscenza - CGIL CISL UIL) and coordinator of the activities on "Responsible Data Science and Training" of SoBigData.it (a project funded by European Union - NextGenerationEU - National Recovery and Resilience Plan - Piano Nazionale di Ripresa e Resilienza - PNRR), Head of the Master's Degree course in Applied Data Science at the University of L'Aquila, and PI of the "FAIR-EDU: Promote FAIRness in EDUcation Institutions" project.