Journal of the Brazilian Computer Society, 2024, 30:1, doi: 10.5753/jbcs.2024.3468

LAGOON: Achieving bounded individual fairness through classification frequency equalization

Maria Silva [**Universidade Federal do Ceará** | ***malu.maia@lsbd.ufc.br***]
Iago Chaves [**Universidade Federal do Ceará** | ***iago.chaves@lsbd.ufc.br***]
Javam Machado [**Universidade Federal do Ceará** | ***javam.machado@lsbd.ufc.br***]

Computer Science Department, Universidade Federal do Ceará, Campus do Pici - Bloco 910, Fortaleza, CE, 60020-181, Brazil.

Received: 19 June 2023 • **Accepted:** 18 April 2024 • **Published:** 27 August 2024

Abstract One of the main concerns about using machine learning models for classification is algorithmic discrimination. Several works define different meanings of fairness to avoid or mitigate unfair classifications against minorities. The achievement of algorithmic fairness implies modifying training data, model operation, or outputs. Hence, the fair algorithm may modify the original classification. Generally, fairness means not discriminating against a person or a group. In a utopia, a system would classify every person or minority as privileged, which may decrease the utility of classification. We define λ-fairness, a relaxation of individual fairness designed to achieve fairness while maintaining utility with configurable parameters. We also propose a post-processing method that uses frequency equalization to achieve fairness in machine learning models by generalizing the outputs into frequencies. We used this flexible approach on LAGOON, an algorithm that achieves λ-fairness using frequency equalization. For experiments, we employ three benchmarks with different contexts to evaluate the quality of our approach. We compared our results to two baselines that aim to achieve fairness and minimize utility loss.

Keywords: Fairness, Classification, Utility

1 Introduction

The increasing use of machine learning classifiers in several application domains shows its adaptability and usefulness for different tasks. Classification tools are broadly used in companies to save or optimize resources, such as time wasted by humans to predict classes and quality in prediction [Ramentol *et al.*, 2021]. Classification models can benefit companies and society by solving real-world problems, e.g., credit card fraud detection [Shen *et al.*, 2007], image classification [Zhang *et al.*, 2019], college application acceptance [Mashat *et al.*, 2012], and document classification [Khan *et al.*, 2010].

Credit approval is an example of how companies use technology to make decisions. Improving credit decisions using algorithms to decide who can get a loan is already a reality [Dash, 2021; Argawal, 2021]. Nevertheless, models may discriminate against people on classification [Pappada and Pauli, 2022]. Disadvantageous treatment of a person may occur for some reason in automated decision systems, like bias in training datasets, discriminatory classifications, or missing data. The mistreatment occurs when a potentially discriminatory (called *protected*) attribute [Law, 2016] affects the classification, harming individuals.

Different researchers such as Dwork *et al.*; Zliobaite; Luong *et al.*; Pedreshi *et al.*; Kusner *et al.* have discussed different perceptions of algorithmic fairness. There are three main understandings of fairness: group, individual, and counterfactual fairness. Group fairness aims to ensure that different protected groups have the same possibility of receiving a positive or negative classification. Individual fairness, on the other hand, focuses on equitable treatment for similar individuals. According to [Kusner *et al.*, 2017], individuals achieve counterfactual fairness if a fairness algorithm produces the same outcome in both the “real” world and a hypothetical world where the individual has a different protected value. In this work, we’ll concentrate on individual fairness.

An example that illustrates individual fairness in society is as follows. Let us picture Person 1 and Person 2 buying a concert ticket on an automatic place allocation system. They both pay the same amount. However, on the concert day, Person 1 gets a broken chair to watch the show. In contrast, Person 2 gets a comfortable chair. It is unfair to Person 1 to get an uncomfortable place to see the concert, given that they have paid the same price.

Now, consider a loan scenario where a bank categorizes loan applicants as either “Good” or “Bad” payers based on various factors such as their income, credit payment history, address, and other relevant attributes. Although the dataset may not include any protected attributes, there is a possibility that some features may be correlated with them. For instance, demographic research can reveal whether there is a strong correlation between a neighborhood and a particular race or gender [Galhotra *et al.*, 2021].

Suppressing the attributes correlated to protected ones may cause a significant loss of information, impacting the quality of prediction and harming decision-making and accuracy. In addition, the suppression of protected attributes and correlations can also lead the model to further bias, conducting to some discrimination degree [Calders and Verwer, 2010]. One way to combat discrimination is adding fairness constraints to the algorithm in order to identify and mitigate

Table 1. Running example. Highlighted rows represent incorrect predictions.

#	*Race* (−)	Zipcode	Credits	Delays	Predicted
1	*White*	104	2	0	Good
2	*African-American*	101	1	0	Good
3	*Asian American*	103	0	0	Good
4	*White*	101	3	2	Bad
5	***African-American***	**102**	**1**	**0**	**Bad**
6	*Hispanic*	105	0	0	Good
7	*Asian American*	104	0	0	Good
8	***White***	**104**	**3**	**2**	**Good**
9	***African-American***	**102**	**1**	**0**	**Bad**
10	*White*	103	1	0	Good

sources of bias [Zafar *et al.*, 2017].

Running Example. Table 1 displays a simplified dataset inspired by the German Credit Risk [Dua *et al.*, 2017], where *Race* is a protected attribute. It represents a ranking of bank loan applicants where the classification does not depend on the *Race* attribute, as it has been suppressed. However, there exists a correlation between *Race* and *Zipcode* [Census, Bureau, 2023], which may lead to discriminatory classifications. We have properly suppressed the protected attribute (*Race*) for prediction purposes but kept it in Table 1 to show the correlation between protected information and zip code. This correlation may impact the fairness in classification. We have denoted the suppressed attribute as *Attribute Name* $(-)$. Although individuals in rows 2 and 5 are similar when considering the credits and delays, their classifications differ because of zip code. Discrimination can occur by association, as there may be a correlation between the protected attribute and the zip code. The classifier would be individually fair if similar individuals had similar labels. For example, the classification in rows 2 and 5 should be the same, just as in rows 4 and 8. The model's utility on the example (considering the accuracy) is 0.7, and the rate of individuals that the model fairly classified is 0.4. This means that six individuals have questionable predictions when comparing them with their similar counterparts. Individuals 2, 5, 9, and 10 are similar but have different labels, just as 4 and 8.

Simply fixing unfair classifications might force us to label everyone the same way, which can harm the model's utility. We propose a more flexible fairness definition and a novel post-processing technique that adjusts the frequency distributions.

Our contributions are summarized as follows:

- We propose λ-fairness, a definition that bounds the minimum proportion of instances that satisfy individual fairness.
- We introduce a novel post-processing method for frequency equalization in classification models. It receives as input a pair of individuals and a mapping that assigns each instance to a distribution over the outcomes.
- We develop an adaptable approach called LAGOON that ensures λ-fairness by equalizing label frequencies.
- We propose a novel fair Decision Tree method that uses LAGOON to achieve the λ-fairness definition.
- We come up with a fair frequency equalization on Gradient Boosting that uses LAGOON to achieve the λ-fairness definition.
- We propose a fair clustering model, called Histogram-based Fair Cohort (HBFC), that uses LAGOON to achieve λ-fairness.
- We develop a fair Neural Network model that uses LAGOON to achieve λ-fairness.

This work performs the experiments in three benchmarks that address different contexts, measuring the utility of the fair proposed models. We use our definition to bind the minimum fairness rate on each fair model. The benchmark datasets are Adult Income [Dua *et al.*, 2017], German Credit [Dua *et al.*, 2017], and COMPAS [Larson *et al.*, 2016]. Each dataset addresses different applications to show that λ-fairness can be satisfied using LAGOON. We evaluate our approaches by measuring well-known metrics of utility and fairness.
The paper is organized as follows: Section 2 provides background information that is necessary to understand the proposed methods and their analysis. Section 3 explains the baselines. Section 4 defines our first contribution, including a practical analysis guide to define the best parameter according to the analyst's interest and notes from a social point of view. The second and third contributions, the frequency equalization and LAGOON, are described in Section 5. Section 6 shows how to use LAGOON in three different histogram-based classification models, and in Neural Network model to demonstrate the flexibility of our approach. The experiments and results are described and analyzed in Section 7 using two baselines for comparison. Finally, the paper concludes with a discussion on future work in Section 8.

2 Background

This section shows the background methods and definitions for theoretical and empirical understanding.

2.1 Fairness

In the fairness literature, the study subjects are individuals represented in databases. The features in the databases can be categorized into four sets:

- The explicit attribute(s);
- The protected attribute(s);
- The target attribute;
- The individuals' data for classification.

Name, ID, and telephone number can uniquely identify a person. However, the training set of machine learning models does not include those features, which are called explicit data.

A protected attribute is an individual's quality, trait, or characteristic that can not be discriminated against by law, such as gender identity, race, religious belief, or disabilities [UK Government, 2013]. The protected data is a set of features that can have one or more attributes.

Classification models are used to assign an individual's data to a specific class. The target attribute is the set of classes in the dataset or application. In the running example, the last column is the predicted attribute, while the two classes are "Good" and "Bad". Since there are only two classes, this is a binary classification. However, some datasets have more than two classes, which is commonly referred to as multiclass classification.

Finally, the individuals' data for classification are valuable features that do not include the explicit, protected, and target attributes.

Individual Fairness

The fulfillment of a fairness definition does not imply the accomplishment of all fairness notions [Chouldechova, 2017]. In this work, we tackle the individual fairness definition to ensure that similar people have similar outputs [Dwork *et al.*, 2012]. Individual fairness does not apply or solve affirmative action as in group fairness, but in both concepts, the target class may not depend on the protected attribute.

The concept of individual fairness ensures that similar individuals receive similar outcomes. In other words, if two people have similar attributes, they should be classified fairly and receive the same outcome. There are two important metrics to determine whether an individual's classification is fair: distance and dissimilarity [Dwork *et al.*, 2012]. Both metrics measure the difference between two objects, but we refer to distance as the difference between two sets of attributes. In contrast, we refer to dissimilarity as the difference between two distributions. The work described by Dwork *et al.* provides further insight into metrics with distance represented by d and dissimilarity as D. For instance, one can use Euclidean distance as a distance metric and Earthmover distance as a dissimilarity metric.

The distance between two individuals indicates the similarity between their skills or features. On the other hand, the dissimilarity between two individuals indicates how different they are regarding the opportunity of receiving specific outcomes. Mathematically, this refers to the distinction between two mappings. Let's consider a set of outcomes or classes A, individuals' data I, and probability distributions over the outcomes $\Delta(A)$. A mapping M associates an individual's data $x \in I$ to a probability density function $\mu_x \in \Delta(A)$. The probability density function reflects the likelihood of each outcome $a \in A$. To determine the dissimilarity between two individuals, we compare the mappings associated with each one of them.

The fairness constraint bounds the dissimilarity between the mappings of two different individuals by their distance. In other words, it guarantees that similar individuals have identical probability density functions. Hence, the model must not harm any person whose similar one has another classification.

The work proposed by Dwork *et al.* achieves a fair treatment between a pair of individuals by using the Definition 1 as a similarity constraint.

Definition 1 (Lipschitz Mapping) *A mapping $M : I \to \Delta(A)$, that assigns a set of individuals I to a set of distributions $\Delta(A)$, satisfies $(D, d) - Lipschitz$ property if for every $x, y \in I$ the following equation is true.*

$$D(M(x), M(y)) \leq d(x, y). \tag{1}$$

Two people have a similar treatment if they have identical features, except for the protected attribute, and identical outcomes [Lohia *et al.*, 2019]. The similarity constraints detect unfair classification and check for similar treatment. If two instances are identical but have different classifications, the fair mechanism flips the unfavored prediction by the class assigned.

Consistency is a standard metric that measures how fair the classification of a set is [Zemel *et al.*, 2013]. The consistency considers the neighborhood of each instance $i \in I$, that contains an individual's data, and $|I|$ represents the size of the set I. For all instances, it compares the model's prediction of i to its k nearest neighbors by using kNN (k Nearest Neighborhood) algorithm [Cover and Hart, 1967].

Equation 2 shows how to compute consistency, given observed target value y and expected value $\hat{y}$.

$$Consistency = 1 - \frac{1}{|I|k} \sum_{i \in I} |\hat{y}_i - \sum_{j \in kNN_i} \hat{y}_j| \tag{2}$$

2.2 Bias Mitigation Methods

Numerous strategies exist to reduce underlying bias through classification in social contexts [Ramos Salas *et al.*, 2017; Mutalemwa *et al.*, 2008; Shih *et al.*, 2013]. Here, we discuss the algorithmic strategies. The three categories where bias mitigation methods can be applied are pre-processing, in-processing, and post-processing [Pitoura *et al.*, 2021].

Pre-processing methods consist of transforming training data. This class of methods does not rely on the model modification. Therefore, predictions based on the pre-processed training data should be free of unfair classifications, e.g., achieving fairness through a database repair [Salimi *et al.*, 2019].

In-processing methods modify an existing algorithm or introduce a new one. An approach can apply fairness by constructing a step in the algorithm that results in fair ranking, classification, or recommendation. A fair algorithm can apply fairness by formulating an optimization problem considering constraints to mitigate discrimination or adding a regularization term to the objective function [Kamishima *et al.*, 2012]. Another way to incorporate fairness in models' construction is to modify the objective function to include fairness metrics, such as consistency [Zemel *et al.*, 2013].

Finally, we have the post-processing methods, which modify the outcomes of a system. Techniques in this category do not change the algorithm's operation. The model can achieve fairness by flipping the predicted values [Hardt *et al.*, 2016; Pleiss *et al.*, 2017].

3 Related Work

The related work was selected based on three points: new individual fairness contributions using a relaxation of the fundamental definition, works that handle the fairness-utility problem, and the study of fairness in machine learning. The first related work proposes iFair, an approach that preserves fairness between individuals while minimizing or bounding the utility loss [Lahoti *et al.*, 2019a]. The second work uses deep neural networks to guarantee their new fairness constraint [Kim *et al.*, 2022]. Lastly, the third work constructs a graph with fair representations, connecting similar instances [Lahoti *et al.*, 2019b].

The work Lahoti *et al.* [2019a] constructs a pre-processing approach to learning a fair data representation[1] introducing a probabilistic map of individuals' records to a low-rank data representation. It designs an optimization problem that uses fairness constraints based on legal requirements. The model demands a numeric matrix representing the set of instances and returns the updated input matrix that assures the fairness properties. They also use a threshold to bind the distance between individuals. The optimization model selects the fair-aware mapping that minimizes the objective function, which considers the utility and fairness loss.

Kim *et al.* propose SLIDE, a new convex surrogate fairness constraint[2]. Their constraint is computationally feasible, and they theoretically prove it and experiment with it using benchmark fairness datasets. SLIDE is applied to prediction models trained by in-processing methods using an optimization model to guarantee fair and accurate outputs. Their function was inspired by Ψ learning [Shen *et al.*, 2003] to compute the loss. Since Ψ function is not appropriate for a surrogate fairness constraint, Kim *et al.* modified it to become SLIDE. The SLIDE constraint depends on a parameter τ, which is the relaxation parameter for uniform individual fairness [Yona and Rothblum, 2018], also a relaxation of Dwork's definition [Dwork *et al.*, 2012]. It is common to use some functions as surrogate functions. However, they are not asymptotically equivalent to the original fairness constraint, SLIDE was designed to satisfy fairness constraints asymptotically while achieving it in a fast convergence rate.

In another work, Lahoti *et al.* describe the Pairwise Fair Representation (PFR), a fairness operationalization of individual fairness that does not need human judgments to specify a distance metric [Lahoti *et al.*, 2019b]. A key strength of this work lies in its successful demonstration of how its approach tackles the utility-fairness problem effectively. The PFR collects information about individuals who equally deserve a benefit. Lahoti *et al.*'s approach constructs a fair graph and learns about the fair representation of individuals. The authors presented two approaches to construct a fair graph: (i) a fairness graph for comparable individuals and (ii) another one for incomparable individuals. They formulate an optimization problem to learn how to represent pairs fairly. The objective function is to minimize the distance between individuals of different groups that were similarly classified [Lahoti *et al.*, 2019b].

4 λ-Fairness

This section describes our novel definition of a relaxed individual fairness definition.

Once an algorithm achieves fairness, it may damage the utility of the model. A model tends to classify items, minimizing incorrect predictions. In cases where the favored group is the majority group — i.e., the dataset or sample is mainly composed of people with a particular characteristic —, the model understands that a false negative for favored is worse than a false negative for unfavored [Kearns and Roth, 2019]. There is a fairness violation when the model gives someone (or some group) an advantage over another.

When the training set discriminates instances with individuals' data, the achievement of fairness may decrease the accuracy of the predictions. Individuals are injured when a model is not adapted to mitigate unfairness or to update itself to achieve fair classifications. Nevertheless, it is necessary to guarantee that the algorithm does not injure individuals by their protected characteristics.

We aim to achieve fairness while maintaining acceptable levels of utility by relaxing the definition of individual fairness. The unfair classification of an individual x occurs when it has similar abilities of a favored person y and the model benefits y but denies a grant to x.

Our definition, called λ-fairness, bounds the minimum fairness rate allowed in the classification model, e.g., the algorithm needs to be fair for at least a subset of a dataset where this subset has a pre-determined size. We consider that an individual's data x has a fair classification when Lipschitz Mapping given by Equation (1) is satisfied when comparing x to all the other instances in the dataset.

Definition 2 (λ-fairness) *A model is λ-fair over a dataset T when at least a proportion λ of instances in T satisfies Lipschitz*[1].

A model classifier can achieve λ-fairness using the methods described in Section 2.2. The impact of λ-fairness guarantees on utility is uncertain and dataset-dependent. Their effect varies based on the training data used. In other words, a model can achieve λ-fairness over a dataset without modifying the input data, model execution, or the model's outcomes. In case of modification, the utility may be impacted, depending on the dataset and model's prediction utility.

Definition 2 is based on the hyperparameter λ, representing the desired fair classification rate. The proportion measurement is the fairly classified samples over the total instances. As a consequence of achieving λ-fairness, the model holds or slightly modifies utility.

The data holder sets the proper λ given a specific scenario, e.g., the right choice of the hyperparameter should address the balance between utility and fairness. The analytical investigation shows how the fairness and utility metrics behave for each choice of proportion. Some applications and datasets are more sensitive to the hyperparameter. The data itself may have a considerable amount of bias, implying that the impact of the λ is significant for the classifier. Given the data holder's interests, the model is adapted to reach the desired amount of fairness in the application.

An ideal approach maximizes the utility while ensuring the fairness constraint. So, the approach must modify the training data, the model operation, or the model's outputs until it reaches the constraint (λ-fairness) while maximizing the utility metric.

[1] Source available on https://github.com/plahoti-lgtm/iFair.

[2] Source available on https://github.com/kwkimonline/SLIDE.

The next section presents an approach to achieve the λ-fair definition. Our approach uses a post-processing method that changes the classification returned by the model whenever necessary, minimizing the modifications and presuming the topics discussed in the current section.

5 LAGOON

This section presents our post-processing approach, called LAGOON (achieving bounded individuaL fAirness throuGh classificatiOn frequency equalizatiON). We discuss social concerns and introduce the idea of frequency equalization. Then, LAGOON is described in detail and applied to classification models.

The context we address is as follows. Consider a binary classification problem. The predictive model maps data points to classes $a \in \{0, 1\}$. For simplicity, we consider one protected attribute. Since the classification must not depend on the protected attribute, the data holder has to suppress it before the model training.

We describe a person as "favored" if they have an advantage over another, while a person without such advantage is "unfavored". For example, in Table 1, the data of individual 10 has an advantage over the data of individual 9. Both have one bank loan and did not delay the credit payment, but the classifier categorized the individual's 9 data as a bad payer (unfavored) differently from the individual's 10 data (favored).

5.1 Social concerns

When striving for λ-fairness, certain questions require human attention to determine the appropriate method. For example, it is important to decide which data needs to be modified. Should we update the data of individuals who are very different and have different outcomes, or should we modify the data of those who are more similar? To illustrate this point, let us consider Table 1 and make two comparisons: (a) between individuals 9 and 10, and (b) between individuals 1 and 9. In (a), the individuals are quite similar, while in (b), they are slightly different. Although both pairs have different outcomes, we must determine which approach is fairer: to modify the data of the most different or most similar individuals.

It is suitable that the individuals with characteristics in common for doing a task have the same classification, so it is more urgent to guarantee it. Given that, the pair of individuals (9 and 10) has priority over the pair (1 and 9). As individual 9 is unfavored, the model has to promote a method to ensure individuals 9 and 10 have the same chance to be accepted to receive credit.

5.2 Frequency Equalization

As discussed in Section 4, we use Lipschitz to detect unfair classifications. Once an algorithm identifies the instances that do not have a fair classification, the next step is to apply a technique that fixes the unfairness for those instances — addressing the accomplishment of λ-fairness. We propose a post-processing histogram-based method to modify classifications when necessary.

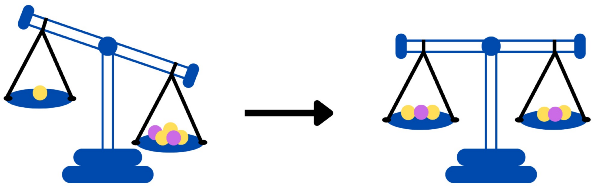


Figure 1. Scales balancing.

Histograms are useful for illustrating discrete distributions and interpreting data since they are visual tools. Converting the decisions made by certain classification models into frequency distributions is a straightforward process. This is because these models provide the count of classes. For instance, in a decision tree, the leaves contain the count of instances that reach that particular path from the root to the leaf for each target class. Therefore, we can map an instance to its leaf information, which contains a frequency distribution. Our proposal aims to equalize the counts of the classes between two frequency distributions to satisfy Constraint 1.

Imagine two scales where one side has more objects than the other. Some objects must be transferred from one side to another to balance the machine's plates, as Figure 1 illustrates. The same occurs in our approach, but now considering the frequency of classes representing the objects. Each plate represents a configuration, and the aim is to reorganize the objects of each configuration to equalize both.

The concept of "equalizing frequencies" is similar to the previous example. Figure 2 shows two frequency distributions considering a binary classification. Balancing consists of moving some units of a frequency distribution to another until both have similar behavior. In Figure 2, the darker area, colored magenta, in class 0 of frequency distribution A, represents the units that must move to class 0 of distribution B to take place in the highlighted area. This process represents the equalization.

We define the flow movement of units by taking k units of the frequency distribution with more occurrences in a certain class and transferring those k units to the same class of the other frequency distribution. To perform equalization between two distributions, we select the class with the most significant occurrence difference between the two distributions. This is done by calculating the absolute value of the difference in the occurrence of each class from the distribution A to B. The frequency of the class with the greatest difference is then chosen as the object of equalization.

Figure 2 illustrates the redistribution of units of class 0 in the frequency distribution A to the class 0 of the frequency

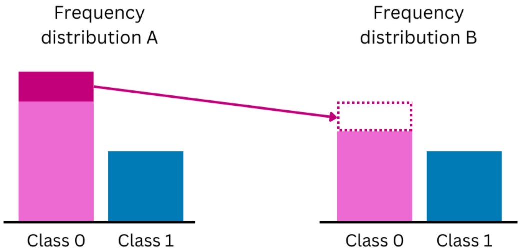


Figure 2. Flow movement of units from a frequency distribution A to a frequency distribution B.

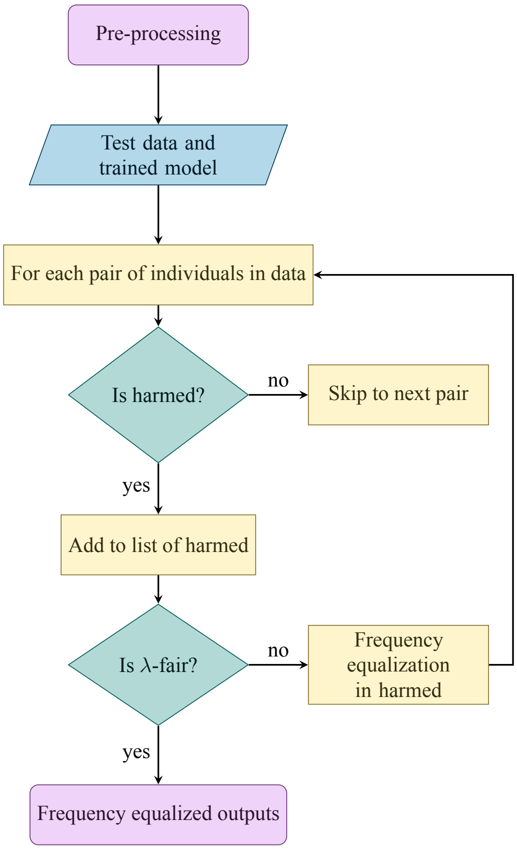

Pre-processing
Test data and trained model
For each pair of individuals in data
Is harmed?
no
Skip to next pair
yes
Add to list of harmed
Is λ-fair?
no
Frequency equalization in harmed
yes
Frequency equalized outputs

Figure 3. LAGOON process flowchart.

distribution B. The same can occur for class 1 if it is necessary. If the transfer of units occurs in both classes, the distributions tend to be the same after the equalization process.

The frequency equalization solves the problem of unfair classifications by guaranteeing that both frequency distributions are similar when the fairness constraint is not satisfied. The following section shows how to implement this technique to satisfy fairness constraints.

5.3 LAGOON Approach

We have developed an algorithm called LAGOON (individuaL fAirness throuGh classificatiOn frequency equalizatiON) to conduct experiments using frequency equalization for achieving λ-fairness. When the proportion of injured individuals in the dataset exceeds $1 - \lambda$, LAGOON runs the frequency equalization process.

To achieve λ-fairness, a proposed solution involves a series of steps that are outlined in Figure 3. The rectangle with rounded corners symbolizes the beginning and end of the algorithm. The trapezium represents input data, the rectangle represents processes, and the lozenge signifies decision points. The arrows indicate the next step in the flow. The algorithms described in this section are denoted using the symbols mentioned in Table 2.

Algorithm 1 creates a set to store the frequency distributions mapped to each instance (lines 1-4). Then, lines 7-9 compute the distance between all pairs of instances, and line 11 normalizes these distances to lie between 0 and 1. Finally, the algorithm returns the set of frequency distributions mapped for each instance and the normalized distances between all pairs of instances.

Algorithm 1 Get distance and frequency distributions

```
Require: set of instances $I$, array of targets $y$, mapping $M$,
    set of classes $A$.
Ensure: Distances $d$ and frequency distributions over the
    outcomes $\Delta(A)$.
 1: $\Delta(A) \leftarrow \varnothing$
 2: for each instance $i$ in $I$ do
 3:     $\mu_i \leftarrow M(i)$
 4:     $\Delta(A) \leftarrow \Delta(A) \cup \mu_i$
 5: end for
 6: $d \leftarrow \varnothing$
 7: for each pair of instances $(i, j) \in I \times I$ do
 8:     $d_{i,j} \leftarrow$ euclidean_distance$(i, j)$
 9:     $d \leftarrow d \cup d_{i,j}$
10: end for
11: $d \leftarrow$ Normalize$(d)$
```

Algorithm 2 works as follows. Firstly, it uses Algorithm 1 to get each instance's distances and frequency distribution (line 1). Next, in lines 3 and 4, the algorithm verifies if each pair of instances satisfies the Lipschitz constraint. If a pair of instances does not satisfy the constraint, it is added to a list of harmed instances (lines 5 and 6). In line 9, the set of harmed instances is updated by classifying them from most to least harmed.

Algorithm 3 implements a single round of frequency equalization for two instances that have been harmed. The first step of the algorithm, in line 1, is finding the instance j that is most similar to a given instance i. The algorithm aims to equalize the most different class in the frequency distributions as a priority to converge faster towards fairness. To achieve that, it compares the frequency distributions of the

Variable	Definition
I	Set of instances in dataset
A	Set of classes
$M(i)$	Mapping that assigns an instance $i \in I$ to a frequency distribution over the classes A
y	Array of classes
d	Distance metric
D	Dissimilarity metric between two frequency distributions
k	Integer that represents how many units need to be swapped from a distribution to another
c	Class $c \in A$
h	Array of harmed or unfair instances sorted in descending order of how harmed each instance was
$\Delta(A)$	Set of all the frequency distributions mapped to the instances
μ_i	Frequency distribution mapped to instance $i \in I$ where $\mu_i \in \Delta(A)$
$d_{i,j}$	Distance between instance i and instance j
$\mu_i[c]$	Frequency of class c on frequency distribution μ_i

Table 2. Notation table for algorithms.

Algorithm 2 Find harmed instances

```
Require: Set of instances $I$, array of targets $y$, mapping $M$,
    set of classes $A$.
Ensure: Set of harmed instances $h$ sorted by their prejudice
    and frequency distributions over the outcomes $\Delta(A)$.
    /* get_distance_and_freq function refers to Algorithm 1.
    */
 1: $d, \Delta(A) \leftarrow$ get_distance_and_freq($I, y, M, A$)
 2: $h \leftarrow$ empty array
 3: for each pair of instances $(i, j) \in I \times I$ do
 4:     if $d_{i,j} < \mathrm{D}(\mu_i \in \Delta(A), \mu_j \in \Delta(A))$ then
 5:         $h.add(i)$
 6:         $h.add(j)$
 7:     end if
 8: end for
    /* sort_by_count function sorts the array $h$ in descending
    order, i.e., if an instance $i$ appears more than any instance
    in $h$, $i$ will be in the first position of $h$. So it sorts $h$ based
    on damage suffered by the instances. */
 9: $h \leftarrow$ sort_by_count($h$)
```

Algorithm 3 Frequency equalization

```
Require: instance $i$, harmed instances $h$
Ensure: Updated frequency distributions $\mu_i, \mu_j \in \Delta(A)$.
    /* most_similar_instance_in_harmed(i, h) searches the
    instance in the harmed array which is the most similar to
    i. */
 1: $j \leftarrow$ most_similar_instance_in_harmed($i, h$)
    /* select_most_different_class function compares each
    class of two frequency distributions $\mu_i, \mu_j \in \Delta(A)$ and
    returns the most different class $c \in \{0, 1\}$ in the two
    distributions. */
 2: $c \leftarrow$ select_most_different_class($\mu_i \in \Delta(A), \mu_j \in$
    $\Delta(A)$)
 3: $k \leftarrow \lfloor \frac{|\mu_i[c] - \mu_j[c]|}{2} \rfloor$
 4: if $\mu_i[c] > \mu_j[c]$ then
 5:     $\mu_i \leftarrow \mu_i - k$
 6:     $\mu_j \leftarrow \mu_j + k$
 7: else
 8:     $\mu_i \leftarrow \mu_i + k$
 9:     $\mu_j \leftarrow \mu_j - k$
10: end if
```

Algorithm 4 LAGOON

```
Require: set of instances $I$, array of targets $y$, mapping $M$,
    set of classes $A$, float value $\lambda$.
Ensure: Updated set of frequency distributions over the out-
    comes $\Delta(A)$.
    /* pre_process function standardizes features, removes
    non-essential features, and drops null values. */
 1: $I, y \leftarrow$ pre_process(I, y)
    /* find_harmed function refers to Algorithm 2. */
 2: $h, \Delta(A) \leftarrow$ find_harmed($I, y, M, A$)
    /* $h$ is a sorted array based on how harmed the instances
    are. The most harmed instance is in the first position: in-
    dex zero. */
 3: most_harmed $\leftarrow h[0]$
    /* frequency_equalization function refers to Algorithm
    3. */
 4: while $1 - \frac{|h|}{|I|} < \lambda$ do
 5:     frequency_equalization(most_harmed, $h$)
 6:     $h, \Delta(A) \leftarrow$ find_harmed($I, y, M, A$)
 7:     most_harmed $\leftarrow h[0]$
 8: end while
```

two instances and retrieves the most different class, which is then saved in variable c (line 2). Lastly, in lines 3-10, the algorithm updates the frequency distribution of class c for both instances by moving units from one distribution to the other. The updated frequency distributions are returned by the algorithm.

Algorithm 4 performs a pre-processing step on the data in line 1. This step involves removing the protected attribute, normalizing the columns to ensure that they all have the same weight, and encoding the target attribute into binary format if necessary. The algorithm then, calls Algorithm 2 and 3 until λ-fairness is satisfied (lines 2-8), i.e., until the proportion of fair instances is less or equal to λ.

The fairness constraint on line 4 of Algorithm 4 ensures the definition of λ-fairness through updates made on the frequency distribution of the harmed individuals. These updates continue until the model achieves its objective, ensuring the definition of fairness. The experimental analysis shows that λ-fairness can achieve high levels of individual fairness, as measured by consistency.

LAGOON is a generic and simple technique with adaptability for different classifiers, as shown in the next sections.

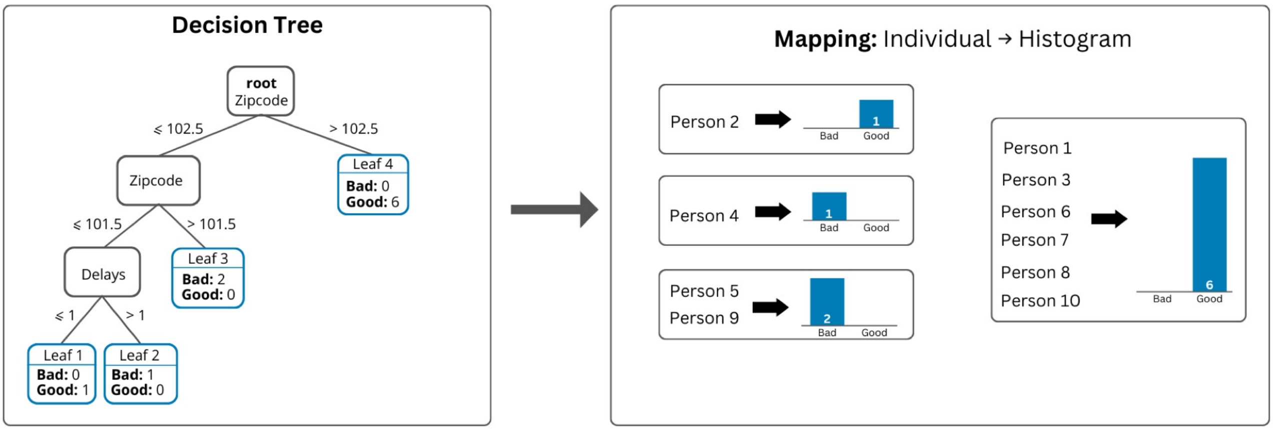
6 Applications

6.1 Fair Decision Tree

Decision Tree is a famous classification model and is hugely used in many different applications because of its simplicity and explainability [Almuallim *et al.*, 2002; Blockeel *et al.*, 2023]. The main advantage of using this model is its interpretability when limiting the height of the tree [Costa and Pedreira, 2023]. It constructs paths that visually guide a sequence of decisions to an outcome. The terminal nodes, also known as leaves, store the counts of each class in a path from the root to the leaf, which represents a subset. This subset has a population sample that satisfies the constraints in the path. In other words, the leaf's data can be described as a frequency distribution.

Figure 4 shows a mapping procedure for the Decision Tree classifier. The illustration (a) considers a decision tree applied on the running example 1. In (b), a mapping assigns individuals to frequency distributions extracted from the leaves. For example, person 2 data belongs to the path between root and leaf 1, so person 2 is mapped to the frequency distribution of leaf 1. The same occurs to person 5 and 9; their data is assigned to the path of root to leaf 3, so they are mapped to the frequency distribution of leaf 3, and so on.

LAGOON computes the distances between individuals in test data and the dissimilarity between distributions assigned to those individuals. The distributions depend on the trained model since the mapping M of an instance to a distribution



(a) (b)

Figure 4. Mapping of instances of the running example 1 to frequency distributions captured of the decision tree leaves.

relies on it. Algorithm 5 implements a λ-fair Decision Tree using LAGOON. First, it constructs the Decision Tree based on training data. After that, the model applies LAGOON on test data to achieve λ-fairness.

Algorithm 5 λ-fair Decision Tree

```
Require: set of instances $I$, array of targets $y$, $\lambda$.
Ensure: $\lambda$-fair model.
    /* split_data function splits data and their corresponding
    labels in train and test, the size of each set is defined
    based on test_size.*/
    train, test = split_data($I, y$, test_size)
    classifier ← DecisionTree
    classifier.fit(train)
    /* The mapping function maps a set of instances to a set
    of frequency distributions obtained after accessing the
    leaves of the Decision Tree classifier. */
    M ← mapping(train.instances, classifier.leaves)
    classifier.leaves ← LAGOON(test.instances,test.$y$, $M$, $\lambda$)
```

6.2 Fair Histogram-based Gradient Boosting

Gradient Boosting is a classification model that uses ensembles of trees to predict classes. It adds tree models sequentially to the ensemble, and each tree updates the prediction, minimizing the error made by the previous tree. In the end, the ensemble assigns each class to a probability based on the prediction of the set of trees on it. The difference between classic Gradient Boosting and Histogram-based Gradient Boosting (HGB) is that HGB uses a technique that discretizes the continuous values. It bounds the size of the bins with discrete values to reduce the computation of the tree construction.

The implementation of LAGOON in HGB addresses mapping individuals to the probability distribution of the predicted classes. The λ-fair HGB is similar to the λ-fair Decision Tree, but the classifier differs, and the probability distribution is directly calculated. The frequency equalization occurs in a sampling based on the class probabilities to ensure that the balancing occurs in discrete values. The Algorithm 6 shows how to implement the described process.

Algorithm 6 λ-fair Histogram-based Gradient Boosting

```
Require: set of instances $I$, array of targets $y$, train_size, $\lambda$.
Ensure: $\lambda$-fair model, $predicted\_y$, test.$y$.
    train, test = split_data($I, y$, test_size)
    classifier ← HGB
    classifier.fit(train)
    $M$ ← mapping(train, classifier.probs)
    classifier.probs ← LAGOON(test,test.$y$, $M$, $\lambda$)
```

6.3 Histogram-based Fair Cohort

We propose the Histogram-based Fair Cohort approach (HBFC) to predict the individual's class using a clustering technique. It has three steps:

i. Groups the individuals by their features' value.
ii. Assign each cluster to a histogram.
iii. Predicts the class of each instance in the test set based on the histogram assigned to the cohort in which the instance belongs.

Considering the running example, individuals 5 and 9 are grouped in the same cluster because they have the same values in *zipcode, credits, and delays*. HBFC assigns a histogram for each cluster with the frequency distribution of classes. Figure 5 shows an example of this approach applied to the running example. For simplicity, we consider only two features (*credits* and *delays*) to construct the cohort.

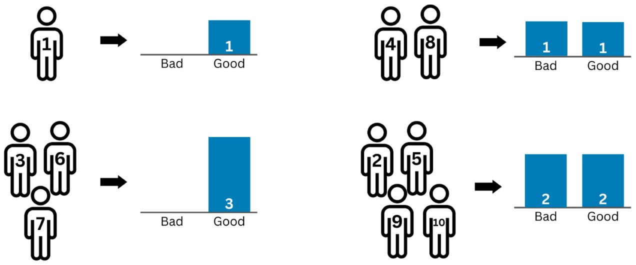
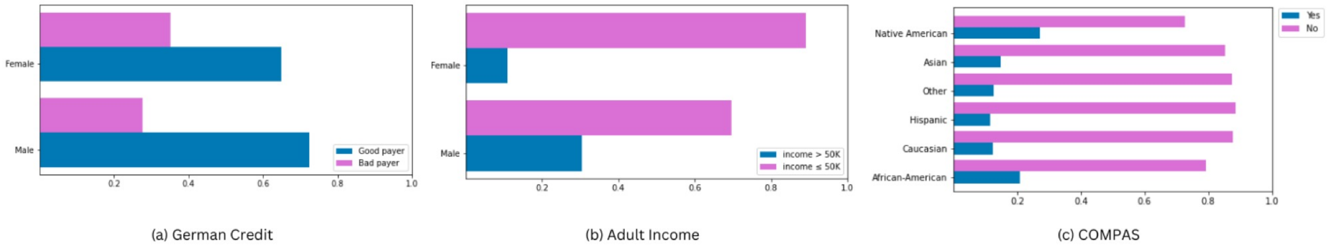


Figure 5. Illustration of HBFC clusterings and assignments.



(a) German Credit

(b) Adult Income

(c) COMPAS

Figure 6. Proportion of classes for each protected feature.

The difference between the implementation of the λ-fair HBFC and λ-fair Decision Tree using LAGOON is that the λ-fair HBFC model has no leaves. The model maps the cohorts to histograms directly. So the modification would be in lines 1, 3, and 4 of Algorithm 5, where the classifier would be HBFC and the *classifier.leaves* would be *classifier.histograms*.

6.4 Fair Neural Network

Similarly to HGB, LAGOON achieves λ-fairness in Neural Networks by sampling instances based on the classes probabilities composed by the output layer. Then, the frequency distributions are balanced using frequency equalization until the model guarantees λ-fairness.

Algorithm 7 uses the trained neural network to find the mapping that assigns each instance to a frequency distribution. Then, LAGOON is used to equalize the frequency probabilities of the instances that do not satisfy fairness.

Algorithm 7 λ-fair Neural Network

```
Require: set of instances $I$, array of targets $y$, train_size, $\lambda$.
Ensure: $\lambda$-fair model, $predicted\_y$, test.$y$.
train, test = split_data($I$, $y$, test_size)
classifier ← NN
classifier.fit(train)
$M$ ← mapping(train.instances, classifier.probs)
classifier.probs ← LAGOON(test.instances,test.$y$, $M$, $\lambda$)
```

7 Experiments

In this section, we evaluate our approach experimenting with LAGOON to achieve λ-fairness and comment on its results. For experiments, we evaluate the four applications described in Section 6: Fair Decision Tree (DT), Fair Histogram-based Gradient Boosting (HGB), Histogram-based Fair Cohort (HBFC), and Fair Neural Networks (NN).

We focus our experiments on three well-known applications in fairness literature: payer profile, annual income, and criminal issues. Considering different applications shows that the proposed method may be applied in a variety of contexts. The sets used for each application are German Credit, Adult Income, and COMPAS, respectively. The three datasets are imbalanced, which may lead to biased predictions considering classifiers that do not include fairness [Chawla, 2010]. We investigate the impact of LAGOON on utility and consistency by analyzing the prediction results in those datasets after applying our fair method in the models. Each dataset is described in the sequel. Table 3 summarizes the description.

Table 3. Table with the description of the datasets used for the experiments for each application.

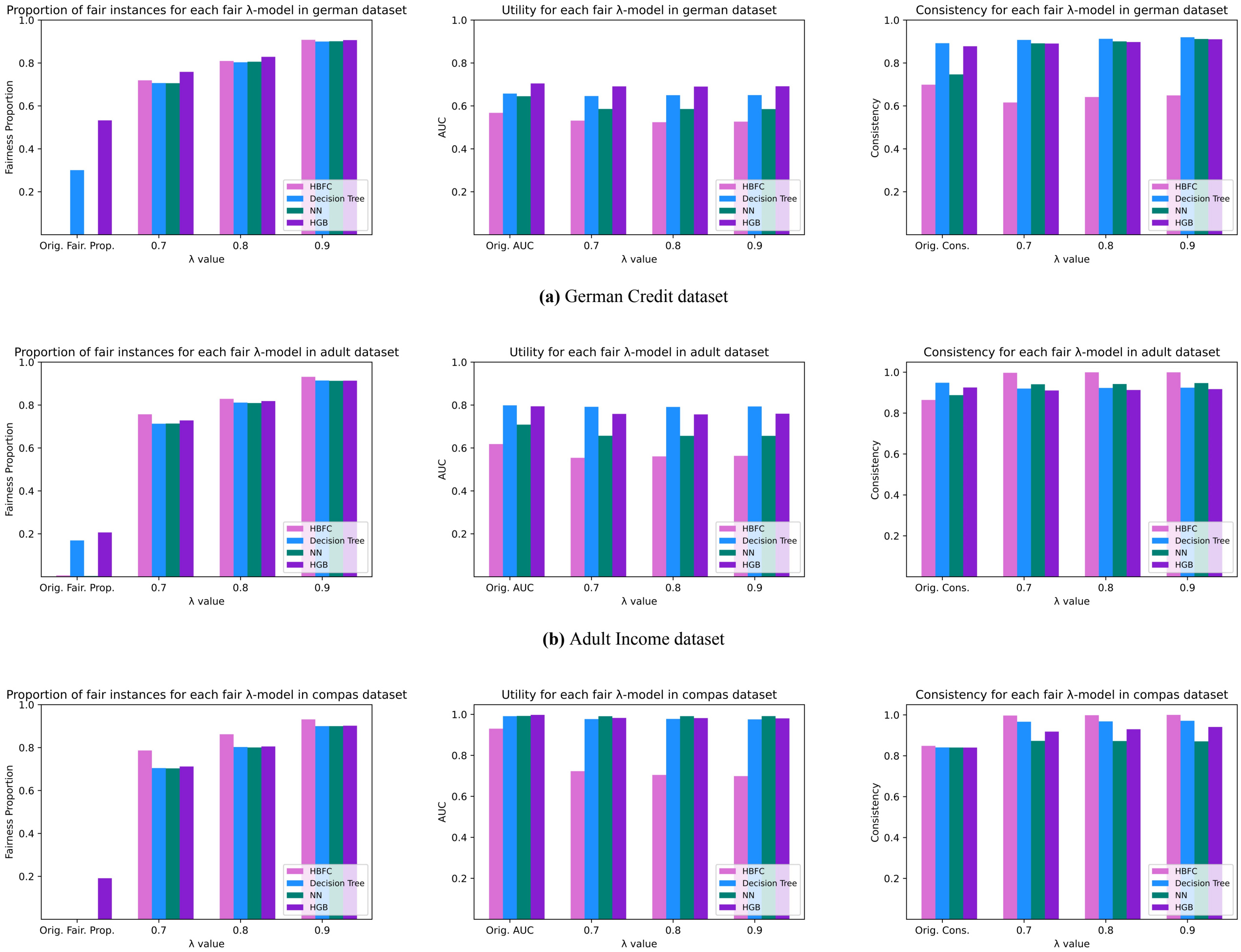
Dataset	#Rows	#Columns	Context
German Credit	1000	20	Payer Profile
Adult Income	48842	14	Income
COMPAS	4743	54	Criminal/Social Issues

Payer Profile: The dataset used for this application is German Credit [Dua *et al.*, 2017], provided by Prof. Hans Hofmann from the *Universit at Hamburg*. It reflects the payment profile of individuals by analyzing their banking history, e.g., how many credits the person requested from the bank, whether there was a delay in credit payment, and some personal attributes. The classification possibilities for a person are good payer or bad payer. The protected attribute is gender, where the favored value is Male. The features used to classification task are: *checking_acc*, *credit_historic*, *saving_acc*, *atual_employ_since*, *installment_rate*, *housing* and *credits_at_bank*.

Annual Income: The chosen dataset for this scenario is the Adult Income dataset [Dua *et al.*, 2017]. The 1994 US Census collected personal data, which associates an income with an individual based on personal attributes, e.g., age, marital status, and education. Each instance has an individual's data. There is an assignment for those instances with an income higher than 50 thousand per year or less than such value. As in the previous application, the protected attribute is the gender of individuals represented in each row where the favored value is Male. For the classification task, we consider the following features: *age*, *workclass*, *education*, *marital-status*, *occupation*, *relationship*, *race* and *native-country*.

Criminal Issues: The dataset representing this context is the COMPAS [Larson *et al.*, 2016], owned by Equivant (previously Northpointe). It contains data collected by Broward County, Florida's COMPAS risk assessment tool. In this system, each sample predicts the recidivism risk of individuals based on personal attributes and criminal history. The target attribute indicates whether an individual has relapsed into crime over the last two years. The protected attribute is "race", in this application. Historically, the unfavored or unprivileged value is "African-American", and the privileged is "Caucasian". The features that are being used for the classification task are: *sex*, *age*, *juv_fel_count*, *decile_score*, *juv_misd_count*, *juv_other_count*, *is_recid*, *is_violent_recid* and *score_text*.

As discussed throughout this paper, the applications have



(a) German Credit dataset

(b) Adult Income dataset

(c) COMPAS dataset

Figure 7. Results of applying LAGOON to each configuration. The metrics in on the y axis.

a favored feature of the protected attribute, primarily based on historical discrimination, and some data analysis or historical study can identify it. Figure 6 shows the proportion of classes for each feature of the protected attribute in the different contexts' datasets.

The datasets are imbalanced considering the target classes in each protected value, as shown in Figure 6. The imbalance also occurs in the three datasets when considering the target attribute and the protected attribute separated. In the German dataset, 69.97% of instances are labeled as "Good" and 30.03% are labeled as "Bad". In addition, in the dataset, 68.97% of the individuals are men, while only 31.03% are women. The adult income dataset has 76.07% of instances labeled as "$\leq 50K$" and 23.93% are labeled as "$> 50K$", and 66.85% of data belongs to men while 33.15% belongs to women. Further, "83.65%" of people in the compas dataset did not relapse in crime in two years, and "16.35%" relapsed. The races in this dataset are very imbalanced, where the unprivileged race appears in 47.82% of instances, and the sum of other races is 52.18%.

The metrics we used to evaluate our approach are:

1. The proportion of similar treatment (Equation 1) for individuals in a dataset to compute fairness.
2. The area under the ROC curve (AUC) for utility.
3. The consistency [Zemel *et al.*, 2013].

The AUC metric is chosen to measure the prediction performance for classification tasks because it considers the true and false positive rates (TPR and FPR). The consistency metric uses the kNN algorithm, so it first finds the $k = 5$ nearest neighbors of an instance and computes how similar is the treatment of an instance compared to its neighborhood. The consistency captures the concept of individual fairness that classifies similars individuals similarly.

We ran the LAGOON approach fifteen times and took a mean for each configuration, dividing the datasets into three parts: 60% train, 20% validation, and 20% test. We stratify data to guarantee that the target feature is proportionally divided. We performed a grid search to validate our data and choose the hyperparameters. The configuration includes the metric, model, λ value, and application choice. The possible values of λ are: 0.7, 0.8 and 0.9. When growing λ value, the fair model tends to reduce the utility since we modify the predicted classification.

Figure 7 shows the graphic of the results for each configuration. The illustration contains the proportion of fair instances after applying λ-fairness, the AUC, and the consistency for each dataset and application.

Table 4. Comparison between the baseline approaches and LAGOON applied to Decision Tree and Neural Networks using $\lambda = 0.9$.

Approach	German		Adult		COMPAS	
	AUC score	Consistency	AUC score	Consistency	AUC score	Consistency
iFair (DT)	0.4889	0.8350	0.4845	0.8891	0.5116	0.9584
SLIDE (NN)	0.5007	0.9968	0.5615	0.9508	0.9154	0.9677
LAGOON (DT)	0.6505	0.9200	0.7934	0.9242	0.9755	0.9712
LAGOON (NN)	0.5851	0.9118	0.6565	0.9467	0.9914	0.8705

7.1 Payer Profile results

Figure 7a shows the results of applying LAGOON to the German Credit Risk dataset. Although the utility has not suffered significant modification, the λ-fair models significantly improved the proportion of fair tuples in that application. The original fairness rates in the classified data were near zero for HBFC and NN, and 0.3006 and 0.5326 for Decision Tree and HGB, respectively. The fairness rates improved proportionally to λ since it bounds the minimum proportion of fair instances allowed. As expected, the fair models had a slight decrease in utility rates when varying λ. As one can see in Figure 7a, the consistency of Neural Network had a significant improvement after LAGOON application that reached 0.9118 for $\lambda = 0.9$.

Applying LAGOON to the HGB model produced the best results for this application. The fair proportion, utility, and consistency rates for $\lambda = 0.9$ were 0.9066, 0.6916, and 0.9105, respectively. LAGOON applied to the Decision Tree also produced promising results. The utility remained almost the same, decreasing from 0.6576 to 0.6505, the fair proportion of instances increased from 0.3006 to 0.9000, and the consistency also had a little improvement of 0.03 added from the original, as showed in Figure 7a .

7.2 Annual Income results

Figure 7b illustrates the results of LAGOON in a census application. Although HBFC had the best results for fair proportion and consistency rates, it did not perform well in ROC AUC. On the other hand, as you can notice in Figure 7b, the λ-fair Decision Tree maintained the utility using the three different values of λ, reaching a rate of about 0.792. The fair proportion of instances for the 0.9-fair Decision Tree was 0.9143, and the consistency was 0.9242.

LAGOON applied to HGB with $\lambda = 0.9$ also reached good results. The fair proportion, utility, and consistency rates are 0.9135, 0.7597, and 0.9171, respectively. Next, 0.9-fair NN achieved 0.9129 of fair instances, 0.6565 on utility, and 0.9467 of consistency. As expected, the utilities decrease when comparing the original AUC score and the utility of λ-fair models, but the models previously mentioned maintained a score near the original.

7.3 Criminal Issues results

The proportion of fairly classified instances calls attention in the context of criminal recidivism in Figure 7c. Including LAGOON for HBFC, DT, NN, and HGB makes a great difference in the proportion of fair instances. However, the utilities were not hugely harmed, except for HBFC. HBFC was visually highlighted by the increase of fair proportion and consistency in the three $lambda$ values, but its AUC score decreased significantly, differently from the other models. Although the consistency reached a perfect result, i.e., 1.0, the fair model achieved a utility of 0.6987.

The other models achieved a high level of utility and very similar rates for that metric. LAGOON applied to Decision Tree performed the best results, considering the three metrics. The results for $\lambda = 0.9$ were 0.9000, 0.9755, and 0.9712 for the proportion of fair instances, utility, and consistency rates, respectively.

7.4 Analysis considering competitors

We ran the same samples and used the same fairness, consistency, and AUC score metrics to compare LAGOON to the literature work, except for Lahoti *et al.* [2019b], although it is an open-source code[3] the graphs were not available so we could not compare it to LAGOON. Using it to compare the approaches may be biased since the graph's construction could have infinite interpretations and possible connections.

We used the Decision Tree model for iFair [Lahoti *et al.*, 2019a], as their approach uses a method that modifies the training data. For SLIDE [Kim *et al.*, 2022], we maintained the model that the authors presented in their paper, the Deep Neural Network, using $\tau = 0.1$ and the Stochastic Gradient Descent (SGD) optimizer. We ran the approaches and compared the results of LAGOON applied to Neural Networks using $\lambda = 0.9$. All the experiments followed the same pattern to mitigate any possible bias. The *consistency* on Table 4 measures how far an instance's classification is of its similars. We compute the consistency by comparing the sample and the five closest neighbors' classifications. The set value for the length of the neighborhood is 5.

Our approaches scored better than the baselines on the German dataset when considering the utility. We improved the AUC score in 0.1616 comparing LAGOON applied to Decision Trees and iFair, and in 0.0844 comparing LAGOON applied to Neural Networks and SLIDE. Except for iFair, all the approaches achieved a high level of consistency.

We also achieved the best results in terms of utility when classifying Adult dataset instances. We reached an AUC score of 0.7934 using LAGOON applied to Decision Trees, and our direct competitor reached 0.4845. In addition, we gained 0.095 when comparing LAGOON applied to NN and the respective competitor SLIDE. All the approaches reached high levels of consistency.

The prediction of instances of the COMPAS recidivism dataset reached incredibly high utility scores in all ap-

[3] https://github.com/plahoti-lgtm/PairwiseFairRepresentations

proaches, except in iFair, despite achieving a high consistency. We could reach 0.9755 and 0.9914 in the utility of LAGOON applied to Decision Trees and Neural Networks, respectively. Although we had a high consistency on LAGOON (NN), SLIDE performed better in this metric since SLIDE is designed to be consistent [Kim *et al.*, 2022]. As we can see in the last subsection, the Decision Tree performed better than the other models after applying fairness.

In general, our approach slightly modifies the outputs, so we scored better in all datasets regarding the AUC score.

8 Conclusion

We proposed a new definition to balance fairness and utility rates, called λ-fairness, a relaxation of the individual fairness definition based on the proportion of fairly classified instances. We also developed and described new techniques to achieve the novel definition called LAGOON. We proposed four fair models that we implemented applying our fairness technique and experimented with them in three benchmark datasets commonly used in fairness literature.

The experiments showed that although fairness accomplishment might affect the utility, we can slightly relax definitions to guarantee good results when comparing the original model's prediction and the fair model, which includes a post-processing step that modifies the outputs to achieve fairness definitions. In addition, we compared our approach to works in the literature and presented the results using consistent metrics to evaluate it. We achieved better results than the baselines regarding utility while ensuring fairness.

For future work, we aim to study new ways of achieving λ-fairness using hybrid approaches that implement more than one method to optimize the parameters, maximizing utility. We also aim to expand our experiments to consider multiclass classification and more than one protected attribute.

Declarations

Acknowledgements

The authors thank LSBD for providing the necessary resources to run the experiments.

Funding

This research was funded by Lenovo, as part of its R&D investment under Brazilian Informatics Law, by *Coordenação de Aperfeiçoamento Pessoal de Nível Superior - Brasil* (CAPES) under grant number 88887.694847/2022-00 and by *Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil* (CNPQ) under grant number 316729/2021-3.

Authors' Contributions

MS contributed to the proposed conception, formal analysis, investigation, methodology, visualization, validation, and writing. IC contributed to the concept, verification, supervision, review, and editing. JM supervised, validated, and reviewed the study.

Competing interests

The authors declare that they have no competing interests

Availability of data and materials

The code and datasets generated and analyzed during the current study are available on https://github.com/malu-maia/LAGOON.

References

Almuallim, H., Kaneda, S., and Akiba, Y. (2002). Development and applications of decision trees. In *Expert Systems*, pages 53–77. Elsevier. DOI: 10.1016/B978-012443880-4/50047-8.

Argawal, P. (2021). How is automated credit decisioning transforming digital lending. Available at:https://www.birlasoft.com/articles/how-is-automated-credit-decisioning-transforming-digital-lending. Last accessed: July 29, 2022.

Blockeel, H., Devos, L., Frénay, B., Nanfack, G., and Nijssen, S. (2023). Decision trees: from efficient prediction to responsible ai. *Frontiers in Artificial Intelligence*, 6. DOI: 10.3389/frai.2023.1124553.

Calders, T. and Verwer, S. (2010). Three naive bayes approaches for discrimination-free classification. *Data mining and knowledge discovery*, 21:277–292. DOI: 10.1007/s10618-010-0190-x.

Census, Bureau (2023). United states census bureau. Available at:https://www.census.gov/quickfacts/fact/table/US/PST045223. Last accessed: July 29, 2022.

Chawla, N. V. (2010). Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, pages 875–886. DOI: 10.1007/978-0-387-09823-4_45.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163. DOI: 10.48550/arXiv.1703.00056.

Costa, V. G. and Pedreira, C. E. (2023). Recent advances in decision trees: An updated survey. *Artificial Intelligence Review*, 56(5):4765–4800. DOI: 10.1007/s10462-022-10275-5.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27. DOI: 10.1109/TIT.1967.1053964.

Dash, R. (2021). Designing next-generation credit-decisioning models. Available at: https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/designing-next-generation-credit-decisioning-models. Last accessed: July 29, 2022.

Dua, D., Graff, C., *et al.* (2017). Uci machine learning repository. Available at: https://archive.ics.uci.edu/.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. DOI: 10.1145/2090236.2090255.

Galhotra, S., Shanmugam, K., Sattigeri, P., and Varshney, K. R. (2021). Interventional fairness with indirect knowledge of unobserved protected attributes. *Entropy*, 23(12):1571. DOI: 10.3390/e23121571.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29. DOI: 10.48550/arXiv.1610.02413.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pages 35–50. Springer. DOI: 10.1007/978-3-642-33486-3_3.

Kearns, M. and Roth, A. (2019). *The ethical algorithm: The science of socially aware algorithm design*. Oxford University Press. Book.

Khan, A., Baharudin, B., Lee, L. H., and Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20. Available at: https://www.researchgate.net/publication/43121576_A_Review_of_Machine_Learning_Algorithms_for_Text-Documents_Classification.

Kim, K., Ohn, I., Kim, S., and Kim, Y. (2022). Slide: A surrogate fairness constraint to ensure fairness consistency. *Neural Networks*, 154:441–454. DOI: 10.1016/j.neunet.2022.07.027.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. *Advances in neural information processing systems*, 30. DOI: 10.48550/arXiv.1703.06856.

Lahoti, P., Gummadi, K. P., and Weikum, G. (2019a). ifair: Learning individually fair data representations for algorithmic decision making. In *2019 ieee 35th international conference on data engineering (icde)*, pages 1334–1345. IEEE. DOI: 10.1109/ICDE.2019.00121.

Lahoti, P., Gummadi, K. P., and Weikum, G. (2019b). Operationalizing individual fairness with pairwise fair representations. *arXiv preprint arXiv:1907.01439*. DOI: 10.14778/3372716.3372723.

Larson, J., Roswell, M., and Atlidakis, V. (2016). Compas. Available at:https://github.com/propublica/compas-analysis. July 29, 2022.

Law, E. U. (2016). Gdpr. Available at:https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng. Last accessed: May 08, 2023.

Lohia, P. K., Ramamurthy, K. N., Bhide, M., Saha, D., Varshney, K. R., and Puri, R. (2019). Bias mitigation post-processing for individual and group fairness. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 2847–2851. IEEE. DOI: 10.48550/arXiv.1812.06135.

Luong, B. T., Ruggieri, S., and Turini, F. (2011). k-nn as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 502–510. DOI: 10.1145/2020408.2020488.

Mashat, A. F., Fouad, M. M., Philip, S. Y., and Gharib, T. F. (2012). A decision tree classification model for university admission system. *International Journal of Advanced Computer Science and Applications*, 3(10). Available at:https://www.researchgate.net/publication/235333385_A_Decision_Tree_Classification_Model_for_University_Admission_System.

Mutalemwa, P., Kisoka, W., Nyingo, V., Barongo, V., and Malecela, M. (2008). Manifestations and reduction strategies of stigma and discrimination on people living with hiv/aids in tanzania. *Tanzania journal of health research*, 10(4). DOI: 10.4314/thrb.v10i4.45077.

Pappada, R. and Pauli, F. (2022). Discrimination in machine learning algorithms. *arXiv preprint arXiv:2207.00108*. DOI: 10.48550/arXiv.2207.00108.

Pedreshi, D., Ruggieri, S., and Turini, F. (2008). Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. DOI: 10.1145/1401890.1401959.

Pitoura, E., Stefanidis, K., and Koutrika, G. (2021). Fairness in rankings and recommenders: Models, methods and research directions. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2358–2361. IEEE. DOI: 10.1109/ICDE51399.2021.00265.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. (2017). On fairness and calibration. *Advances in neural information processing systems*, 30. Available at:https://proceedings.neurips.cc/paper/2017/hash/b8b9c74ac526fffbeb2d39ab038d1cd7-Abstract.html.

Ramentol, E., Olsson, T., and Barua, S. (2021). Machine learning models for industrial applications. In *AI and Learning Systems-Industrial Applications and Future Directions*. IntechOpen. DOI: 10.5772/intechopen.93043.

Ramos Salas, X., Alberga, A., Cameron, E., Estey, L., Forhan, M., Kirk, S., Russell-Mayhew, S., and Sharma, A. (2017). Addressing weight bias and discrimination: moving beyond raising awareness to creating change. *Obesity Reviews*, 18(11):1323–1335. DOI: 10.1111/obr.12592.

Salimi, B., Rodriguez, L., Howe, B., and Suciu, D. (2019). Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, pages 793–810. DOI: 10.1145/3299869.3319901.

Shen, A., Tong, R., and Deng, Y. (2007). Application of classification models on credit card fraud detection. In *2007 International conference on service systems and service management*, pages 1–4. IEEE. DOI: 10.1109/ICSSSM.2007.4280163.

Shen, X., Tseng, G. C., Zhang, X., and Wong, W. H. (2003). On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734. DOI: 10.1198/016214503000000639.

Shih, M., Young, M. J., and Bucher, A. (2013). Working to reduce the effects of discrimination: Identity management strategies in organizations. *American Psychologist*, 68(3):145. DOI: 10.1037/a0032250.

UK Government (2013). Equality act 2010: Chapter 1 protected characteristics. Available at:

https://www.legislation.gov.uk/ukpga/2010/15/part/2/chapter/1/2013-06-25. Last accessed: October 19, 2023.

Yona, G. and Rothblum, G. (2018). Probably approximately metric-fair learning. In *International conference on machine learning*, pages 5680–5688. PMLR. DOI: 10.48550/arXiv.1803.03242.

Zafar, M. B., Valera, I., Rodriguez, M. G., and Gummadi, K. P. (2017). Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR. Available at:https://proceedings.mlr.press/v54/zafar17a.html.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR. Available at:https://proceedings.mlr.press/v54/zafar17a.html.

Zhang, J., Xie, Y., Wu, Q., and Xia, Y. (2019). Medical image classification using synergic deep learning. *Medical image analysis*, 54:10–19. DOI: 10.1016/j.media.2019.02.010.

Zliobaite, I. (2015). A survey on measuring indirect discrimination in machine learning. *arXiv preprint arXiv:1511.00148*. DOI: 10.48550/arXiv.1511.00148.