

Optimization for Fair Classification Methods in Heterogeneous Data

Dissertation approved by Faculty IV of Trier University
for the award of the academic degree
Dr. rer. pol.

by

João Vitor Pamplona

Trier, 2025

Supervisor: PD Dr. Jan Pablo Burgard
1st Reviewer: PD Dr. Jan Pablo Burgard
2nd Reviewer: Prof. Dr. Domingo Morales González
Date of the disputation: 10.02.2025

Acknowledgements

I thank my parents, Antônio and Maria Helena, and my brother Antônio for always supporting me, no matter the circumstances. Your support is fundamental in my life, I love you all.

I thank my advisor, Jan Pablo Burgard, for believing in me and always being available in moments of difficulty.

I thank Professor Domingo Morales González and Professor Martin Schmidt for agreeing to be part of the defense committee.

I gratefully thank the support of the German Federal Ministry of Education and Research (BMBF) for this research project, as well as for the “OptimAgent Project.”

I would also like to express my sincere appreciation for the generous support provided by the German Research Foundation (DFG) within Research Training Group 2126 “Algorithmic Optimization”.

Finally, I thank my wife, who has been present in every moment of my life since the beginning of my undergraduate studies. Maria, among all the things that life could offer me, the best was certainly you.

Curriculum Vitae

Personal Information

Name	João Vitor Pamplona
Nationality	Brazilian
Date of Birth	April 24, 1997
Place of Birth	Blumenau, Santa Catarina, Brazil

Education

11/2022 - 02/2025	Research Assistant Universität Trier Trier, Rheinland-Pfalz, Germany
09/2022	Master Degree in Numerical Methods in Engineering
10/2020 - 09/2022	Universidade Federal do Paraná Curitiba, Paraná, Brazil
11/2020	Licentiate Degree in Mathematics
08/2016 - 12/2019	Universidade Federal de Santa Catarina Blumenau, Santa Catarina, Brazil
12/2014	High School Colégio Universitário Gaspar, Santa Catarina, Brazil

Abstract

Ensuring fairness in machine learning models is crucial for ethical and unbiased automated decision-making. Classifications from fair machine learning models should not discriminate against sensitive variables such as sexual orientation and ethnicity. However, achieving fairness is complicated by biases inherent in training data, particularly when data is collected through group sampling, like stratified or cluster sampling as often occurs in social surveys. Unlike the standard assumption of independent observations in machine learning, clustered data introduces correlations that can amplify biases, especially when cluster assignment is linked to the target variable.

To address these challenges, this cumulative thesis focuses on developing methods to mitigate unfairness in machine learning models. We propose a fair mixed effects support vector machine algorithm, a Cluster-Regularized Logistic Regression and a fair Generalized Linear Mixed Model based on boosting, all of them are capable of handling both grouped data and fairness constraints simultaneously. Additionally, we introduce a Julia package, `FairML.jl`, which provides a comprehensive framework for addressing fairness issues. This package offers a preprocessing technique, based on resampling methods, to mitigate biases in the data, as well as a post-processing method, that seeks for a optimal cut-off selection. To improve fairness in classifications both processes can be incorporated in any classification method available in the `MLJ.jl` package. Furthermore, `FairML.jl` incorporates in-processing approaches, such as optimization-based techniques for logistic regression and support vector machine, to directly address fairness during model training in regular and mixed models.

By accounting for data complexities and implementing various fairness-enhancing strategies, our work aims to contribute to the development of more equitable and reliable machine learning models.

Author's Contribution

This thesis is based on three submitted papers that are currently in review. In Part I of this thesis we present an extended summary of the main motivations and contributions of each paper. Part II displays the reprints of all submitted papers. The contribution of the author of this thesis to each paper is identified in the following

[JVP1] J.P. Burgard and J.V. Pamplona [Fair Mixed Effects Support Vector Machine](https://arxiv.org/abs/2405.06433). Preprint 2024. Under review. URL:<https://arxiv.org/abs/2405.06433>

The author of this thesis contributed to the theoretical development and practical implementation of the presented model. He was the primary writer and oversaw algorithm implementation under the guidance of his advisor. The primary motivation for fair support vector machine in mixed models was provided by Jan Pablo Burgard.

[JVP2] J.P. Burgard and J.V. Pamplona [Fair Generalized Linear Mixed Models](https://arxiv.org/abs/2405.09273). Preprint 2024. Under review. URL:<https://arxiv.org/abs/2405.09273>

The author of this thesis provided substantial contributions, including the development of theoretical results and the implementation of algorithms. The model was a product of collaborative research efforts, with the author taking the lead in writing and overseeing the technical aspects. The primary motivation for fair generalized linear mixed models was provided by Jan Pablo Burgard.

[JVP3] J.P. Burgard and J.V. Pamplona [FairML: A Julia Package for Fair Classification](https://arxiv.org/abs/2412.01585). Preprint 2024. Under review. URL:<https://arxiv.org/abs/2412.01585>

The author of this thesis developed the techniques, and theoretical foundations. These contributions were shaped through collaborative discussions with his advisor. The primary impetus for creating a Julia package for fair classifications was provided by Jan Pablo Burgard.

Contents

I	Extended Summary	1
1	Introduction	2
1.1	Fair Machine Learning and Mixed Models	2
1.2	Contributions and Organization	3
2	Machine Learning for fair classification and Mixed Models	5
2.1	Fairness Concepts	5
2.2	Mixed Models Concepts	11
3	Support Vector Machine	15
3.1	Fair Support Vector Machine	16
3.2	Fair Mixed Effects Support Vector Machine	18
3.3	Simulation Study	22
4	Logistic Regression	28
4.1	Fair Logistic Regression	29
4.2	Fair Generalized Linear Mixed Models based on boosting	30
4.3	Cluster-Regularized Logistic Regression	32
4.4	Numerical Results	33
5	FairML: A Julia Package for Fair Classification	37
5.1	Preprocessing	40
5.2	Post-processing	43
6	Conclusions and outlook	48
	Bibliography	50
II	Reprints of the Scientific Papers	56
	Fair Mixed Effects Support Vector Machine	57
	Fair Generalized Linear Mixed Models	75
	FairML: A Julia Package for Fair Classification	101

Part I

Extended Summary

Chapter 1

Introduction

1.1 Fair Machine Learning and Mixed Models

Machine Learning, a powerful tool for decision-making, merges computational techniques and mathematical models to uncover hidden patterns within datasets. These algorithms can then be used to make informed classifications, including classifying data into specific categories. One strategy for developing Machine Learning algorithms involves solving optimization problems (Carrizosa et al. 2022, 2023a,b). This process entails finding the optimal parameters that minimize a given loss function, for example. With a good optimization model and using appropriate solvers (Gurobi Optimization, LLC 2024; Wächter and Biegler 2006), we can find these parameters, leading to accurate classifications.

The increasing reliance on automatic decision-making processes should conform to societal limitations, such as prohibitions against discrimination of parts of the population. Automated decision systems necessitates a parallel increase in the development of fair algorithms since this principle of treating all individuals equally is a keystone of democratic governance (Barocas et al. 2017; Habermas 2003). Machine learning algorithms, while offering efficiency, can inadvertently perpetuate unfairness in critical areas like loan approvals (Das et al. 2021) and criminal justice (Green 2018). In loan applications, factors like marital status can lead to unfair disadvantages for single individuals, while in criminal justice, algorithms might associate race with recidivism risk, leading to discriminatory sentencing despite individual circumstances. This highlights the need for fair AI frameworks to ensure equal opportunities and outcomes for all.

Motivated by the need to mitigate algorithmic discrimination, researchers are increasingly focusing on fair classification methods. This has led to a proliferation of studies investigating novel approaches to achieve equitable results. Notable examples include fair versions of Logistic and Linear Regression (Berk et al. 2017;

Do et al. 2022), Support Vector Machine (Olfat and Aswani 2017), Random Forests (Zhang et al. 2021) and Decision Trees (Aghaei et al. 2019). These methods aim to address potential discrimination arising from historical data or algorithmic design, ensuring fairer outcomes for all individuals.

A major hurdle in developing machine learning models for automated decision-making lies in the quality of training data. Frequently gathered through surveys, this data often doesn't adhere to the fundamental machine learning assumption that all elements are sampled independently with equal probability of inclusion. Moreover, some data may suffer from cluster effects, e.g., in marketing, customers who buy a particular product might also be interested in a similar product. Ignoring these effects can lead to misleading results, such as underestimating the true variability in the data. To overcome it, random effects are incorporated into the statistical model, which, together with the existing fixed effects, leads to a mixed effects model, as can be seen in Oberg and Mahoney (2007).

1.2 Contributions and Organization

This cumulative thesis comprises three research papers aimed at resolving the issue of unfairness in classifications, specifically considering the scenario where data exhibits random effects. As a first contribution, we propose a Mixed Effects Support Vector Machine for fair classifications. We demonstrate how to estimate the model and assess its performance relative to the current model, which does not account for potential data clustering. Similar approaches for longitudinal data can be found in Hu et al. (2022), for Least-squares support vector machine in Cheng et al. (2014), and with applications in agriculture in Srivastava et al. (2016).

The second contribution is a fair Generalized Linear Mixed Model based on boosting. We demonstrate how the method proposed in Tutz and Groll (2010) can be adapted to incorporate fairness constraints. We also introduce cluster-regularized logistic regression and its fair variant, addressing the same problem as the previous approach but through an optimization problem.

The third contribution pertains to a `Julia` package called `FairML.jl` for fair classification. The `Julia` programming language has been growing increasingly, especially in the field of machine learning. One reason is the availability of robust tools for optimization problems (Berman and Ginesin 2024). For this reason, a package for fair classification in `Julia` that takes into account an optimization problem adds value to the academic community. Our propose integrates with the established `MLJ.jl` package (Blaom et al. 2019), enabling the utilization of all classification methods therein. In this context, we introduce a novel resampling method for preprocessing data with the objective of mitigating disparate impact or disparate mistreatment. We also model various optimization problems, both

previously proposed in the literature and others originally formulated by us to address unfairness metrics in regular and mixed effects models. Finally, we develop a cross-validation-based post-processing method to determine an optimal cut-off value for the fair classification process.

This extended summary is organized as follows. In Chapter 2, we introduce several fairness concepts, including metrics and the theory behind the creation of the fairness constraints utilized in this thesis. We also provide an explanation of how mixed models works and the adaptations created for the fairness constraints to be able to handle this additional problem. Chapters 3 and 4 provide a comprehensive overview of the literature advancements in our work. Chapter 3 specifically addresses the support vector machine algorithm and a one-hot encoding alternative for big data, while Chapter 4 presents the advancements made in logistic regression and generalized linear mixed models.

In Chapter 5, we provide a comprehensive documentation related to `FairML.jl` package, including the classification process and all potential user applications. We conclude the first part of this thesis in Chapter 6.

Finally, in Part II we present the reprints of the three papers.

Chapter 2

Machine Learning for fair classification and Mixed Models

Binary classification algorithms in machine learning are used to estimate a specific label $\hat{y} \in \{-1, 1\}$ for a new data point x based on a training set $\mathcal{D} = (x^\ell, y_\ell)_{\ell=1}^n$, with n being the number of points. For the point $x^\ell \in X = [x^1, \dots, x^n]$, if $y_\ell = 1$, we say that x^ℓ is in the positive class and if $y_\ell = -1$, x^ℓ belongs to the negative class for each $\ell \in [1, n] := \{1, \dots, n\}$. Moreover, $x^\ell \in \mathbb{R}^{p+1}$, for each $\ell \in [1, n]$, due to the addition of an extra column with the value 1 as the data intercept.

Training a model typically involves solving an optimization problem. This process is done using a objective function such as a loss function.

In recent years, there has been a growing interest in developing fair machine learning algorithms. Fairness is a complex concept, but it generally refers to the idea that algorithms should not discriminate against certain categories of people (Caton and Haas 2020). This is important because algorithms are increasingly being used to make decisions about people's lives, such as whether to grant them a loan or admit them to college.

2.1 Fairness Concepts

When aiming for fairness in binary classification, we balance achieving good classifications with ensuring fairness for observations $\ell \in [1, n]$ based on their sensitive feature $s_\ell \in \{0, 1\}$. Fairness in machine learning can be evaluated using various metrics as demographic parity (Jiang et al. 2022), false positive rate difference (Long 2021), false negative rate difference (Mijalkovic and Spognardi 2022), equal

opportunity (Wang et al. 2023) but here, we focus on two specific ones: disparate impact (DI) and disparate mistreatment (DM) that are proposed by Barocas and Selbst (2016) and Zafar et al. (2019) respectively, with the last being a combination of the fairness metrics of false positive rate and false negative rate.

After have the coefficients (solution of a optimization problem, per example) we can examining the classifications (compared to true labels) allowing us to categorize the data into four groups. A point is classified as true positive (TP) or true negative (TN) if its predicted class (positive or negative, respectively) matches its true label. On the other hand, points are classified as false positives (FP) or false negatives (FN) if their predicted class differs from the true label. Within this classification framework, accuracy becomes a key metric. Higher values indicate better classification performance. The formula for accuracy is as follows:

$$AC := \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1].$$

Now, we present two fairness metrics.

Disparate Impact

Disparate impact refers to a situation where the classification of a model disproportionately harm points with different sensitive feature values. That is, a classifier is considered fair with respect to disparate impact if the probability, of the model (\mathbb{P}_m), of its classification remains constant across both values of the sensitive feature s , i.e.,

$$\mathbb{P}_m(\hat{y}_\ell = 1 | s_\ell = 0, X) = \mathbb{P}_m(\hat{y}_\ell = 1 | s_\ell = 1, X). \quad (1)$$

To compute the disparate impact of a specific sensitive feature s consider:

$$\begin{aligned} \mathcal{S}_1 &= \{x^\ell : \ell \in [1, n], s_\ell = 1\}, & \mathcal{S}_0 &= \{x^\ell : \ell \in [1, n], s_\ell = 0\}, \\ \mathcal{P} &= \{x^\ell : \ell \in [1, n], y_\ell = 1\}, & \mathcal{N} &= \{x^\ell : \ell \in [1, n], y_\ell = -1\}, \\ \mathcal{D}_0^{\mathcal{P}} &= \mathcal{S}_0 \cap \mathcal{P}, & \mathcal{D}_0^{\mathcal{N}} &= \mathcal{S}_0 \cap \mathcal{N}, \\ \mathcal{D}_1^{\mathcal{P}} &= \mathcal{S}_1 \cap \mathcal{P}, & \mathcal{D}_1^{\mathcal{N}} &= \mathcal{S}_1 \cap \mathcal{N}. \end{aligned} \quad (2)$$

Defining \mathcal{S}_0 and \mathcal{S}_1 as disjoint subsets of dataset X , where the sensitive feature of all points in this subset is 0 and 1, respectively and \mathcal{P} and \mathcal{N} being the subsets where the true labels of the training set \mathcal{D} are positive and negative, respectively. Then, we have the following disparate impact metric, based on Radovanović et al. (2020):

$$di := \frac{|\{\ell : \hat{y}_\ell = 1, x^\ell \in \mathcal{S}_0\}|}{|\mathcal{S}_0|} \frac{|\mathcal{S}_1|}{|\{\ell : \hat{y}_\ell = 1, x^\ell \in \mathcal{S}_1\}|} \in [0, \infty).$$

Note that di is the ratio between the proportion of points in \mathcal{S}_0 classified as positive and the proportion of points in \mathcal{S}_1 classified as positive. Hence disparate Impact, as a metric, should ideally be equal to 1 to indicate fair classifications. Values greater or lower than 1 suggest the presence of unfairness. For instance, both $\text{di} = 2$ and $\text{di} = 0.5$ represent the same amount of discrimination, but in opposite directions. To address this limitation and achieve a more nuanced metric, we use the minimum value between the di and its inverse $\frac{1}{\text{di}}$. Furthermore, to align with the convention of other fairness metrics where a value closer to 0 indicates greater fairness (as will be show later), we redefine the DI as follows

$$\text{DI} := 1 - \min(\text{di}, \text{di}^{-1}) \in [0, 1]. \quad (3)$$

Hence, a value closer to 0 indicates better performance and a value closer to 1 indicates worse performance.

As stated in Equation (1), to ensure a classification is free from disparate impact, a point should have an equal probability of being classified as 1, whether it has the sensitive feature 0 or 1. In other words, we must maintain a proportional relationship between the classifications for both classes of the sensitive feature.

Since $s_\ell \in \{0, 1\}$, we obtain the mean of s being:

$$\frac{\sum_{\ell=1}^n s_\ell}{n} = \frac{|\mathcal{S}_1|}{n} =: \bar{s}. \quad (4)$$

As classification in this work is primarily based on the inner product $\beta^\top x$ with $\beta \in \mathbb{R}^{p+1}$, as will be seen in the next chapters. To maintain the proportionality of classifications for both sensitive categories, we want that:

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{S}_1} \beta^\top x^\ell = \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{S}_0} \beta^\top x^\ell. \quad (5)$$

This means that

$$\begin{aligned} 1 - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{S}_1} \beta^\top x^\ell &= \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{S}_0} \beta^\top x^\ell \\ \implies (1 - \bar{s}) \sum_{x^\ell \in \mathcal{S}_1} \beta^\top x^\ell &= \bar{s} \sum_{x^\ell \in \mathcal{S}_0} \beta^\top x^\ell \\ \implies \sum_{x^\ell \in \mathcal{S}_1} (1 - \bar{s}) \beta^\top x^\ell &= \sum_{x^\ell \in \mathcal{S}_0} (\bar{s} - 0) \beta^\top x^\ell \\ \implies \sum_{x^\ell \in \mathcal{S}_1} (1 - \bar{s}) \beta^\top x^\ell + \sum_{x^\ell \in \mathcal{S}_0} (0 - \bar{s}) \beta^\top x^\ell &= 0 \end{aligned}$$

$$\implies \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) = 0.$$

To maintain consistency with the previously proposed literature (Zafar et al. 2017), we divide the sum above by n . Note that this does not alter the equality.

While achieving zero disparate impact is a desirable goal, it can potentially come at the expense of a good classification as we have a trade-off between fairness and accuracy (Menon and Williamson 2018; Zhao and Gordon 2022). To address this exchange, we can introduce a fairness threshold, denoted by $c \in \mathbb{R}^+$, which allows us to adjust the relative importance placed on fairness compared to accuracy. This approach formalizes the construction of our disparate impact constraints.

$$\begin{aligned} \frac{1}{n} \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) &\leq c \\ \frac{1}{n} \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) &\geq -c. \end{aligned} \tag{DI}$$

Since, we now have our disparate impact constraints we present the next fairness metric, called disparate mistreatment.

Disparate Mistreatment

Disparate mistreatment, also known as equalized odds (Hardt et al. 2016), is defined as the condition in which the misclassification rates for points with different categories of sensitive features differ. In other words, a classification is free of disparate mistreatment when the classification algorithm is equally likely to misclassify points in both positive and negative classes, regardless of their sensitive characteristics.

A classification is considered free of disparate mistreatment if the rate of false positives and false negatives is equal for both categories of a sensitive feature s . That is,

$$\mathbb{P}(\hat{y}_\ell = 1 | \ell \in \mathcal{D}_0^{\mathcal{N}}) = \mathbb{P}(\hat{y}_\ell = 1 | \ell \in \mathcal{D}_1^{\mathcal{N}})$$

and

$$\mathbb{P}(\hat{y}_\ell = -1 | \ell \in \mathcal{D}_0^{\mathcal{P}}) = \mathbb{P}(\hat{y}_\ell = -1 | \ell \in \mathcal{D}_1^{\mathcal{P}}).$$

To measure the extent of disparate mistreatment with regard to a sensitive attribute s , we first establish the equations for the false positive rate (FPR) and false negative rate (FNR) metrics. The FPR fairness metric is defined as the absolute value of the difference between the false positive rates of the categories defined by the sensitive feature s , as follows:

$$\begin{aligned}
FPR &:= |FPR_{s=0} - FPR_{s=1}| \\
&= \left| \frac{FP_{s=0}}{FP_{s=0} + TN_{s=0}} - \frac{FP_{s=1}}{FP_{s=1} + TN_{s=1}} \right| \tag{FPR} \\
&= \left| \frac{|\{\ell : \hat{y}_\ell = 1, x^\ell \in \mathcal{D}_0^{\mathcal{N}}\}|}{|\mathcal{D}_0^{\mathcal{N}}|} - \frac{|\{\ell : \hat{y}_\ell = 1, x^\ell \in \mathcal{D}_1^{\mathcal{N}}\}|}{|\mathcal{D}_1^{\mathcal{N}}|} \right| \in [0, 1].
\end{aligned}$$

Similarly, the FNR fairness metric is given by:

$$\begin{aligned}
FNR &:= |FNR_{s=0} - FNR_{s=1}| \\
&= \left| \frac{FN_{s=0}}{FN_{s=0} + TP_{s=0}} - \frac{FN_{s=1}}{FN_{s=1} + TP_{s=1}} \right| \tag{FNR} \\
&= \left| \frac{|\{\ell : \hat{y}_\ell = -1, x^\ell \in \mathcal{D}_0^{\mathcal{P}}\}|}{|\mathcal{D}_0^{\mathcal{P}}|} - \frac{|\{\ell : \hat{y}_\ell = -1, x^\ell \in \mathcal{D}_1^{\mathcal{P}}\}|}{|\mathcal{D}_1^{\mathcal{P}}|} \right| \in [0, 1],
\end{aligned}$$

Lower values (closer to 0) for both metrics signify fairer classifications, in terms of disparate mistreatment, as they imply a higher degree of similarity between the false negative rates and false positive rates across sensitive categories. In other words, the differences between these rates are minimized, approaching zero.

To formulate the constraints pertaining to disparate mistreatment, we begin by considering the FNR constraint. A point is a false negative if $y_\ell = 1$ and $\beta^\top x^\ell < 0$, that is, if and only if

$$\min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) \tag{6}$$

is greater than zero, being β the coefficient. In fact, let us examine all four possibilities:

1. True Negative: $y_\ell = -1$ and $\beta^\top x^\ell < 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = 0$.
2. False Positive: $y_\ell = -1$ and $\beta^\top x^\ell > 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = 0$.
3. False Negative: $y_\ell = 1$ and $\beta^\top x^\ell < 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = \beta^\top x^\ell$.
4. True Positive: $y_\ell = 1$ and $\beta^\top x^\ell > 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = 0$.

In view of this, Zafar et al. (2016) uses the Expression (6) to select the false negative points among all points. However, note that in the FNR constraint, we only need to care about the points that belong to \mathcal{P} , defined in (2), because for the point that belongs to \mathcal{N} the Expression (6) is always equal to 0. Moreover, for a point $x^\ell \in \mathcal{P}$ we have $y_\ell = 1$ and Expression (6) becomes $\min(0, \beta^\top x^\ell)$. Therefore, in [JVP3] we propose a version that sums only over points in \mathcal{P} . Note that it reduces

computational cost. To maintain the same proportion of false negatives in both sensitive categories, the *FNR* constraints impose that the sums of the minimum between 0 and the inner products of the coefficient and a positive point are close to each other in each sensitive category, as follows:

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^\top x^\ell) \leq c \quad (7a)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^\top x^\ell) \geq -c \quad (7b)$$

For false positive points, we employ the same logic as in the false negative points, however, replacing the Expression (6) with:

$$\min\left(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell\right).$$

This means that a point is

1. True Negative: $y_\ell = -1$ and $\beta^\top x^\ell < 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = 0$.
2. False Positive: $y_\ell = -1$ and $\beta^\top x^\ell > 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = -\beta^\top x^\ell$.
3. False Negative: $y_\ell = 1$ and $\beta^\top x^\ell < 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = 0$.
4. True Positive: $y_\ell = 1$ and $\beta^\top x^\ell > 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = 0$.

That is, in the *FPR* constraints, we only need to care of the points that belong to \mathcal{N} . Again, in [JVP3] we created constraints that sums only over points in \mathcal{N} , to reduce the computational costs. Similarly to the *FNR* constraints, the *FPR* constraints impose that the sums of the minimum between 0 and minus the inner products of the coefficient and a negative point are close to each other in each sensitive category. That is,

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) \leq c \quad (8a)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) \geq -c \quad (8b)$$

Observe that, again, in both metrics, we tolerate a fairness threshold c , which allows us to calibrate the relative importance placed on fairness compared to accuracy.

Therefore, the Disparate Mistreatment constraints is calculated as the average of these two metrics, and is given by

$$\text{DM} = \frac{FPR + FNR}{2} \in [0, 1]. \quad (\text{DM})$$

That is, the constraints of disparate mistreatment are the union of constraints (7) and (8). Again, a lower value indicates a fairer outcome.

The threshold c must be equal for both pairs of constraints when using the disparate mistreatment metric. Varying c would invalidate the mean-based calculation of the metric value. Although using different values c is possible, it essentially defines two separate fairness metrics the FPR and FNR that should be computed separately.

2.2 Mixed Models Concepts

Real-world data often shows heterogenic variations within groups. Consider the example with a dataset comprising multiple schools where we aim to ascertain whether distinct teachers yield varying outcomes for a particular variable. In a predictive modeling context, we would classify these teaching capacity variables as fixed effects, that is, effects that represent factors that are of direct interest and are explicitly modeled (Agasisti et al. 2017). However, it is plausible that other unmeasured factors, such as the quality of teaching materials or the overall school environment, also influence the outcomes.

To capture the latent heterogeneity present in these types of data, which can encompass cultural, demographic, biological, and behavioral aspects, it is imperative to incorporate the capacity to deal with random effects into the predictive model (Greene 1997). Omitting these effects can lead to substantial bias in classifications, compromising the accuracy and generalization of the results (Barili et al. 2018; Yang et al. 2014).

In [JVP1] and in [JVP2] we adapted the fairness metrics DI to account for the presence of these random effects in the SVM and logistic regression respectively. In [JVP3] we also adapted the fairness metric DM for a mixed effects model. This section presents an overview of the constraints for mixed model problems.

Let g being the random vector and g_i with $i \in [1, K]$, representing the group-specific random effect, with g following a normal distribution with mean zero. Consider Γ_i the size of the group i for each $i \in [1, K]$ and y_{ij} the label of $(x^{ij})^\top = (1, x_1^{ij}, \dots, x_p^{ij})$ with $j \in [1, \Gamma_i]$.

In essence, the fairness constraints discussed in the section above only consider fixed effects (β). However, in mixed models we need to consider the random effects g_i , that is, $\beta^\top x^{ij} + g_i$ being the inner product used to compute the probability of

the point x^{ij} being classified as 1. In light of these considerations, we propose the following adaptation to the created subgroups in Equation (2):

$$\begin{aligned}
\mathcal{S}_1^i &= \{x^{ij} : j \in [1, \Gamma_i], s_{ij} = 1\}, & \mathcal{S}_0^i &= \{x^{ij} : j \in [1, \Gamma_i], s_{ij} = 0\}, \\
\mathcal{P}^i &= \{x^{ij} : j \in [1, \Gamma_i], y_{ij} = 1\} & \mathcal{N}^i &= \{x^{ij} : j \in [1, \Gamma_i], y_{ij} = -1\}, \\
\mathcal{D}_0^{\mathcal{P}^i} &= \mathcal{S}_0^i \cap \mathcal{P}^i, & \mathcal{D}_0^{\mathcal{N}^i} &= \mathcal{S}_0^i \cap \mathcal{N}^i, \\
\mathcal{D}_1^{\mathcal{P}^i} &= \mathcal{S}_1^i \cap \mathcal{P}^i, & \mathcal{D}_1^{\mathcal{N}^i} &= \mathcal{S}_1^i \cap \mathcal{N}^i.
\end{aligned}$$

Observe that each subset is created for each cluster $i \in [1, K]$.

Moreover, we need to modify the prediction function in the fairness constraints to account for random effects. We now present the modifications for DI, as proposed in [JVP1] and [JVP2].

Disparate Impact

Following the same logic as presented in Subsection 2.1 but considering a group-to-group analysis, we have a similar construction for the disparate impact constraints:

$$\begin{aligned}
\frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (\beta^\top x^{ij} + g_i) &= \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\beta^\top x^{ij} + g_i) \\
\implies 1 - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (\beta^\top x^{ij} + g_i) &= \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\beta^\top x^{ij} + g_i) \\
\implies (1 - \bar{s}) \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (\beta^\top x^{ij} + g_i) &= \bar{s} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\beta^\top x^{ij} + g_i) \\
\implies \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (1 - \bar{s})(\beta^\top x^{ij} + g_i) &= \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\bar{s} - 0)(\beta^\top x^{ij} + g_i) \\
\implies \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (1 - \bar{s})(\beta^\top x^{ij} + g_i) - \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (0 - \bar{s})(\beta^\top x^{ij} + g_i) &= 0 \\
\implies \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} + \bar{s})(\beta^\top x^{ij} + g_i) &= 0.
\end{aligned}$$

Based on the previously mentioned justifications, the fairness constraint for dis-

parate impact that consider the mixed effects proposed in [JVP1] is given by:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) &\leq c, \\ \frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) &\geq -c. \end{aligned} \tag{MEDI}$$

Disparate Mistreatment

We now discuss the DM metric for mixed effects. For the *FNR* constraints, in [JVP3] we adapt the Expression (6) to incorporate the random effects as follows:

$$\min \left(0, \frac{1 + y_{ij}}{2} y_{ij} (\beta^\top x^{ij} + g_i) \right).$$

As done for the regular models, we only need take care about the positive points. And, for these points, the expression above become $\min(0, \beta^\top x^{ij} + g_i)$.

On the other hand, for the *FPR* constraints, the selection of the false positive points is adapted to

$$\min \left(0, \frac{1 - y_{ij}}{2} y_{ij} (\beta^\top x^{ij} + g_i) \right).$$

Here we only need take care about the negative points. And, for these points, the expression above become $\min(0, -\beta^\top x^{ij} - g_i)$. Combining all constraints, in [JVP3] we propose the following set of constraints for a classification free of disparate mistreatment in mixed models:

$$\begin{aligned} \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{P^i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{P^i}} \min(0, \beta^\top x^{ij} + g_i) &\leq c \\ \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{P^i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{P^i}} \min(0, \beta^\top x^{ij} + g_i) &\geq -c \\ \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{N^i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{N^i}} \min(0, -\beta^\top x^{ij} - g_i) &\leq c \\ \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{N^i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{N^i}} \min(0, -\beta^\top x^{ij} - g_i) &\geq -c. \end{aligned} \tag{MEDM}$$

The first summation iterates over all groups, while the second summation iterates only over the desired points within each group.

An important observation is that several sensitive features can be used in the optimization problem. Simply add a new pair of constraints to each feature, and each feature can have its individual threshold c , giving different levels of importance to them, independent of the fairness metric.

In the following chapter, we investigate the SVM problem. More specifically we briefly discuss how we adapted the problem to simultaneously address both problems, mixed effects and unfair classifications in [\[JVP1\]](#).

Chapter 3

Support Vector Machine

When data points belonging to different classes can be separated by a hyperplane, we consider the dataset to be linearly separable. This is where the goal of SVM comes into play: identify this hyperplane that effectively separates the data points into distinct classes, positive and negative, as mentioned in Chapter 2, i.e. $\hat{y}_\ell \in \{-1, 1\}$.

However, since data is not always linearly separable, we seek an alternative SVM that can handle these situations, the well-known soft-margin SVM. To achieve this, slack variables ξ_ℓ , $\ell \in [1, \dots, n]$ are introduced, which allow some misclassification. This leads to the following optimization problem, proposed by Vapnik and Chervonenkis (1964) and Hearst et al. (1998):

$$\begin{aligned} \min_{(\beta, \xi)} \quad & \frac{1}{2} \|\beta\|^2 + \mu \sum_{\ell=1}^n \xi_\ell \\ \text{s.t.} \quad & y_\ell(m_\beta^{SVM}(x^\ell)) \geq 1 - \xi_\ell, \\ & \xi_\ell \geq 0, \ell = 1, \dots, n. \end{aligned} \tag{SVM}$$

with the prediction function given by

$$m_\beta^{SVM}(x) := \beta^\top x. \tag{9}$$

It is worth noting that the bias commonly found in the formulation of Support Vector Machine problems is not explicitly stated. This is due to the fact that in our formulation the first column of X is equal to a vector of ones due to the intercept. Hence, the bias parameter is the first entry in β and this strategy can be seen in Hsieh et al. (2008).

The objective function of optimization problem (SVM) can be seen as a compromise between the maximization of the distance between the two classes and minimizing the classification error. The penalty parameter μ aims to control the

importance of slack variables, which represents the flexibility in classifying a point considering the optimal hyperplane. The constraint of problem (SVM) determines on which side of the hyperplane the labeled data point x^ℓ , $\ell \in [1, \dots, n]$ should reside.

In the following, we present how to add fairness in the context of Support Vector Machine and more details can be seen in Das et al. (2021).

3.1 Fair Support Vector Machine

Support Vector Machine are a powerful machine learning algorithm commonly used for classification tasks but it can be susceptible to discrimination because can perpetuate unfairness present in the training data. This occurs when the data reflects historical inequities in a specific population such as gender or race (Miller J 1966).

To address this issue, in the disparate impact context, Zafar et al. (2019) proposed to add the constraint of disparate impact in the SVM problem. In the case with the intercept, considered in this extended summary, this means to join the Problem (SVM) with the Equation (DI). This leads to the following optimization problem, that aims to minimize disparate impact while maintaining the classification performance of the SVM.

$$\min_{(\beta, \xi)} \quad \frac{1}{2} \|\beta\|^2 + \mu \sum_{\ell=1}^n \xi_\ell \quad (10a)$$

$$\text{s.t.} \quad y_\ell(m_\beta^{SVM}(x^\ell)) \geq 1 - \xi_\ell, \quad (10b)$$

$$\frac{1}{n} \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) \leq c, \quad (10c)$$

$$\frac{1}{n} \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) \geq -c, \quad (10d)$$

$$\xi_\ell \geq 0, \quad \ell = 1, \dots, n. \quad (10e)$$

Observe that the objective function (10a) and the constraints (10b) and (10e) are from the Support Vector Machine problem. Constraints (10c) and (10d) guarantees the fairness, in terms of disparate impact as discussed in Chapter 2 and the parameter $\mu > 0$ control the importance of ξ . In other words, a larger value of μ implies a higher penalty in the misclassifications.

In the following example, the two sensitive categories are distinguished by the shape of the point (diamond and star). The color differentiates the label, with red being positive (1) and blue being negative (-1).

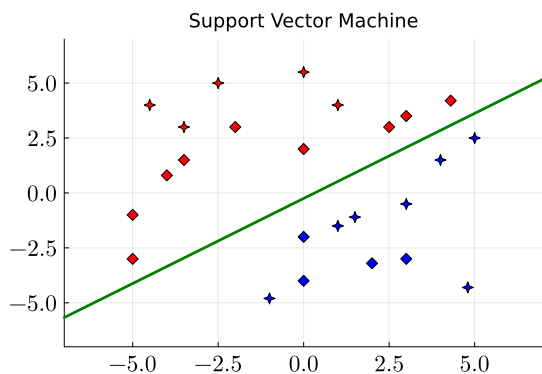


Figure 3.1: Regular SVM.

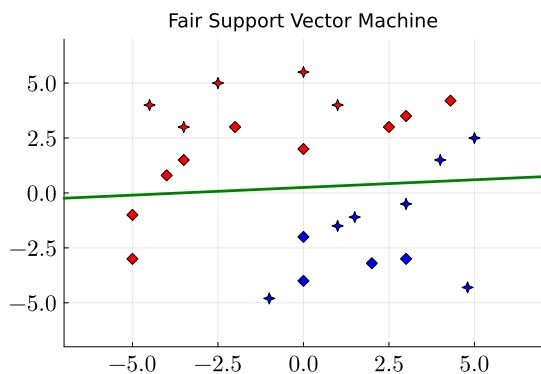


Figure 3.2: SVM free of disparate impact.

Figure 3.1 demonstrates 100% accuracy, as each point is located on the correct side of the hyperplane. However, when we look at each sensitive feature category, we observe that approximately 36%(5/14) of the star points are classified as positive while this happens for approximately 64%(9/14) of the diamond points. That is, the proportions of the two sensitive categories being classified as positive are not equal.

Figure 3.2 shows an example of SVM free of disparate impact as in Problem (10), considering $c = 0$. We can see that 50% of the star points and approximately 53% of the diamond points are classified as positive. That is, the proportion of positive classifications is similar for both sensitive categories. However this comes at the cost of accuracy, as 4 points were misclassified, reducing the accuracy to 84%. This can be mitigated by choosing a threshold c , sufficiently good so as not to significantly decrease accuracy.

Building upon the discussion in Chapter 2, we now present the optimization problem for support vector machine (SVM) that are improved in disparate mistreatment. This formulation is proposed by Zafar et al. (2019) and aims to identify a hyperplane that maximizes the margin between the two classes while ensuring equal misclassification rates for both sensitive categories to do this we join the Problem (SVM) with the Constraints (7a),(7b),(8a) and (8b). This lead to the following optimization problem:

$$\min_{(\beta, \xi)} \frac{1}{2} \|\beta\|^2 + \mu \sum_{\ell=1}^n \xi_{\ell} \quad (11a)$$

$$\text{s.t. } y_{\ell}(m_{\beta}^{SVM}(x^{\ell})) \geq 1 - \xi_{\ell}, \quad (11b)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^{\ell} \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^{\top} x^{\ell}) - \frac{|\mathcal{S}_1|}{n} \sum_{x^{\ell} \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^{\top} x^{\ell}) \leq c, \quad (11c)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^\top x^\ell) \geq -c, \quad (11d)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) \leq c, \quad (11e)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) \geq -c, \quad (11f)$$

$$\xi_\ell \geq 0, \quad \ell = 1, \dots, n. \quad (11g)$$

The objective function (11a) and the constraints (11b) and (11g) are from the Support Vector Machine problem. Constraints (11c), (11d), (11e) and (11f) guarantee the fairness, in terms of disparate mistreatment. This model can be formulated using only False Negative Rate or False Positive Rate constraints, if desired.

3.2 Fair Mixed Effects Support Vector Machine

In [JVP1], we propose an approach for a Mixed Effects Support Vector Machine, that incorporates the assumption of random effects following a normal distribution with mean zero. This method specifically considers the fact that each data point belongs to a single group. Consequently, the hyperplane solution takes into account both fixed effects, that consider all data points, and random effects, generated only by the points in each particular group. Given this, the optimization problem for the mixed effects support vector machine (MESVM) proposed in [JVP1] is the following:

$$\begin{aligned} \min_{(\beta, b, \xi)} \quad & \frac{1}{2} \|\beta\|^2 + \mu \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \xi_{ij} + \lambda \sum_{i=1}^K g_i^2 \\ \text{s.t.} \quad & y_{ij}(m_{\beta, g}^{SVM}(x^{ij})) \geq 1 - \xi_{ij} \\ & \xi_{ij} \geq 0, \quad i = 1, \dots, K, \quad j = 1, \dots, \Gamma_i, \end{aligned} \quad (\text{MESVM})$$

with the prediction function given by

$$m_{\beta, g}^{SVM}(x^{ij}) := \beta^\top x_{ij} + g_i \quad (12)$$

The final term in the objective function of Problem (MESVM) penalizes the variance of the random effects through L2 regularization, aiming to minimize it. As $\lambda > 0$ increases, the problem increasingly prioritizes the minimization of the random effects variance. Similar approaches for least-square support vector machine can be found in Luts et al. (2012) and as demonstrated in Domingos (2012), Friedman

et al. (2010), and Zou and Hastie (2005), minimizing the variance enhances model performance and generalization. High variance often implies that the model is excessively sensitive to the training data, resulting in sub-optimal performance on unseen data. Hence the sum $\sum_i^K g_i^2$ is from the fact that the variance of g is given by:

$$\sigma^2 = \frac{\sum_{i=1}^K (g_i - \bar{g})^2}{K - 1},$$

and the random effect g follows a normal distribution with mean zero.

$$\sigma^2 = \frac{\sum_{i=1}^K (g_i - 0)^2}{K - 1} = \frac{\sum_{i=1}^K g_i^2}{K - 1}.$$

This means that our regularization term is $\frac{\sum_{i=1}^K g_i^2}{K-1}$. Since we have a minimization problem and $K - 1$ is fixed, we can consider w.l.o.g a parameter λ that controls the importance of the random effects, as already mentioned.

In our numerical experiments, we set variables μ and λ equal to 1. The choice of μ reflects our desire to assign equal importance to maximizing the margin and minimizing the classification error. The value of λ is determined by our aim of balancing the influence of random and fixed effects within the optimization problem. Preliminary tests revealed that setting λ significantly higher than 1 tends to prioritize random effects, resulting in a random vector with excessive variance, even when our objective is to minimize it. On the other hand, if λ is considerably lower than 1, the optimization process becomes overly focused on fixed effects, leading to random effects that are almost negligible.

A geometric representation of what the solution to problem (MESVM) generates can be seen in Figure 3.3.

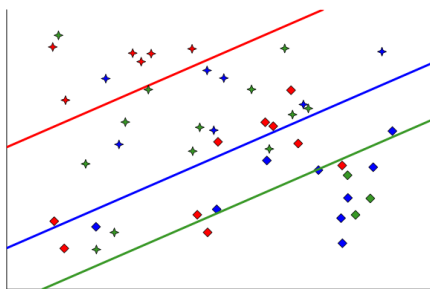


Figure 3.3: Mixed Effects Support Vector Machine.

We can see that each group receives its own separating hyperplane, with all the hyperplanes being parallel, considering that their inclination is given by the fixed effects, but they cut the Y-axis in different positions, positions that are associated

with the random effects. We can also see Figure 3.4 as a separation of the data by groups, with each group considering its own hyperplane.

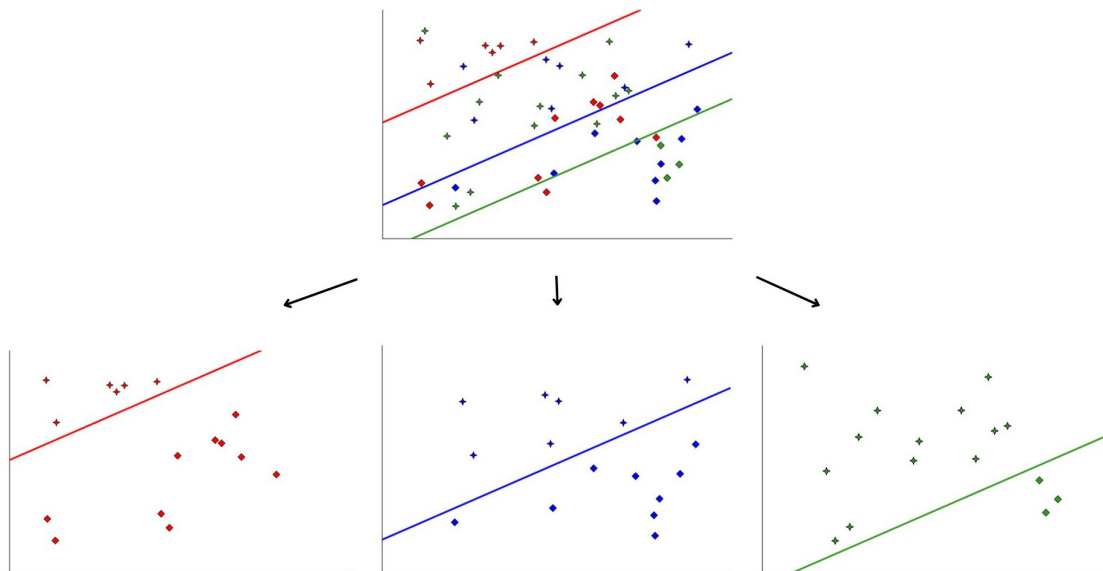


Figure 3.4: Mixed Effects Support Vector Machine.

Moreover, already discussed in Section 2.2, fairness constraints can also be adapted to account mixed effects. In [JVP1] we also propose the Mixed Effects SVM free of disparate impact, that combines (MESVM) and the Constraints (MEDI), leading to

$$\min_{(\beta, b, \xi)} \frac{1}{2} \|\beta\|^2 + \mu \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \xi_{ij} + \lambda \sum_{i=1}^K g_i^2 \quad (13a)$$

$$\text{s.t. } y_{ij}(m_{\beta, g}^{SVM}(x^{ij})) \geq 1 - \xi_{ij}, \quad (13b)$$

$$\frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) \leq c, \quad (13c)$$

$$\frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) \geq -c, \quad (13d)$$

$$\xi_{ij} \geq 0, \quad i = 1, \dots, K, \quad j = 1, \dots, \Gamma_i, \quad (13e)$$

The objective function (13a) and the constraints (13b) and (13e) are a adapted version of Support Vector Machine problem to deal with random effects. Constraints (13c) and (13d) guarantees the fairness, in terms of disparate impact.

In [JVP3], we propose the Mixed Effects Support Vector Machine free of disparate mistreatment, join the Problem (MESVM), and the Equations (MEDM) we have the following optimization problem:

$$\begin{aligned}
\min_{(\beta, b, \xi)} \quad & \frac{1}{2} \|\beta\|^2 + \mu \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \xi_{ij} + \lambda \sum_{i=1}^K g_i^2 \\
\text{s.t.} \quad & y_{ij}(m_{\beta, g}^{SVM}(x^{ij})) \geq 1 - \xi_{ij}, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{P_i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{P_i}} \min(0, \beta^\top x^{ij} + g_i) \leq c, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{P_i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{P_i}} \min(0, \beta^\top x^{ij} + g_i) \geq -c, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{N_i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{N_i}} \min(0, -\beta^\top x^{ij} - g_i) \leq c, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{N_i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{N_i}} \min(0, -\beta^\top x^{ij} - g_i) \geq -c, \\
& \xi_{ij} \geq 0, \quad i = 1, \dots, K, \quad j = 1, \dots, \Gamma_i.
\end{aligned} \tag{14}$$

Below, we show how, in addition to fairness constraints, the proposed problems can be used as an alternative to one-hot encoding.

A One-hot encoding alternative

One alternative approach to incorporating group effects into the data analysis involves the use of dummy variables, also known as one-hot encoding as can be seen in Fernandes et al. (2022). This strategy involves creating a separate binary variable for each group, taking a value of 1 for observations within that group and 0 otherwise. While this method can effectively capture group-level variation, preliminary numerical results in [JVP1] have demonstrated that our proposed approach (MESVM) offers significant advantages in terms of both computational efficiency and memory usage. This happens because, in the one-hot encoding, the dimension of each x^ℓ is increased in the number K of groups, that is, $x \in \mathbb{R}^{p+1+K}$.

In [JVP1] we consider the following preliminary numerical experiments: we create a dataset consisting of 100,000 points with the training set having 3 to 5 points per group and systematically varying the number of groups within the data,

considering configurations with 2, 10, 50, 500, 1000, 1250, 2000, 2500, 3125, 4000, and 5000 groups.

Figures 3.5 shows the Memory comparison in Bytes and Figure 3.6 the time comparison. The first row of each figure present a second-order polynomial fit, where the x-axis represents the number of groups and the y-axis the memory usage. The second row present the Performance Profile proposed by Dolan and Moré (2002) of both approaches. In this specific numerical test, the problem (13a)-(13e) is the Mixed Effects Support Vector Machine free of Disparate Impact and is labeled as FMESVM.

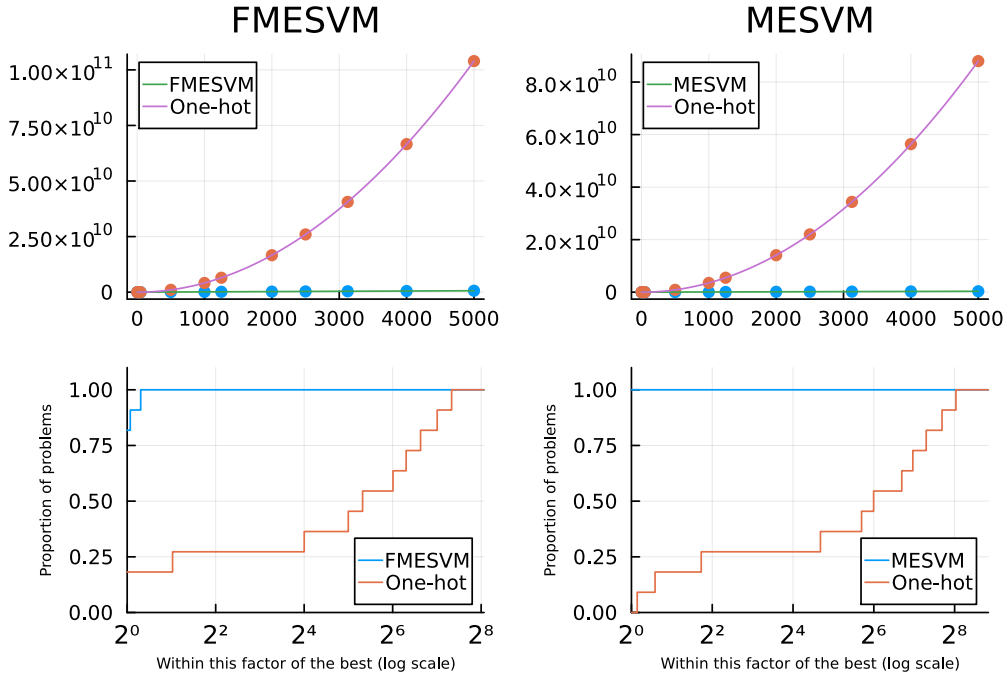


Figure 3.5: Memory usage comparison in Bytes.

As can be seen in Figures 3.5 and 3.6, our optimization models outperforms the one-hot encoding in time and memory. Hence, our approach is a more resource-friendly option, mainly for large datasets and complex models in both situations where the datasets present unfairness and when they do not. While one-hot encoding can be applied to SVM, our model is a better choice.

3.3 Simulation Study

In this Section we present some numerical results present in [JVP3] that show the advantage of the mixed effects SVM models that take into account the random

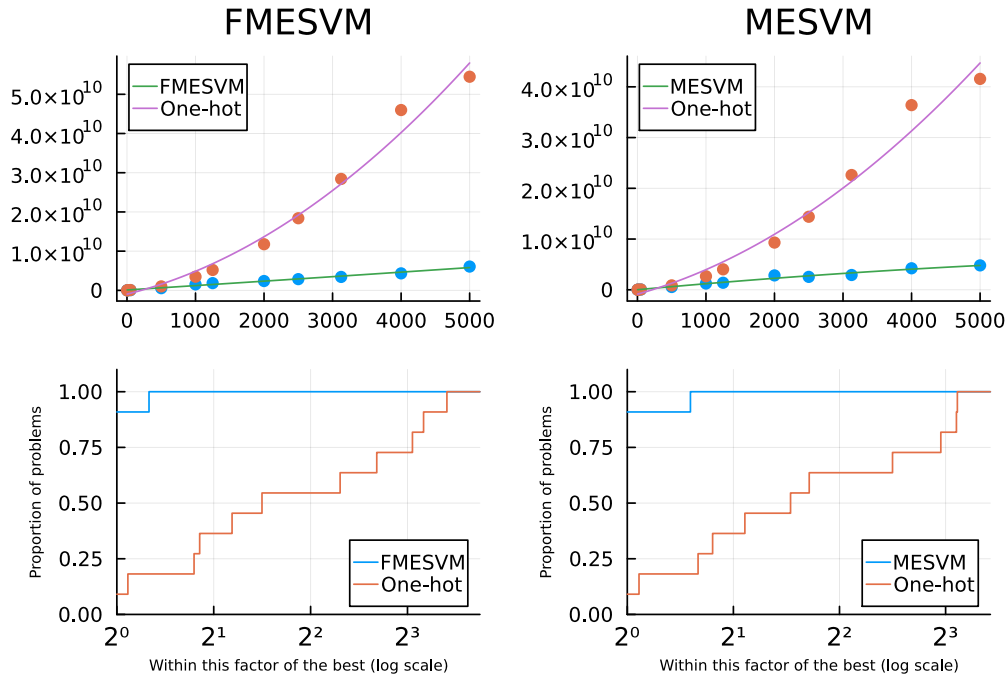


Figure 3.6: Time comparison in Microseconds.

effects but also the unfairness. First we present the step-by-step strategy used to create the datasets and to conduct the numerical experiments. Using the Julia 1.9 (Bezanson et al. 2017) language with the packages `Distributions` (Besançon et al. 2021), `DataFrames` (Bouchet-Valat and Kamiński 2023), `MLJ` (Blaom et al. 2019) and `MKL` (Corporation 2023), we generate the following parameters:

- *Number of points*: Number of points in the dataset (100,000);
- *β 's*: The fixed effects;
- *g 's*: The random effects with distribution $N(0, 3)$, if necessary;
- *Data points*: The covariate vector associated with fixed effects with distribution $N(0, 1)$;
- *c* : Threshold from fairness constraints;
- *seed*: Random seed used in the generation of data;
- *Train-Test split*: Approximately 1% of the dataset was used for the training set, and 99% for the test set. This percentage was due to the fact that we randomly selected 8 to 12 points from each stratum for the training set.

To verify the algorithm’s suitability for real-world applications, the tests were done with a very small training set (1% of 100,000 points). This approach is valid because even small percentages can yield substantial size training data in Big Data. Similar strategies are employed in Ma et al. (2025).

Then the true classifications of the synthetic dataset were computed using Expression (12) in tests where the dataset wants consider random effects in the creation process, and Expression (9) in the tests where the dataset wants consider just the fixed effects in the creation process. Since $m_{\beta}^{SVM}, m_{\beta,g}^{SVM} \in [-\infty, \infty]$, we project the value to $[0, 1]$ using the functions:

$$M = \frac{\exp(m_{\beta}^{SVM})}{1 + \exp(m_{\beta}^{SVM})}, \quad M = \frac{\exp(m_{\beta,g}^{SVM})}{1 + \exp(m_{\beta,g}^{SVM})}. \quad (15)$$

Finally,

$$y = \text{Bernoulli}(M).$$

Thus, our datasets are ready. Regarding the tests, the comparisons present in [JVP3] are the following metrics:

- Accuracy (AC);
- Disparate Impact (DI);
- False Negative Rate (FNR);
- False Positive Rate (FPR);
- True Negative Rate (TNR);
- True Positive Rate (TPR);
- Disparate Mistreatment (DM).

We compare the following optimization problems:

- Regular Support vector Machine (SVM);
- Support vector Machine free of Disparate Impact (10);
- Support vector Machine free of Disparate Mistreatment (11);
- Mixed Effects Support vector Machine (MESVM);
- Mixed Effects Support vector Machine free of Disparate Impact (13);
- Mixed Effects Support vector Machine free of Disparate Mistreatment (14).

To compute the metrics, we need compute the labels given by each approach for each point in the dataset. For that we use Equation (15) with,

$$\hat{y} = \begin{cases} 1 & \text{if } M \geq 0.5 \\ -1 & \text{if } M < 0.5 \end{cases}.$$

The parameters for the unfair cases are:

- β 's = $[-2.0; 0.4; 0.8; 0.5; 2.0]$;
- $c = 10^{-3}$;
- g 's: 100 clusters with $g_i \sim N(0, 2)$,
with $i \in [1, 100]$;
- $K = 1$;
- $\lambda = 1$.

The β_0 is the intercept, and β_4 is the coefficient associated to the binary sensitive feature. In the unfair cases the coefficient was randomly selected using numbers between 0 and 1, except for β_0 and β_4 . The reason for this is that we assign a high value to β_4 , to give more importance to the sensitive variable in the label prediction process. In other words, data points with the sensitive categories equal to 1 are more likely to be classified as positive. This practice results in a dataset that is inherently unfair in terms of both disparate impact and disparate mistreatment, as needed to test our methods. For all experiments, the matrix X was randomly generated from a multivariate normal distribution with zero mean and independent variables. Using the generated coefficients, we employed Prediction function (9) to obtain the labels.

Additional numerical tests, including fair cases or scenarios without mixed effects, can be found in [JVP1].

Before delving into the results of our proposed models, let us examine the performance of the regular models suggested by Zafar et al. (2016). This also provide us a baseline for our subsequent tests involving mixed models.

Regular Models

In [JVP3], 100 datasets were generated. Each of them was tested on all optimization problems and the boxplots (McGill et al. 1978) were constructed for each fairness metric.

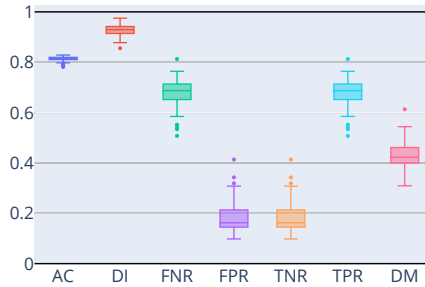


Figure 3.7: Regular Support Vector Machine.

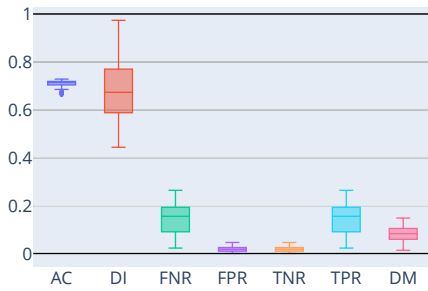


Figure 3.8: Support Vector Machine free of Disparate Impact.

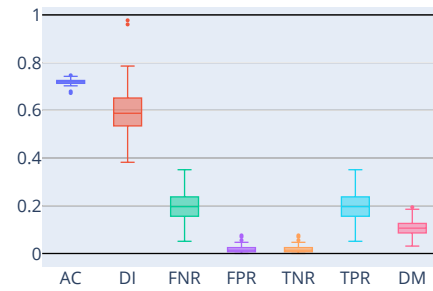


Figure 3.9: Support Vector Machine free of Disparate Mistreatment.

We can verify that, when compared to tests without fairness constraints, in Figures 3.7, the fair optimization problems, in Figures 3.8 and 3.9 can effectively reduce the fairness metrics they are designed to mitigate, even at a minor cost to accuracy.

Mixed Models

Following a similar methodology to the regular models, 100 datasets were generated. Each dataset was subjected to testing on all optimization problems, and box plots were constructed to visualize the distribution of each fairness metric.

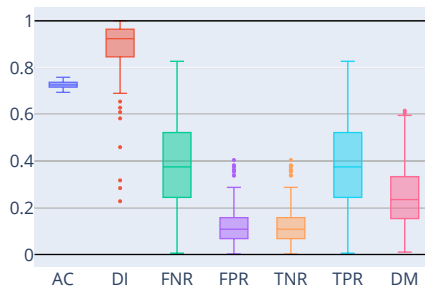


Figure 3.10: Mixed Effects Support Vector Machine.

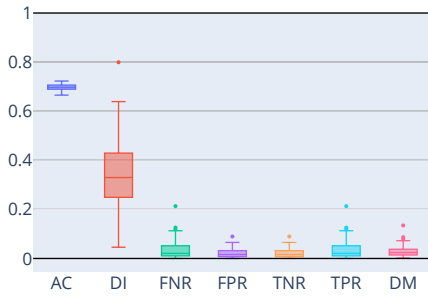


Figure 3.11: Mixed Effects Support Vector Machine free of Disparate Impact.

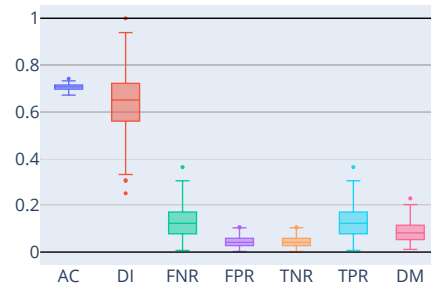


Figure 3.12: Mixed Effects Support Vector Machine free of Disparate Mistreatment.

Compared to mixed effects SVM without fairness constraints in Figures 3.10, we can see in Figures 3.11 and 3.12 that this set of experiments confirms that fair optimization problems successfully improve the fairness metrics they were designed to address, while also managing to handle the bias related to random effects. Given that the data was generated with random effects, the accuracy would be significantly lower if the optimization problem were unable to handle them.

In [JVP1], the strategies described herein are tested using the Adult dataset. To evaluate the method’s efficiency, we created groups based on individuals age and marital status, and a sensitive feature, gender (Speicher et al. 2018) that suffers from disparate impact. The Adult dataset is a famous dataset in machine learning, where the goal is predicting whether the individuals earn more ($y = 1$) or less ($y = -1$) than 50000 USD annually.

In [JVP2] we adapt these approaches for logistic regression and some important details are discussed in the next chapter.

Chapter 4

Logistic Regression

Even when employing the soft margin strategy, effectively separating the data points using a hyperplane, as SVM does, can be challenging. In this cases other approaches such as logistic regression can be used to solve the classification problem. In [JVP2] we propose to incorporate the disparate impact constraint in a generalized linear mixed models based on boosting (GLMM) and a cluster-regularized logistic regression (CRLR). In [JVP3] we adapt the disparate mistreatment constraints in the optimization problems.

Consequently, the next step in our research involved extending the previously mentioned findings to the well-established logistic regression method. To ensure that in all of our optimization problems we have $y \in \{-1, 1\}$, we adapt, w.l.o.g., the logistic regression model proposed by Neter et al. (2004).

It is crucial to emphasize that the modification of logistic regression to accommodate $y \in \{-1, 1\}$ was carried out exclusively in this thesis, while, in [JVP2], the regular logistic regression ($y \in \{0, 1\}$) was used. Hence, the adapted logistic regression problem is given by

$$\min_{\beta} - \sum_{\ell=1}^n \left[\left(\frac{y_{\ell} + 1}{2} \right) \log(m_{\beta}^{LR}(x^{\ell})) + \left(\frac{y_{\ell} - 1}{2} \right) \log(1 - m_{\beta}^{LR}(x^{\ell})) \right] \quad (\text{LR})$$

with the probability of a point be classified as 1 given by

$$m_{\beta}^{LR}(x) := \frac{1}{1 + e^{-\beta^{\top} x}}. \quad (16)$$

The loss function in Problem (LR) quantifies the discrepancy between the predicted probabilities and the actual class labels. The goal of logistic regression is to minimize this function, effectively reducing the overall error and improving the model's ability to accurately classify data points.

In the following, we present adaptations for logistic regression that are free of disparate impact and free of disparate mistreatment based on the ones proposed by Zafar et al. (2019).

4.1 Fair Logistic Regression

To generate a classification that is free of disparate impact Zafar et al. (2017) joined the Problem (LR) with the constraints (DI). This leads to the following optimization problem, that aims to minimize disparate impact while maintaining the classification performance of the Logistic Regression.

$$\min_{\beta} - \sum_{\ell=1}^n \left[\left(\frac{y_{\ell} + 1}{2} \right) \log(m_{\beta}^{LR}(x^{\ell})) + \left(\frac{y_{\ell} - 1}{2} \right) \log(1 - m_{\beta}^{LR}(x^{\ell})) \right] \quad (17a)$$

$$\text{s.t.} \quad \frac{1}{n} \sum_{\ell=1}^n (s_{\ell} - \bar{s})(\beta^{\top} x^{\ell}) \leq c, \quad (17b)$$

$$\frac{1}{n} \sum_{\ell=1}^n (s_{\ell} - \bar{s})(\beta^{\top} x^{\ell}) \geq -c, \quad (17c)$$

where s is the sensitive feature and s_{ℓ} is defined in (4).

For the disparate mistreatment problem we add the adapted constraints (7) and (8), that can be seen in [JVP3], in Problem (LR), leading to:

$$\begin{aligned} \min_{\beta} & - \sum_{\ell=1}^n \left[\left(\frac{y_{\ell} + 1}{2} \right) \log(m_{\beta}^{LR}(x^{\ell})) + \left(\frac{y_{\ell} - 1}{2} \right) \log(1 - m_{\beta}^{LR}(x^{\ell})) \right] \\ \text{s.t.} & \quad \frac{|\mathcal{S}_0|}{n} \sum_{x^{\ell} \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^{\top} x^{\ell}) - \frac{|\mathcal{S}_1|}{n} \sum_{x^{\ell} \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^{\top} x^{\ell}) \leq c, \\ & \quad \frac{|\mathcal{S}_0|}{n} \sum_{x^{\ell} \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^{\top} x^{\ell}) - \frac{|\mathcal{S}_1|}{n} \sum_{x^{\ell} \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^{\top} x^{\ell}) \geq -c, \quad (18) \\ & \quad \frac{|\mathcal{S}_0|}{n} \sum_{x^{\ell} \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^{\top} x^{\ell}) - \frac{|\mathcal{S}_1|}{n} \sum_{x^{\ell} \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^{\top} x^{\ell}) \leq c, \\ & \quad \frac{|\mathcal{S}_0|}{n} \sum_{x^{\ell} \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^{\top} x^{\ell}) - \frac{|\mathcal{S}_1|}{n} \sum_{x^{\ell} \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^{\top} x^{\ell}) \geq -c, \end{aligned}$$

To incorporate random effects into our model, we studied a pre-existing algorithm that solves the mixed effects problem using the boosting method (Tutz and Groll 2010). In the next section, we provide an overview of this algorithm

and explain how we adapted it to make fairer classifications in terms of disparate impact.

4.2 Fair Generalized Linear Mixed Models based on boosting

Generalized Linear Models (GLMs) (Hastie and Pregibon 2017) are a class of models that can be used to model a variety of response variables, including continuous and binary. Ignoring the mixed effects we can state that the logistics regression is a special case of GLMs. Following the same idea, but with some changes, the Generalized Linear Mixed Models (GLMM) (Stroup 2012) allows the inclusion of random effects. However, GLMMs, like many other statistical models, can lead to unfair outcomes.

Please note that in [JVP2], b defines the group effect vector. In this thesis, it has been changed to g to maintain consistency with all other methods.

In [JVP2] we propose a Fair GLMM that is a type of a Fischer-scoring algorithm based on Newton's method (Lange 2002). Hence, it is not suitable for optimization problems with constraints, such as the fair ones mentioned in the previously chapters. Therefore, it is necessary to convert the Logistic Regression free of disparate impact, that is a constrained optimization problem into an unconstrained one using Lagrange's penalty that can be seen in Nocedal (2006).

To achieve this, we first adapt Problem (17) to incorporate the groups, as will be shown in detail in Problem (21), and then fix $c = 0$, resulting in a problem with a unique constraint. Observe that this is not a problem since we can control the constraint with the Lagrange multiplier ρ allowing a penalized violation of it. So, we have the problem:

$$\min_{\beta, g} - \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \left[\left(\frac{y_{ij} + 1}{2} \right) \log(m_{\beta, g}^{LR}(x^{ij})) + \left(\frac{y_{ij} - 1}{2} \right) \log(1 - m_{\beta, g}^{LR}(x^{ij})) \right] + \lambda \sum_{i=1}^K g_i^2 + \frac{\rho}{n} \|a^\top \delta\|_2^2 \quad (19)$$

with

$$m_{\beta, g}^{LR}(x^{ij}) = \frac{1}{1 + e^{-(\beta^\top x^{ij} + g_i)}}, \quad (20)$$

$\delta^\top = (\beta, g^\top) \in \mathbb{R}^{1+p+K}$ and

$$a^\top = \left[\sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s}) [(x^{ij})^\top] \quad \sum_{j=1}^{\Gamma_1} (s_{1j} - \bar{s}) \dots \sum_{j=1}^{\Gamma_K} (s_{Kj} - \bar{s}) \right] \in \mathbb{R}^{1 \times (1+p+K)}.$$

Note that the last term in the objective function (19) aims to preserve the proportion between classifications for both sensitive categories, as explained in (5).

Now that we have our optimization problem (19), we need to adapt the Hessian (\mathcal{FH} in [JVP2]) and the gradient (\mathcal{FS} in [JVP2]), considering the fairness, given that the algorithm we propose is based on Newton's method, which requires such information.

Subsequently, based on the coefficients β , several directions are calculated for finding the best direction to update. For this, we use the Bayesian Information Criterion (BIC) $\in \mathbb{R}^p$. The BIC is a popular model selection criterion for GLMMs for being relatively easy to calculate, and it has been shown to perform well in a variety of simulations as can be seen in Vrieze (2012). A more detailed explanation and the calculation of all the components mentioned here can be found in [JVP2].

Finally, the stopping criteria is determined by calculating the covariance matrix (Q) calculated from the components of the pseudo-Fisher information matrix (V). For a more detailed explanation, see [JVP2] and Gu et al. (2012).

With all this information at hand, in [JVP2] we propose the Fair Generalized Linear Mixed Model based on boosting free of disparate impact that is Algorithm 1 in [JVP2]. In the algorithm below, we present an adapted and reduced version of it.

Algorithm 1 : Fair Generalized Linear Mixed Model based on boosting
<p>Given: Initial parameter that can be seen in [JVP2].</p> <p>Iteration:</p> <p style="margin-left: 20px;">(1) Refitting of residuals:</p> <p style="margin-left: 40px;">For $l \in [1, l_{max}]$,</p> <p style="margin-left: 60px;">(i) Computation of parameters</p> <p style="margin-left: 80px;">For $r \in [1, p]$ the r-th Newton's method step has the form:</p> $\delta_r^{(l)} = (\mathcal{FH}_r^{(l-1)})^{-1}(\mathcal{FS}_r^{(l-1)})$ <p style="margin-left: 60px;">(ii) Selection step</p> <p style="margin-left: 80px;">Select from $r \in [1, p]$ the index j corresponding to the smallest $BIC_r^{(l)}$ and select the related $(\delta_j^{(l)})^\top = (\beta_0^*, \beta_j^*, (g^*)^\top)$.</p> <p style="margin-left: 60px;">(iii) Update</p> <p style="margin-left: 80px;">Set</p> $\beta_0^{(l)} = \beta_0^{(l-1)} + \beta_0^*, \text{ being } \beta_0 \text{ the intercept,}$ $g^{(l)} = g^{(l-1)} + g^*$ <p style="margin-left: 40px;">and for $r \in [1, p]$ set</p> $\beta_r^{(l)} = \begin{cases} \beta_r^{(l-1)} & \text{if } r \neq j \\ \beta_r^{(l-1)} + \beta_r^* & \text{if } r = j \end{cases}$ <p style="margin-left: 40px;">with $A := [X]$ update</p> $\eta = A\delta^{(l)}$ <p style="margin-left: 20px;">(2) Computing of variance-covariance components:</p> $Q^{(l)} = \frac{1}{K} \sum_{i=1}^K (V_i^{(l)} + g_i^{(l)}(g_i^{(l)})^\top).$ <p>until $Q^{(l)} = Q^{(l-1)}$.</p>

The main motivation for developing the Fair BGLMM, rather than solving the optimization problem (19), is that in each iteration of Fair BGLMM, λ is updated, changing the importance of the variance of the random effects in each iteration instead of fixing it, as in model (19).

It can be observed that the Algorithm 1 is only proposed for the Disparate Impact fairness constraint. This is due to the fact that, as mentioned previously, the algorithm utilizes the Hessian and gradient calculations. Consequently, the Disparate Mistreatment constraint, which involves minimum value functions in the constraints, becomes into a non-differentiable penalized objective function, leading to issues in constructing the necessary components.

In [JVP2] we also present a cluster-regularized logistic regression and a brief description is show below.

4.3 Cluster-Regularized Logistic Regression

Preliminaries numerical tests observed that, when the survey constructs a dataset that indeed has random effects, the results of the Algorithm 1 are very good, as will be seen in the next section. However, it is noted that when this is not the case, the algorithm worsens the results compared to regular logistic regression.

This is due to the fact that if there is no random effect, $g_i = 0, \forall i \in [1, K]$. Hence we obtain the variance $\sigma_g^2 = 0$ which imposes difficulties when estimating the Variance-Covariance matrix via maximum likelihood. It is noteworthy that this would be irrelevant to the optimization problem itself because the solver employs algorithms that do not rely on the Variance-Covariance matrix as GLMM does.

For this reason, in [JVP2] we also propose the following optimization model that deal with this issue using the same $L2$ regularization as Problem (MESVM).

$$\min_{\beta, g} - \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \left[\left(\frac{y_{ij} + 1}{2} \right) \log(m_{\beta, g}^{LR}(x^{ij})) + \left(\frac{y_{ij} - 1}{2} \right) \log(1 - m_{\beta, g}^{LR}(x^{ij})) \right] + \lambda \sum_{i=1}^K g_i^2 \quad (21a)$$

$$\text{s.t. } \frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) \leq c \quad (21b)$$

$$\frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) \geq -c \quad (21c)$$

with $m_{\beta, g}^{LR}(x^{ij})$ from (20), \bar{s} from Expression (4).

The last term in the objective function of the problem (21) penalizes the variance of the random effects as seen in mixed effects SVM.

The optimization problem for the constraints of disparate mistreatment was also proposed in [JVP3].

$$\begin{aligned}
\min_{\beta, g} & - \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \left[\left(\frac{y_{ij} + 1}{2} \right) \log(m_{\beta, g}^{LR}(x^{ij})) + \left(\frac{y_{ij} - 1}{2} \right) \log(1 - m_{\beta, g}^{LR}(x^{ij})) \right] + \lambda \sum_{i=1}^K g_i^2 \\
\text{s.t.} & \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) \leq c, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) \geq -c, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) \leq c, \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) \geq -c.
\end{aligned} \tag{22}$$

Below, we will examine the effectiveness of all methods proposed in this section.

4.4 Numerical Results

In this section we present the numerical results regarding the fair generalized linear mixed models and the cluster-regularized logistic regression presented in [JVP2] and [JVP3]. The strategy for creating synthetic datasets is similar to the one present in Section 3.3, but using the prediction functions (16). Details and the complete tests can be seen in [JVP2] and [JVP3]. All tests can be reproduced, and the codes of all functions used can be found in [GitHub](#).

The comparisons present in [JVP2] is the accuracy and the disparate impact between the Algorithms:

1. Generalized linear mixed model (GLMM);
2. Generalized linear mixed model free of disparate impact (Fair GLMM);
3. Cluster-Regularized Logistic Regression (CRLR) (21a);
4. Cluster-Regularized Logistic Regression free of disparate impact (Fair CRLR) (21),

5. Logistic Regression (LR) (LR);
6. Logistic Regression free of disparate impact (Fair LR) (17).

Furthermore, two populations are compared, each possessing a sensitive variable, with your coefficient being β_4 . One population is created with a random effect acting in the classification, while the other not.

Unfair population with group effect

Parameters of data generation:

- β 's = $[-2.0, 0.4, 0.8, 0.5, 3.0]$;
- $c = 0.1$;
- g 's: 100 strata with $g_i \sim N(0, 2)$, with $i \in [1, 100]$;
- $\rho = 0.8$;
- $\lambda = 1$.

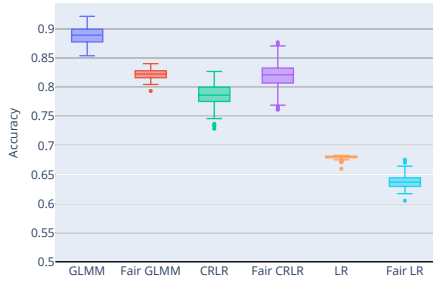


Figure 4.1: Accuracy.

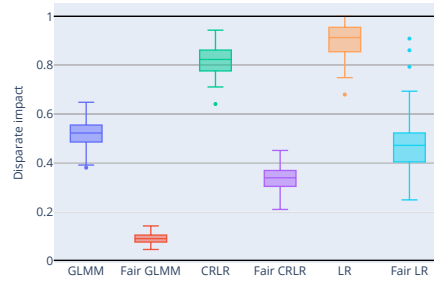


Figure 4.2: Disparate impact.

The results in Figure 4.1 demonstrate superior accuracy for GLMM, Fair GLMM, CRLR, and Fair CRLR in this experiment set. This is unsurprising, as logistic regression doesn't account for random effects. We can also see that the GLMM performs better on both metrics compared to the optimization problems.

Figure 4.2 show that we also obtained an improvement in the disparate impact on the Fair algorithms. Note that make sense since we have an unfair population.

Unfair population without group effect

Parameters of data generation:

- β 's = $[-0.8, 0.4, 0.8, 0.5, 1.8]$;
- $c = 0.1$;
- $\rho = 0.8$;
- $\lambda = 1$.

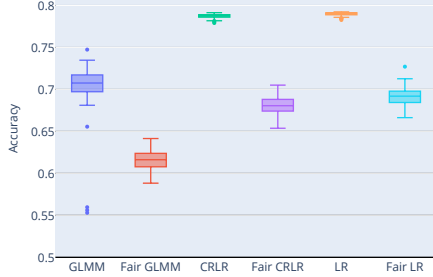


Figure 4.3: Accuracy.

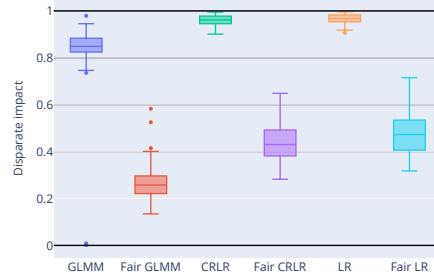


Figure 4.4: Disparate impact.

Our experiments, detailed in Figure 4.3, show a decrease in accuracy for GLMM algorithms. This is likely because GLMMs attempt to account for random effects, even when they are absent. This additional parameter estimation in GLMMs can lead to a reduction in accuracy compared to regular logistic regression optimization problems.

Figure 4.4 show that we obtained an improvement in the disparate impact on the Fair algorithms. Note that make sense since we still have a unfair population.

Cluster-Regularized Logistic Regression free of disparate mistreatment

Given that Problems (22) and (18) were introduced in [JVP3], we have opted to separate the numerical experiments from those presented in earlier work, as detailed in [JVP2]. Parameters of data generation:

- β 's = $[-4.0; 0.4; 0.8; 0.5; 4.0]$;
- $c = 0.1$;
- g 's: 100 groups with $g_i \sim N(0, 2)$, with $i \in [1, 100]$.

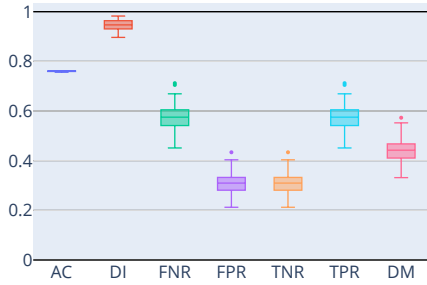


Figure 4.5: Regular logistic regression.

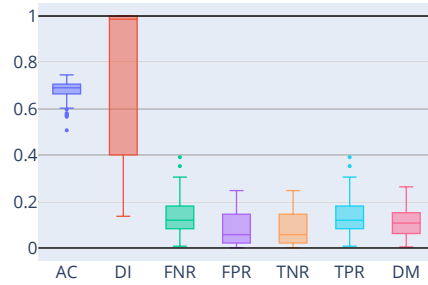


Figure 4.6: Logistic Regression free of disparate mistreatment.

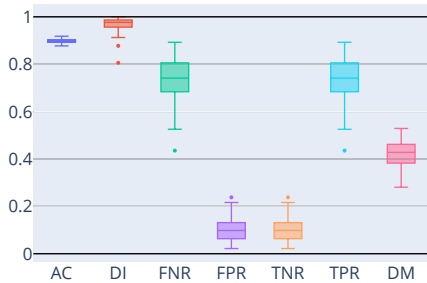


Figure 4.7: Cluster-Regularized Logistic Regression.

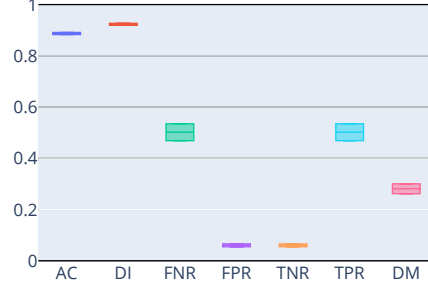


Figure 4.8: CRLR free of disparate mistreatment.

In this set of tests, we confirm that standard logistic regression, as illustrated in Figure 4.5, exhibits worse disparate mistreatment compared to its fair adaptation, in Figure 4.6. Figures 4.7 and 4.8 follows the same logic, yet achieving higher accuracy when compared to the standard logistic regression model that does not account for random effects.

In [JVP2], the bank marketing dataset was analyzed using the methods presented here. This dataset exhibits a group bias related to the duration of telephone calls. Longer call duration are associated with a higher likelihood of the target variable being 1. Additionally, the dataset includes a sensitive feature: housing loan, suffering of disparate impact. This feature can be considered sensitive due to its direct connection to systemic injustice, as discussed in *Fair Housing Trends Report (2023)* and in Howell (2006).

In the next chapter, we explore how to integrate all of the components developed thus far into a powerful tool for fair classifications.

Chapter 5

FairML: A Julia Package for Fair Classification

In [JVP3], we propose the FairML.jl a Julia package for fair classifications, that seamlessly integrate with the established MLJ.jl package (Blaom et al. 2019), enabling the utilization of all classification methods therein. In this extend summary we present an overview of [JVP3]. The package can be downloaded on [GitHub](#).

Packages for fair classification are already part of the literature, with versions available for Python (Jesus et al. 2024) and R (Scutari 2023). There also exists the Fairness package (Agrawal et al. 2020) in Julia, aiming to equalize accuracies across sensitive groups. Although these packages present several techniques, none of them consider mixed effects.

The first thing to be considered is that the first column of the matrix X should be a vector of ones, that is, the first entrance of $x^\ell, \forall \ell \in [1, n]$, is equal to 1. If this column does not exist, the functions of this package automatically add one.

Our Julia package, FairML, employ a variety of optimization problems and a resampling technique to enforce a user-defined fairness metric in classifications. The package operates under a three-step framework:

1. Preprocessing: This stage encompasses the implementation of functions that perform initial data manipulation aimed at enhancing fairness metrics. Returns the modified version of the training set;
2. In-processing: This stage constitutes the main part of the package, where optimization problems are addressed with the aim of improving a specific fairness metric, such as disparate impact, false positive rate, false negative rate, or disparate treatment. Returns all probabilities of the points being classified as positive in the training set and in the new dataset;

3. Post-processing: Following the previous stage, which outputs class membership probabilities, this phase is responsible for performing classification. It may or may not employ strategies to optimize a specific fairness metric. Returns the final classification.

All the theoretical underpinnings, construction and explanation of each stage are detailed in [JVP3]. In this chapter we present the package’s core functionality. They are the Classification function (Julia Code 1) and the Classification function for mixed models (Julia Code 2). These functions unifies all stages into a single, user-friendly interface:

Julia Code 1 Classification function.

```

1 classification = fair_pred(xtrain::DataFrame,
  ↪ ytrain::Vector{Union{Float64, Int64}}, newdata::DataFrame,
  ↪ inprocess::Function, SF::Array{String}, preprocess::Function=id_pre,
  ↪ postprocess::Function=id_post, c::Real=0.1, R::Int64=1,
  ↪ seed::Int64=42, SFpre::String=SF, SFpost::String=SF)

```

And an alternative for mixed models:

Julia Code 2 Classification function for mixed models.

```

1 classification = me_fair_pred(xtrain::DataFrame, ytrain::Vector,
  ↪ newdata::DataFrame, group_id_train::CategoricalVector,
  ↪ group_id_test::CategoricalVector, inprocess::Function,
  ↪ SF::Array{String}, postprocess::Function=id_post, c::Real=0.1,
  ↪ SFpost::String=SF)

```

Being:

- Input arguments:
 1. *xtrain*: The dataset that the labels are known (training set);
 2. *ytrain*: The labels of the dataset *xtrain*;
 3. *newdata*: The new dataset for which we want to obtain the *classification*;
 4. *inprocess*: One of the several optimization problems discussed in Chapters 3 and 4 or any machine learning method present in MLJ.jl package;

5. *SF*: One or a set of sensitive features (variables names. E.g Sex, race...), that act in the in-processing phase. If the algorithm come from the MLJ.jl package, no fairness constraint are acting in this phase;
 6. *group_id_train*: Training set group category;
 7. *group_id_newdata*: New dataset group category.
- Optional argument:
 1. *preprocess*: A preprocessing function as will be explained in Section 5.1 (Not enabled by default);
 2. *postprocess*: A post-processing function as will be explained in Section 5.2 (Not enabled by default);
 3. *c*: The threshold of the fair optimization problems as explained in Chapter 2, 0.1 by default;
 4. *R*: Number of iterations of the preprocessing phase, each time sampling differently using the resampling method that will be seen in 5.1, 1 by default;
 5. *seed*: For sample selection in *R*, 42 by default;
 6. *SFpre*: One sensitive features (variable name), that act in the preprocessing phase, disabled by default;
 7. *SFpost*: One sensitive features (variable name), that act in the post-processing phase, disabled by default.
 - Output arguments:
 1. *classification*: Classification of the *newdata* points.

The mixed models classification function ignores the preprocessing phase, as this phase tends to eliminate numerous data points, as will be discussed in 5.1. Such elimination can lead to empty groups, which is not permissible in the classification functions for mixed models, hence no parameter *R* is necessary. Moreover, the allowed functions in in-processing phase are only the ones present in [JVP1], [JVP2] and [JVP3] for mixed effects that are explained in Chapter 3 and 4. Functions from MLJ.jl package are not allow because they do not deal with the mixed effects.

It is essential to highlight that both the preprocessing and post-processing stages should be limited to handling a single sensitive feature each. Only the in-processing stage can handle with multiple sensitive features at the same time, creating multiples fairness constraints for the optimization problems. However, sensitive features can differ across the three phases with the aim to achieve fairness through various potential discrimination classes.

5.1 Preprocessing

In [JVP3] we propose a novel preprocessing technique based on resampling techniques (Good 2006). Resampling methods can serve various purposes, as can be seen in Good (2013). In our case, the goal is to mitigate disparate impact or disparate mistreatment in the data. We achieve this by generating multiple datasets that exhibit less unfairness than the original. In this context, we developed a hybrid approach that combines an adapted undersampling technique with cross-validation to address this issue. Undersampling (Mohammed et al. 2020) reduces the majority class, in the sensitive feature, to balance the dataset, while cross-validation (Blagus and Lusa 2015) provides a evaluation of the model by iteratively training and testing on different subsets. Similar approaches have been used for class-imbalanced data in Zughrat et al. (2014) and Jesus et al. (2024).

As indicated by Equation (3), regarding to disparate impact, our goal is to ensure equal representation of positive and negative labels across both categories of the sensitive features. To achieve this, we enforce this condition within the training set \mathcal{D} using the following strategy:

1. Separate the training data \mathcal{D} as in Equation (2);
2. Compute the size of the smallest among the four subsets:

$$J = \min(|\mathcal{D}_0^{\mathcal{N}}|, |\mathcal{D}_1^{\mathcal{N}}|, |\mathcal{D}_0^{\mathcal{P}}|, |\mathcal{D}_1^{\mathcal{P}}|).$$

3. For each subset do a random sampling with replacement, (M) as follows:

$$\begin{aligned} M_J^{\mathcal{D}_0^{\mathcal{N}}} &\subseteq \mathcal{D}_0^{\mathcal{N}}, \text{ with } |M_J^{\mathcal{D}_0^{\mathcal{N}}}| = J, & M_J^{\mathcal{D}_0^{\mathcal{P}}} &\subseteq \mathcal{D}_0^{\mathcal{P}}, \text{ with } |M_J^{\mathcal{D}_0^{\mathcal{P}}}| = J, \\ M_J^{\mathcal{D}_1^{\mathcal{N}}} &\subseteq \mathcal{D}_1^{\mathcal{N}}, \text{ with } |M_J^{\mathcal{D}_1^{\mathcal{N}}}| = J, & M_J^{\mathcal{D}_1^{\mathcal{P}}} &\subseteq \mathcal{D}_1^{\mathcal{P}}, \text{ with } |M_J^{\mathcal{D}_1^{\mathcal{P}}}| = J. \end{aligned}$$

4. Create the new training dataset:

$$\mathbb{D} = M_J^{\mathcal{D}_0^{\mathcal{N}}} \cup M_J^{\mathcal{D}_1^{\mathcal{N}}} \cup M_J^{\mathcal{D}_0^{\mathcal{P}}} \cup M_J^{\mathcal{D}_1^{\mathcal{P}}}.$$

Since the proportion of labels across each different sensitive feature categories is the same, we expected to have a new dataset with less disparate impact than the previous one. The documentation of this function can be found on [GitHub](#).

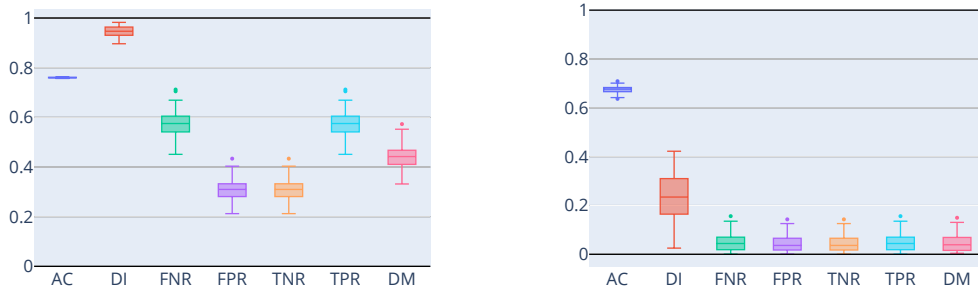
Observe that we have a random choice of which points belong to the new dataset. Hence we can construct the new dataset in more than one way. We do this using the R and the *seed* variables, in `Julia code 1`. Thus, given a desired value by the user, the code generate R datasets using the preprocessing function and choose the best one as follows:

1. Do the preprocessing phase R times, generating R different datasets;
2. For each dataset:
 - (a) Calculate the coefficients using the in-processing phase;
 - (b) Compute the classifications on the full training set (before resampling);
 - (c) Use the classifications to calculate disparate impact or disparate mistreatment;
3. Select the classification with the best fairness metric value;
4. Use the coefficients given by the in-processing phase for the dataset chosen above to calculate classifications on new data.

That is, from all the R calculated coefficients, this phase selects the one that generate the smallest disparate impact or disparate mistreatment on the full training set, and uses it to classify the points in the new dataset (input *newdata*).

While the algorithm was designed to address disparate impact, preliminary numerical tests have shown that it can also mitigate disparate treatment. This makes it a flexible tool, allowing the user to choose the specific focus.

Building upon the synthetic dataset creation techniques presented in Chapter 3, we now show some numerical results present in [JVP3] that demonstrate the efficacy of our preprocessing method. For more details we refer to [JVP3].



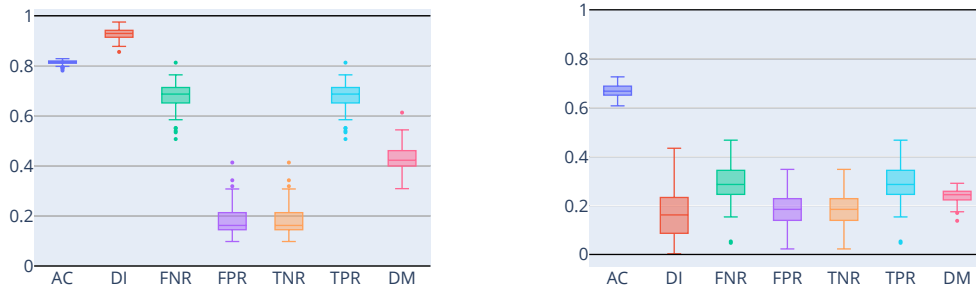


Figure 5.1: Preprocessing results: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Without preprocessing. Right: With preprocessing ($R=1$)

As can be seen in Figure 5.1, for both problems logistic regression and SVM, the proposed resample method significantly reduces the disparate impact when compared to the standard approaches (without the preprocessing phase). It is also worth nothing that the proposed preprocessing method has also lower value in other fairness metrics as well. As expected in the field of fair classification, the cost of the improvement in the fairness metrics is the drop of the accuracy. Now, considering the same preprocessing but being executed multiple times:

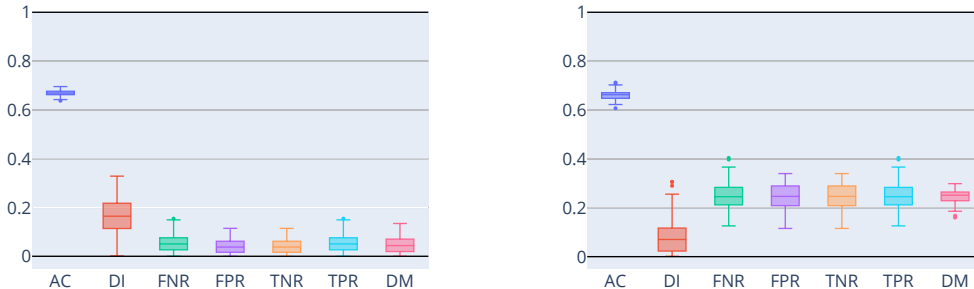


Figure 5.2: Preprocessing with multiple runs ($R=5$): Left: Logistic regression. Right: Support vector machine.

Figure 5.2 shows the preprocessing process executed 5 times, that is, $R = 5$. It can be seen that repeating the resampling method and selecting the best solution has a lower value of DI when compared with do the preprocessing only one time. Observe that given R , the preprocessing phase run R times. This means that the higher the R the longer it takes to run. Hence, a large value of R must be chosen when time is not an issue.

5.2 Post-processing

The post-processing phase of the package proposed in [JVP3] is an algorithm that seeks an optimal cut-off value for classification (Cheong et al. 2013; Ren et al. 2016). An approach that implements a similar strategy, but considering each sensitive group, can be seen in Jesus et al. (2024). In our approach, we consider the entire dataset to ensure that no particular sensitive group is advantaged or disadvantaged. This phase can be applied for any fairness metric and in datasets with fixed and mixed effects. Documentations for all available options are provided on [GitHub](#).

Given the probabilities from both the training and the new datasets obtained in the in-processing phase:

1. For each cut-off value v ranging from 0.01 to 0.99 (with an increment of 0.01), do:
 - Generate classifications for X_{train} as follows: if the probability is greater or equal v , classify as positive, otherwise as negative;
 - Compute the accuracy (AC_v) and desired fairness metric value (fm_v) for X_{train} .
2. Select only the values of v that decrease at most 5% of the accuracy compared to the accuracy given by the cut-off value $v = 0.5$. Among them, select the best result using $B = \operatorname{argmax}_v(AC_v - fm_v)$;
3. Use the new cut-off value, B , for the test set classification.

If the user does not wish to use this phase in the classification process, the cut-off value v is 0.5 by default. The value of 5% was determined through preliminary tests which demonstrated that allowing a greater reduction in accuracy could misclassify a significant number of data points into a specific class.

We now present some of the numerical tests in [JVP3] that shows the benefits of the post-processing phase. The post-processing phase, as the preprocessing phase, can be utilized independently, without the in-processing phase affecting the fairness metrics, or both phases can be employed simultaneously.

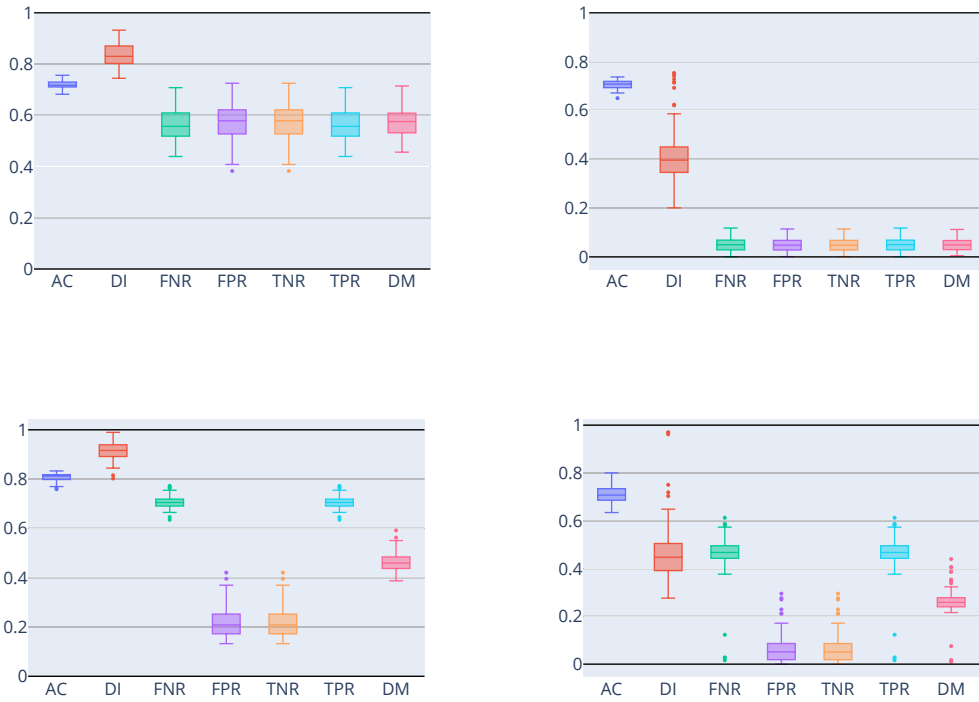
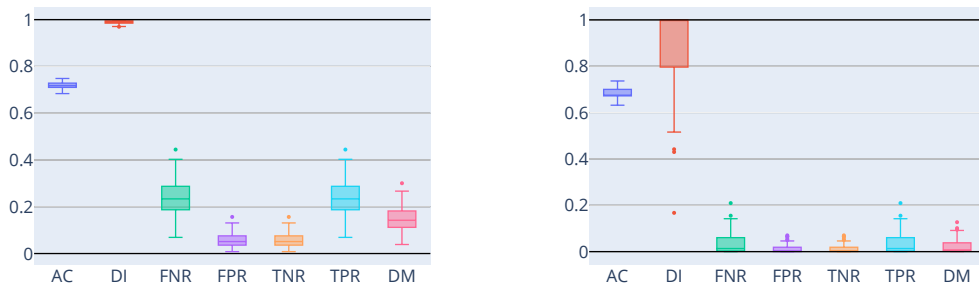


Figure 5.3: Post-processing results for disparate impact: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.



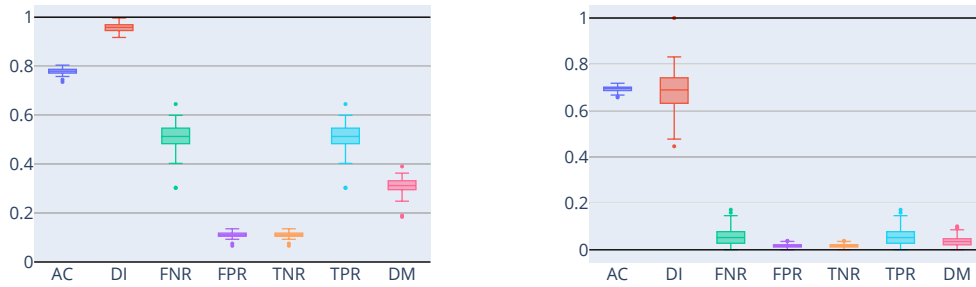
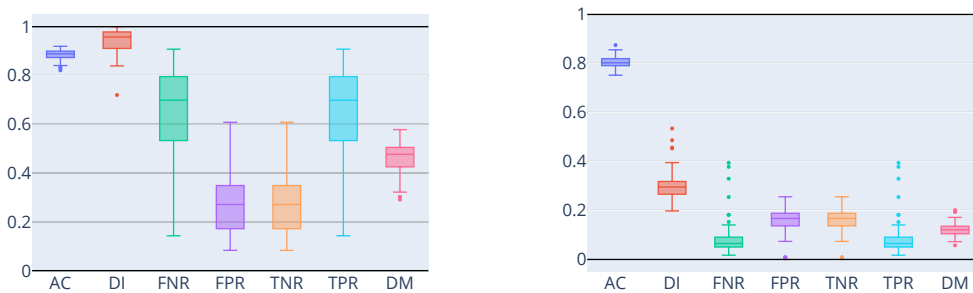


Figure 5.4: Post-processing results for disparate mistreatment: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.

The post-processing phase can be utilized independently, without the in-processing phase affecting the fairness metrics, or both phases can be employed simultaneously. It can be observed, in the left side of Figures 5.3 and 5.4, that employing solely the post-processing phase leads to a slight improvement in the desired fairness metric without significantly compromising accuracy. Furthermore, it can be noted that utilizing both strategies in conjunction, as can be seen in the right side of Figures 5.3 and 5.4, yields superior outcomes compared to employing either in-processing or post-processing alone. Consequently, our recommendation is to utilize both strategies simultaneously.

At last, we show the effectiveness of the post-processing step on the mixed models problems.



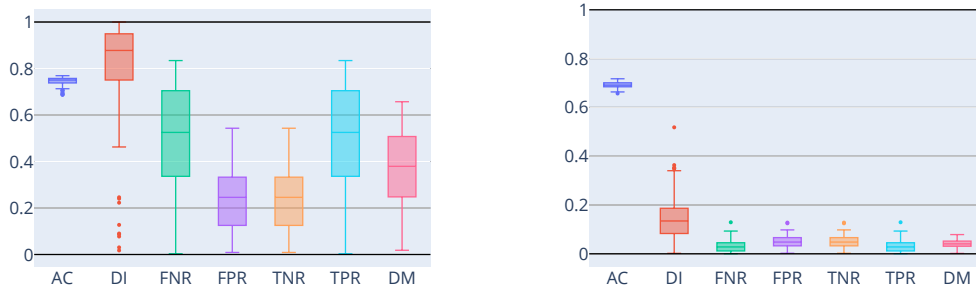


Figure 5.5: Post-processing results in mixed models for disparate impact: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.

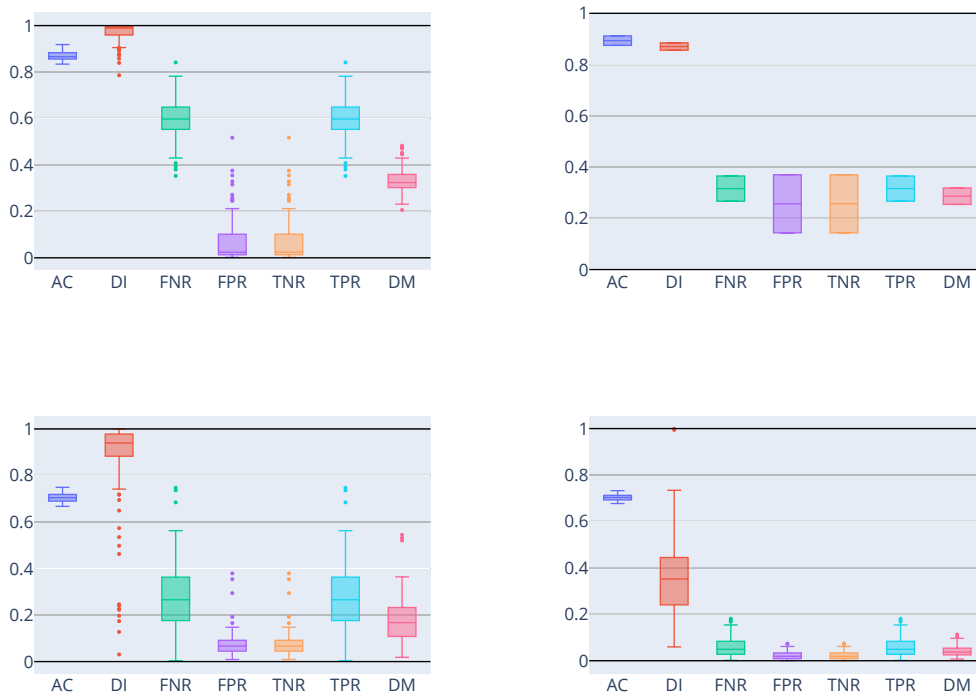


Figure 5.6: Post-processing results in mixed models for disparate mistreatment: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.

Just as in post-processing tests for regular models, while the post-processing phase can function independently, its integration with the in-processing phase yields

superior results. Consequently, we reiterate our recommendation to employ both phases simultaneously as can be seen in Figures 5.5 and 5.6. For more numerical results we refer to [JVP3].

Chapter 6

Conclusions and outlook

In numerous classification tasks, the mitigation of bias poses a significant challenge. Our research delves into two distinct types of bias: potential discrimination inherent in the dataset and biases introduced by group sampling strategies commonly employed in constructing these datasets. In these thesis, we propose three novel models to address both issues: a Fair Mixed Effects Support Vector Machine, a Fair Cluster Regularized Logistic Regression, and a Fair Generalized Linear Mixed Model based on boosting. Additionally, we have developed a comprehensive package that encapsulates all the stated strategies and more.

Furthermore, regarding fair mixed effects support vector machine we propose a novel algorithm addressing the dual challenges of disparate impact (or disparate mistreatment) and mixed effects in machine learning models. By integrating mixed effects into the SVM framework and employing innovative regularization techniques, the model effectively mitigates unfairness while preserving accuracy. Comprehensive evaluations across various datasets demonstrate the algorithm's superior performance in reducing disparate impact compared to regular SVM. This research significantly advances fairness-aware machine learning by providing a unified solution to complex challenges, paving the way for more equitable and reliable models.

In the context of logistic regression we introduce a novel algorithm for fair generalized linear mixed models based on boosting and a cluster-regularized logistic regression, both addressing unfairness metrics, the first dealing with disparate impact and the second dealing also with disparate mistreatment, in the presence of group structures. Through simulations we demonstrate its effectiveness in mitigating discrimination while preserving predictive accuracy. Our results emphasize the significance of incorporating random effects into predictive models. We demonstrate that neglecting these effects can lead to a substantial decline in accuracy, resulting in poor classification performance.

Our proposed `Julia` package, the `FairML.jl` addresses fairness in machine learning, offering versatile tools to tackle unfairness at various stages of the classification process. In the preprocessing phase we mitigate disparate impact or disparate mistreatment through a resampling strategy. After we propose constrained optimization models that mitigate unfairness in homogeneous and heterogeneous populations. This phase also allows the utilization of any binary classifier from package `MLJ.jl` as learning tool. Additionally, post-processing method was designed to identify a solution that improves a specified fairness metric, without significantly compromising accuracy. Our simulations demonstrate the effectiveness of these approaches and their potential for combination.

Despite the contributions made in the context of binary classification, there remains ample scope for further refinement and investigation. A primary area for future research is the exploration of novel fairness metrics and their adaptation to the constraints of optimization problems, particularly within the framework of mixed models where existing literature is notably sparse. Moreover, these new metrics would be integrated into the proposed package.

Another extension of our research would be to modify other machine learning algorithms, such as random forests and decision trees, to mitigate both unfairness and mixed effects bias simultaneously. Additionally, we could consider extending our proposed approach to multi-class classification algorithms and investigating how fairness metrics would function without binary labels or with continuous sensitive features.

Bibliography

- Agasisti, T, F Ieva, and A. M. Paganoni (2017). “Heterogeneity, school-effects and the North/South achievement gap in Italian secondary education: evidence from a three-level mixed model.” In: *Statistical Methods & Applications* 26, pp. 157–180.
- Aghaei, S, M. J. Azizi, and P Vayanos (2019). “Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making.” In: *CoRR* abs/1903.10598. arXiv: 1903.10598. URL: <http://arxiv.org/abs/1903.10598>.
- Agrawal, A, J Chen, S Vollmer, and A Blaom (Aug. 2020). *ashryaagr/Fairness.jl*. Version v0.1.2. DOI: 10.5281/zenodo.3977197. URL: <https://doi.org/10.5281/zenodo.3977197>.
- Barili, F, A Parolari, P. A. Kappetein, and N Freemantle (2018). “Statistical Primer: heterogeneity, random-or fixed-effects model analyses?” In: *Interactive cardiovascular and thoracic surgery* 27.3, pp. 317–321.
- Barocas, S, E Bradley, V Honavar, and F Provost (2017). “Big data, data science, and civil rights.” In: *arXiv preprint arXiv:1706.03102*.
- Barocas, S and A. D. Selbst (2016). “Big data’s disparate impact.” In: *Calif. L. Rev.* 104, p. 671.
- Berk, R, H Heidari, S Jabbari, M Joseph, M Kearns, J Morgenstern, S Neel, and A Roth (2017). “A convex framework for fair regression.” In: *arXiv preprint arXiv:1706.02409*.
- Berman, E and J Ginesin (2024). “The State of Julia for Scientific Machine Learning.” In: *arXiv preprint arXiv:2410.10908*.
- Besançon, M, T Papamarkou, D Anthoff, A Arslan, S Byrne, D Lin, and J Pearson (2021). “Distributions.jl: Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem.” In: *Journal of Statistical Software* 98.16, pp. 1–30. DOI: 10.18637/jss.v098.i16.
- Bezanson, J, A Edelman, S Karpinski, and V. B. Shah (2017). “Julia: A fresh approach to numerical computing.” In: *SIAM review* 59.1, pp. 65–98. DOI: 10.1137/14100067.

- Blagus, R and L Lusa (2015). “Joint use of over-and under-sampling techniques and cross-validation for the development and assessment of prediction models.” In: *BMC bioinformatics* 16, pp. 1–10.
- Blaom, A, F Kiraly, T Lienart, and S Vollmer (Nov. 2019). *alan-turing-institute/MLJ.jl: v0.5.3*. Version v0.5.3. DOI: [10.5281/zenodo.3541506](https://doi.org/10.5281/zenodo.3541506).
- Bouchet-Valat, M and B Kamiński (2023). “DataFrames.jl: Flexible and Fast Tabular Data in Julia.” In: *Journal of Statistical Software* 107.4, pp. 1–32. DOI: [10.18637/jss.v107.i04](https://doi.org/10.18637/jss.v107.i04).
- Carrizosa, E, M Galvis-Restrepo, and D Romero Morales (Feb. 2022). *Improving the fairness of linear models in supervised classification by feature shrinkage*.
- Carrizosa, E, T Halskov, and D Romero Morales (June 2023a). *Wasserstein SVM: Support Vector Machines Made Fair*.
- Carrizosa, E, K Kurishchenko, and D Romero Morales (Sept. 2023b). *On enhancing the explainability and fairness of tree ensembles*.
- Caton, S and C Haas (2020). “Fairness in machine learning: A survey.” In: *ACM Computing Surveys*.
- Cheng, Q, J Tezcan, and J Cheng (2014). “Confidence and prediction intervals for semiparametric mixed-effect least squares support vector machine.” In: *Pattern Recognition Letters* 40, pp. 88–95. DOI: <https://doi.org/10.1016/j.patrec.2013.12.010>.
- Cheong, K. C., A. F. Yusoff, S. M. Ghazali, K. H. Lim, S Selvarajah, J Haniff, G. L. Khor, S Shahar, R. J. Abd, A. A. Zainuddin, et al. (2013). “Optimal BMI cut-off values for predicting diabetes, hypertension and hypercholesterolaemia in a multi-ethnic population.” In: *Public health nutrition* 16.3, pp. 453–459.
- Corporation, I. (2023). *Intel Math Kernel Library*.
- Das, S, M Donini, J Gelman, K Haas, M Hardt, J Katzman, K Kenthapadi, P Larroy, P Yilmaz, and M. B. Zafar (2021). “Fairness measures for machine learning in finance.” In: *The Journal of Financial Data Science*.
- Do, H, P Putzel, A. S. Martin, P Smyth, and J Zhong (2022). “Fair generalized linear models with a convex penalty.” In: *International Conference on Machine Learning*. PMLR, pp. 5286–5308.
- Dolan, E. D. and J. J. Moré (2002). “Benchmarking Optimization Software with Performance Profiles.” In: *Math. Program.* 91, pp. 201–213. DOI: [10.1007/s101070100263](https://doi.org/10.1007/s101070100263).
- Domingos, P (2012). “A few useful things to know about machine learning.” In: *Communications of the ACM* 55.10, pp. 78–87.
- Fair Housing Trends Report* (2023). <https://nationalfairhousing.org/resource/2023-fair-housing-trends-report/>.
- Fernandes, A, Solimun, and Nurjannah (Feb. 2022). “Computational Statistics with Dummy Variables.” In: DOI: [10.5772/intechopen.101460](https://doi.org/10.5772/intechopen.101460).

- Friedman, J, T Hastie, and R Tibshirani (2010). “Regularization paths for generalized linear models via coordinate descent.” In: *Journal of statistical software* 33.1, p. 1.
- Good, P (2013). *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media.
- Good, P. I. (2006). *Resampling methods*. Springer.
- Green, B (2018). “Fair risk assessments: A precarious approach for criminal justice reform.” In: *5th Workshop on fairness, accountability, and transparency in machine learning*, pp. 1–5.
- Greene, W. H. (1997). *Econometric Analysis*. Prentice-Hall international editions. Prentice Hall.
- Gu, Q, Z Li, and J Han (2012). “Generalized fisher score for feature selection.” In: *arXiv preprint arXiv:1202.3725*.
- Gurobi Optimization, LLC (2024). *Gurobi Optimizer Reference Manual*. URL: <https://www.gurobi.com>.
- Habermas, J (2003). “Intolerance and discrimination.” In: *International Journal of Constitutional Law* 1.1, pp. 2–12.
- Hardt, M, E Price, and N Srebro (2016). “Equality of opportunity in supervised learning.” In: *Advances in neural information processing systems* 29.
- Hastie, T. J. and D. Pregibon (2017). “Generalized linear models.” In: *Statistical models in S*. Routledge, pp. 195–247.
- Hearst, M. A., S. T. Dumais, E Osuna, J Platt, and B Scholkopf (1998). “Support vector machines.” In: *IEEE Intelligent Systems and their applications* 13.4, pp. 18–28.
- Howell, B (2006). “Exploiting Race and Space: Concentrated Subprime Lending as Housing Discrimination.” In: *California Law Review* 94.1, pp. 101–147.
- Hsieh, C. J., K Chang, and C. J. Lin (Jan. 2008). “A dual coordinate descent method for large-scale linear SVM.” In: *Proceedings of the Twenty-fifth International Conference on Machine Learning*, pp. 1369–1398.
- Hu, S, Y Wang, C Drovandi, and T Cao (Sept. 2022). “Predictions of machine learning with mixed-effects in analyzing longitudinal data under model misspecification.” In: *Statistical Methods & Applications* 32, pp. 1–31. DOI: [10.1007/s10260-022-00658-x](https://doi.org/10.1007/s10260-022-00658-x).
- Jesus, S, P Saleiro, B. M. Jorge, R. P. Ribeiro, J Gama, P Bizarro, R Ghani, et al. (2024). “Aequitas Flow: Streamlining Fair ML Experimentation.” In: *arXiv preprint arXiv:2405.05809*.
- Jiang, Z, X Han, C Fan, F Yang, A Mostafavi, and X Hu (2022). “Generalized demographic parity for group fairness.” In: *International Conference on Learning Representations*.

- Lange, K (2002). “Newton’s Method and Scoring.” In: *Mathematical and Statistical Methods for Genetic Analysis*. New York, NY: Springer New York, pp. 39–58. DOI: [10.1007/978-0-387-21750-5_3](https://doi.org/10.1007/978-0-387-21750-5_3).
- Long, R (2021). “Fairness in machine learning: Against false positive rate equality as a measure of fairness.” In: *Journal of Moral Philosophy* 19.1, pp. 49–78.
- Luts, J, G Molenberghs, G Verbeke, S Huffel, and J Suykens (Mar. 2012). “A mixed effects least squares support vector machine model for classification of longitudinal data.” In: *Computational Statistics & Data Analysis* 56, pp. 611–628. DOI: [10.1016/j.csda.2011.09.008](https://doi.org/10.1016/j.csda.2011.09.008).
- Ma, Y, Y Qiao, M Chen, D Rui, X Zhang, W Liu, and L Ye (2025). “How small is big enough? Big data-driven machine learning predictions for a full-scale wastewater treatment plant.” In: *Water Research* 274, p. 123041.
- McGill, R, J. W. Tukey, and W. A. Larsen (1978). “Variations of box plots.” In: *The American Statistician* 32.1, pp. 12–16.
- Menon, A. K. and R. C. Williamson (2018). “The cost of fairness in binary classification.” In: *Conference on Fairness, accountability and transparency*. PMLR, pp. 107–118.
- Mijalkovic, J and A Spognardi (2022). “Reducing the false negative rate in deep learning based network intrusion detection systems.” In: *Algorithms* 15.8, p. 258.
- Miller J, R. S. (1966). “Sex discrimination and title VII of the Civil Rights Act of 1964.” In: *Minn. L. Rev.* 51, p. 877.
- Mohammed, R, J Rawashdeh, and M Abdullah (2020). “Machine learning with oversampling and undersampling techniques: overview study and experimental results.” In: *2020 11th international conference on information and communication systems (ICICS)*. IEEE, pp. 243–248.
- Neter, D. J., M. H. Kutner, and C. J. Nachtsheim (2004). *MP Applied Linear Regression Models-Revised Edition with Student CD*. McGraw-Hill Education.
- Nocedal, J (2006). “Penalty and Augmented Lagrangian Methods.” In: *Numerical Optimization*. Springer New York, pp. 497–528.
- Oberg, A. L. and D. W. Mahoney (2007). “Linear mixed effects models.” In: *Topics in biostatistics*, pp. 213–234.
- Olfat, M and A Aswani (2017). “Spectral Algorithms for Computing Fair Support Vector Machines.” In: *CoRR* abs/1710.05895. arXiv: [1710.05895](https://arxiv.org/abs/1710.05895). URL: <http://arxiv.org/abs/1710.05895>.
- Radovanović, S, A Petrović, B Delibašić, and M Suknović (2020). “Enforcing fairness in logistic regression algorithm.” In: *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–7. DOI: [10.1109/INISTA49547.2020.9194676](https://doi.org/10.1109/INISTA49547.2020.9194676).

- Ren, Q, C Su, H Wang, Z Wang, W Du, and B Zhang (2016). “Prospective study of optimal obesity index cut-off values for predicting incidence of hypertension in 18–65-year-old Chinese adults.” In: *PloS one* 11.3, e0148140.
- Scutari, M (2023). *fairml: A Statistician’s Take on Fair Machine Learning Modelling*. arXiv: 2305.02009 [stat.ML]. URL: <https://arxiv.org/abs/2305.02009>.
- Speicher, T, H Heidari, N Grgic-Hlaca, K. P. Gummadi, A Singla, A Weller, and M. B. Zafar (2018). “A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual & Group Unfairness via Inequality Indices.” In: *CoRR* abs/1807.00787. arXiv: 1807.00787.
- Srivastava, P, A Yaduvanshi, S Singh, T Islam, and M Gupta (Feb. 2016). “Support vector machines and generalized linear models for quantifying soil dehydrogenase activity in agro-forestry system of mid altitude central Himalaya.” In: *Environmental Earth Sciences* 75. DOI: [10.1007/s12665-015-5074-3](https://doi.org/10.1007/s12665-015-5074-3).
- Stroup, W. W. (2012). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.
- Tutz, G and A Groll (2010). “Generalized linear mixed models based on boosting.” In: *Statistical Modelling and Regression Structures: Festschrift in Honour of Ludwig Fahrmeir*, pp. 197–215.
- Vapnik, V and A. Y. Chervonenkis (1964). “A class of algorithms for pattern recognition learning.” In: *Avtomat. i Telemekh* 25.6, pp. 937–945.
- Vrieze, S. I. (2012). “Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).” In: *Psychological methods* 17.2, p. 228.
- Wächter, A and L. T. Biegler (2006). “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.” In: *Mathematical programming* 106, pp. 25–57.
- Wang, Y, D Sridhar, and D Blei (2023). “Adjusting Machine Learning Decisions for Equal Opportunity and Counterfactual Fairness.” In: *Transactions on Machine Learning Research*.
- Yang, J, N. A. Zaitlen, M. E. Goddard, P. M. Visscher, and A. L. Price (2014). “Advantages and pitfalls in the application of mixed-model association methods.” In: *Nature genetics* 46.2, pp. 100–106.
- Zafar, M, I Valera, M Rodriguez, and K. P. Gummadi (Oct. 2016). “Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment.” In.
- Zafar, M. B., I Valera, M Gomez-Rodriguez, and K. P. Gummadi (2017). “Fairness Constraints: Mechanisms for Fair Classification.” In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by A.

- Singh and J. Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, pp. 962–970.
- Zafar, M. B., I Valera, M Gomez-Rodriguez, and K. P. Gummadi (2019). “Fairness Constraints: A Flexible Approach for Fair Classification.” In: *Journal of Machine Learning Research* 20.75, pp. 1–42.
- Zhang, W, A Bifet, X Zhang, J. C. Weiss, and W Nejdl (2021). “FARF: A Fair and Adaptive Random Forests Classifier.” In: *CoRR* abs/2108.07403. arXiv: [2108.07403](https://arxiv.org/abs/2108.07403). URL: <https://arxiv.org/abs/2108.07403>.
- Zhao, H and G. J. Gordon (2022). “Inherent tradeoffs in learning fair representations.” In: *Journal of Machine Learning Research* 23.57, pp. 1–26.
- Zou, H and T Hastie (2005). “Regularization and variable selection via the elastic net.” In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2, pp. 301–320.
- Zughrat, A, M Mahfouf, Y. Y. Yang, and S Thornton (2014). “Support vector machines for class imbalance rail data classification with bootstrapping-based over-sampling and under-sampling.” In: *IFAC Proceedings Volumes* 47.3, pp. 8756–8761.

Part II

Reprints of the Scientific
Papers

Paper 1

Fair Mixed Effects Support Vector Machine

Jan Pablo Burgard, João Vitor Pamplona

Preprint under review.

URL: <https://arxiv.org/abs/2405.06433>

FAIR MIXED EFFECTS SUPPORT VECTOR MACHINE

JAN PABLO BURGARD, JOÃO VITOR PAMPLONA

ABSTRACT. To ensure unbiased and ethical automated classifications, fairness must be a core principle in machine learning applications. Fairness in machine learning aims to mitigate biases present in the training data and model imperfections that could lead to discriminatory outcomes. This is achieved by preventing the model from making decisions based on sensitive characteristics like ethnicity or sexual orientation. A fundamental assumption in machine learning is the independence of observations. However, this assumption often does not hold true for data describing social phenomena, where data points are often clustered based. Hence, if the machine learning models do not account for the cluster correlations, the results may be biased. Especially high is the bias in cases where the cluster assignment is correlated to the variable of interest. We present a fair mixed effects support vector machine algorithm that can handle both problems simultaneously. With a reproducible simulation study we demonstrate the impact of clustered data on the quality of fair machine learning classifications.

1. INTRODUCTION

The rise of automated decision-making systems calls for the development of fair algorithms. These algorithms must be constrained by societal values, particularly to avoid discrimination against any population group. While machine learning offers efficiency gains, it can also unintentionally perpetuate biases in critical areas like loan approvals (Das et al. 2021) and criminal justice (Green 2018). In loan applications, factors like marital status can lead to unfair disadvantages for single individuals. At the same time, in criminal justice, algorithms might associate race with recidivism risk, leading to discriminatory sentencing despite individual circumstances. This highlights the need for fair and unbiased AI frameworks to ensure equal opportunities and outcomes for all.

Driven by the need to mitigate bias in algorithms, fair machine learning is experiencing a surge in research activity. Numerous research articles are exploring innovative approaches to achieve fairer outcomes. Notable examples include fair versions of Logistic and Linear Regression (Berk et al. 2017), Support Vector Machines (SVM) (Olfat and Aswani 2017), Random Forests (Zhang et al. 2021), Decision Trees (Aghaei et al. 2019), and Generalized Linear Models (GLMs) (Do et al.

Date: December 2, 2024.

2020 Mathematics Subject Classification. 90C90, 90-08, 68T99 .

Key words and phrases. Support Vector Machine, Fair Machine Learning, Mixed Models.

2022). These methods aim to address potential discrimination arising from historical data or algorithmic design, ensuring fairer outcomes for all individuals.

A key challenge in machine learning for automated decision-making is the training data. Often sourced from surveys, this data may not perfectly align with the assumption common in machine learning that all elements are sampled independently with equal probability of inclusion. However, some data may suffer from cluster effects, e.g., in marketing, customers who buy a particular product might also be interested in a similar product. Ignoring these effects can lead to misleading results, such as underestimating the true variability in the data. To overcome it, random effects are incorporated into the statistical model, which, together with the existing fixed effects, leads to a mixed effects model, as can be seen in Oberg and Mahoney (2007).

In this paper we propose a Mixed Effects Support Vector Machine for fair classifications. We show how to estimate the model and evaluate its performance against the standard SVM model that does not take the possible clustering of the data into account. Similar approaches for longitudinal data can be found in Luts et al. (2012) and Hu et al. (2022), for Least-squares support vector machine in Cheng et al. (2014) and applications in agricultural activities in Srivastava et al. (2016).

The paper is organized as follows: In Section 2 we explore the theory and metrics behind fairness in machine learning and how it applies to support vector machines. In Section 3 we establish the theoretical underpinnings of fair mixed effects support vector machine and propose a strategy for solving it. In Section 4, we conduct a comprehensive evaluation of our proposed method’s effectiveness through various tests. Finally, in Section 5, we demonstrate the practical applicability of the algorithm by solving a real-world problem using the Adult dataset (Becker and Kohavi 1996). Our key findings and potential future directions are presented in Section 6.

2. FAIR SUPPORT VECTOR MACHINE

Classification algorithms in machine learning are used to estimate a specific classification $\hat{y} \in \{-1, 1\}$ for a new data point x based on a training set $\mathcal{D} = (x^\ell, y_\ell)_{\ell=1}^n$. For the point x^ℓ , if $y_\ell = 1$, we say that x^ℓ is in the positive class and if $y_\ell = -1$, x^ℓ belongs to the negative class with $x^\ell \in \mathbb{R}^{p+1}$, for each $\ell \in [1, n] := \{1, \dots, n\}$, belonging to data $X = [x^1, \dots, x^n]$ being the dimension of x , $p + 1$, due to the addition of an extra column with the value 1 as the data intercept.

In the realm of fair binary classification, each observation x^ℓ possesses a corresponding sensitive attribute s_ℓ a binary value of either 0 or 1, the goal becomes identifying a solution that balances accuracy with fairness. Fairness in machine learning can be assessed using various metrics. In this paper, we specifically focus on disparate impact (DI). Alternative fairness metrics for binary classifiers can be found in Barocas and Selbst (2016).

Consider $\mathcal{S}_1 = \{x^\ell : s_\ell = 1\}$ and $\mathcal{S}_0 = \{x^\ell : s_\ell = 0\}$ as disjoint subsets of dataset X , where the sensitive feature of all points in this subset is 1 and 0, respectively. A

binary classifier is considered free of disparate impact if the probability of classifying an instance as either 0 or 1 is equal in \mathcal{S}_1 and \mathcal{S}_0 . In other terms, disparate impact is absent when the classifier treats all groups equally regardless of their sensitive feature values. Mathematically, we can write this as:

$$P(\hat{y}_\ell = 1 | x_\ell \in \mathcal{S}_0) = P(\hat{y}_\ell = 1 | x_\ell \in \mathcal{S}_1).$$

To this end, we must maintain a proportional relationship between the classifications for both classes of the sensitive feature. Defining

$$\frac{\sum_{\ell=1}^n s_\ell}{n} = \frac{|\mathcal{S}_1|}{n} =: \bar{s}$$

because $s_\ell \in \{0, 1\}$ and knowing that in SVM, proposed by Vapnik and Chervonenkis (1964), a hyperplane $\beta^\top x$ splits the feature space of the data based on the classification of each point we want to maintain the proportionality of classifications for both sensitive categories, so:

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{S}_1} \beta^\top x^\ell = \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{S}_0} \beta^\top x^\ell.$$

This means that

$$\begin{aligned} 1 - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{S}_1} \beta^\top x^\ell &= \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{S}_0} \beta^\top x^\ell \\ \implies (1 - \bar{s}) \sum_{x^\ell \in \mathcal{S}_1} \beta^\top x^\ell &= \bar{s} \sum_{x^\ell \in \mathcal{S}_0} \beta^\top x^\ell \\ \implies \sum_{x^\ell \in \mathcal{S}_1} (1 - \bar{s}) \beta^\top x^\ell &= \sum_{x^\ell \in \mathcal{S}_0} (\bar{s} - 0) \beta^\top x^\ell \\ \implies \sum_{x^\ell \in \mathcal{S}_1} (1 - \bar{s}) \beta^\top x^\ell + \sum_{x^\ell \in \mathcal{S}_0} (0 - \bar{s}) \beta^\top x^\ell &= 0 \\ \implies \sum_{\ell=1}^n (s_\ell - \bar{s}) (\beta^\top x^\ell) &= 0. \end{aligned}$$

To maintain consistency with the previously proposed literature (Zafar et al. 2017), we divide the sum above by n . Note that this does not alter the equality.

In the following, we present how to add fairness in the context of Support Vector Machine (SVM). Note that in our formulation the first column of X is equal to 1 for all points and, hence, the bias parameter is the first entry in β (Hsieh et al. 2008). By adding the fairness constraint, the fixed effect β can be found by solving the following quadratic optimization problem.

$$\min_{(\beta, \xi)} \frac{1}{2} \|\beta\|^2 + K \sum_{\ell=1}^N \xi_\ell \quad (1a)$$

$$\text{s.t. } y_\ell (\beta^\top x_\ell) \geq 1 - \xi_\ell, \ell = 1, \dots, N, \quad (1b)$$

$$\frac{1}{N} \sum_{\ell=1}^N (s_{\ell} - \bar{s})(\beta^{\top} x_{\ell}) \leq c, \quad (1c)$$

$$\frac{1}{N} \sum_{\ell=1}^N (s_{\ell} - \bar{s})(\beta^{\top} x_{\ell}) \geq -c, \quad (1d)$$

$$\xi_{\ell} \geq 0, \ell = 1, \dots, N. \quad (1e)$$

Constraints (1a), (1b) and (1e) are from the Support Vector Machine problem. Constraints (1c) and (1d) guarantees the fairness in the classification. This model is proposed by Zafar et al. (2019). While achieving zero disparate impact is a desirable goal, it can potentially come at the expense of a good classification as we have a trade-off between fairness and accuracy (Menon and Williamson 2018; Zhao and Gordon 2022). To address this exchange, we can introduce a fairness threshold, denoted by $c \in \mathbb{R}^+$, which allows us to adjust the relative importance placed on fairness compared to accuracy. The penalty parameter K aims to control the importance of slack variables, which represents the flexibility in classifying a point considering the optimal hyperplane. We refer to the problem above as Fair Support Vector Machine (SVMF).

In the following example, the two sensitive categories are distinguished by the shape of the point (diamond and star). The color differentiates the label, with red being positive (1) and blue being negative (-1).

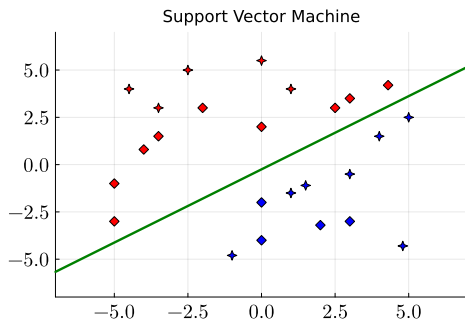


FIGURE 1. Regular SVM.

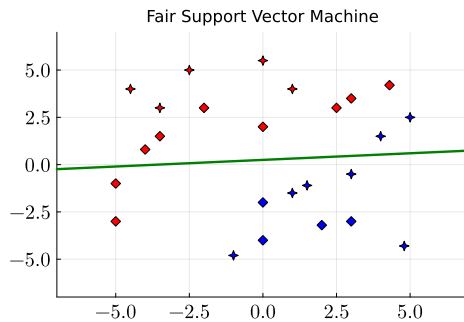


FIGURE 2. SVM free of disparate impact.

Figure 1 demonstrates 100% accuracy, as each point is located on the correct side of the hyperplane. However, when we look at each sensitive feature category, we observe that approximately 36%(5/14) of the star points are classified as positive while this happens for approximately 64%(9/14) of the diamond points. That is, the proportions of the two sensitive categories being classified as positive are not equal.

Figure 2 shows an example of SVM free of disparate impact as in Problem (1), considering $c = 0$. We can see that 50% of the star points and approximately 53% of the diamond points are classified as positive. That is, the proportion of positive classifications is similar for both sensitive categories. However this comes at the cost

of accuracy, as 4 points were misclassified, reducing the accuracy to 84%. This can be mitigated by choosing a threshold c , sufficiently good so as not to significantly impair accuracy.

3. FAIR MIXED EFFECTS SUPPORT VECTOR MACHINE

Real-world data often shows heterogenic variations within groups. Consider a dataset comprising multiple schools where we aim to ascertain whether distinct teachers yield varying outcomes for a particular variable. In a predictive modeling context, we would classify these teaching capacity variables as fixed effects, that is effects that represents factors that are of direct interest and are explicitly modeled (Agasisti et al. 2017). However, it is plausible that other unmeasured factors, such as the quality of teaching materials or the overall school environment, also influence the outcomes. These unmeasured factors are typically modeled as random effects, that is factors that are not of primary interest but introduce variability into the data (Greene 1997).

To account for the heterogeneity introduced by these random effects and to obtain unbiased estimates of the impact of teachers on student outcomes, it is imperative to incorporate them into the predictive model. Neglecting these random effects could lead to substantial bias in the classifications, particularly when comparing outcomes across different schools. Statistical tools can estimate their impact on grouped data.

Let g being the random vector and g_i with $i \in [1, K]$, representing the group-specific random effect, with g following a normal distribution with mean zero. Consider Γ_i the size of the group i for each $i \in [1, K]$ and y_{ij} the label of $(x^{ij})^\top = (x_1^{ij}, \dots, x_p^{ij})$ with $j \in [1, \Gamma_i]$.

In essence, the fair constraints discussed in the section above only consider fixed effects (β 's). However, in mixed models we need to consider the random effects g_i , that is, $\beta^\top x^{ij} + g_i$ being the inner product used to compute the probability of the point x^{ij} being classified as 1. In light of these considerations, we propose the following adaptation in the constraints:

$$\mathcal{S}_1^i = \{x^{ij} : j \in [1, \Gamma_i], s_{ij} = 1\} \quad \mathcal{S}_0^i = \{x^{ij} : j \in [1, \Gamma_i], s_{ij} = 0\}$$

Observe that each subset is created for each cluster $i \in [1, K]$.

Moreover, we need to modify the prediction function in the fair constraints to account for random effects.

Following the same logic as presented before, but considering a group-to-group analysis, we have a similar construction for the disparate impact constraints:

$$\begin{aligned} \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (\beta^\top x^{ij} + g_i) &= \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\beta^\top x^{ij} + g_i) \\ \implies 1 - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (\beta^\top x^{ij} + g_i) &= \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\beta^\top x^{ij} + g_i) \end{aligned}$$

$$\begin{aligned}
&\implies (1 - \bar{s}) \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (\beta^\top x^{ij} + g_i) = \bar{s} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\beta^\top x^{ij} + g_i) \\
&\implies \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (1 - \bar{s})(\beta^\top x^{ij} + g_i) = \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (\bar{s} - 0)(\beta^\top x^{ij} + g_i) \\
&\implies \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_1^i} (1 - \bar{s})(\beta^\top x^{ij} + g_i) - \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{S}_0^i} (0 - \bar{s})(\beta^\top x^{ij} + g_i) = 0 \\
&\implies \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} + \bar{s})(\beta^\top x^{ij} + g_i) = 0.
\end{aligned}$$

Besides that is important control the random effects variance we do that penalizing it through L2 regularization, aiming to minimize it. Similar approaches for least-square support vector machine can be found in Luts et al. (2012) and as demonstrated in Domingos (2012), Friedman et al. (2010), and Zou and Hastie (2005), minimizing the variance enhances model performance and generalization. High variance often implies that the model is excessively sensitive to the training data, resulting in sub-optimal performance on unseen data. Hence the sum $\sum_i^K g_i^2$ is from the fact that the variance of g is given by:

$$\sigma^2 = \frac{\sum_{i=1}^K (g_i - \bar{g})^2}{K - 1},$$

and the random effect g follows a normal distribution with mean zero.

$$\sigma^2 = \frac{\sum_{i=1}^K (g_i - 0)^2}{K - 1} = \frac{\sum_{i=1}^K g_i^2}{K - 1}.$$

This means that the regularization term is $\frac{\sum_{i=1}^K g_i^2}{K-1}$. Since we have a minimization problem and $K - 1$ is fixed, we can consider w.l.o.g a parameter λ that controls the importance of the random effects.

Consequently, we formulate the following constrained optimization problem:

$$\min_{(\beta, b, \xi)} \frac{1}{2} \|\beta\|^2 + \mu \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \xi_{ij} + \lambda \sum_{i=1}^K g_i^2 \quad (2a)$$

$$\text{s.t. } y_{ij}(m_{\beta, g}^{SVM}(x^{ij})) \geq 1 - \xi_{ij}, \quad (2b)$$

$$\frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) \leq c, \quad (2c)$$

$$\frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) \geq -c, \quad (2d)$$

$$\xi_{ij} \geq 0, \quad i = 1, \dots, K, \quad j = 1, \dots, \Gamma_i, \quad (2e)$$

The objective function (2a) and the constraints (2b) and (2e) are a adapted version of Support Vector Machine problem to deal with random effects. Constraints (2c)

and (2d) guarantees the fairness, in terms of disparate impact. We refer to the problem above as Fair Mixed Effects Support Vector Machine (FMESVM) and without the fairness constraints as Mixed Effects Support Vector Machine (MESVM).

In our numerical experiments, we set variables μ and λ equal to 1. The choice of μ reflects the desire to assign equal importance to maximizing the margin and minimizing the classification error. The value of λ is determined by the aim of balancing the influence of random and fixed effects within the optimization problem. Preliminary tests revealed that setting λ significantly higher than 1 tends to prioritize random effects, resulting in a random vector with excessive variance, even when the objective is to minimize it. On the other hand, if λ is considerably lower than 1, the optimization process becomes overly focused on fixed effects, leading to random effects that are almost negligible.

A One-hot encoding alternative. One alternative approach to incorporating group effects into grouped data analysis involves the use of dummy variables, also known as one-hot encoding as can be seen in Fernandes et al. (2022). This strategy involves creating a separate binary variable for each group, taking a value of 1 for observations within that group and 0 otherwise. To the best of our knowledge, no one has previously employed this technique in the SVM setting in the literature; however, we consider this method, and, while it can effectively capture group-level variation, preliminary numerical results have demonstrated that the proposed approach offers significant advantages in terms of both computational efficiency and memory usage, as can be seen below. This happens because, in the one-hot encoding, the dimension of each x^ℓ is increased in the number K of groups, that is, $x \in \mathbb{R}^{p+1+K}$.

The following preliminary numerical experiment employed a dataset consisting of 100000 points with the training set having 3 to 5 points per group and systematically varying the number of groups within the data, considering configurations with 2, 10, 50, 500, 1000, 1250, 2000, 2500, 3125, 4000, and 5000 groups.

Figure 3 shows the Memory comparison in Bytes and Figure 4 the time comparison. The first row of each figure present a second-order polynomial fit, where the x-axis represents the number of groups and the y-axis the memory. The second row present the Performance Profile proposed by Dolan and Moré (2002) of both approaches. In this specific numerical test, the problem (2) is the Mixed Effects Support Vector Machine free of Disparate Impact and is labeled as FMESVM.

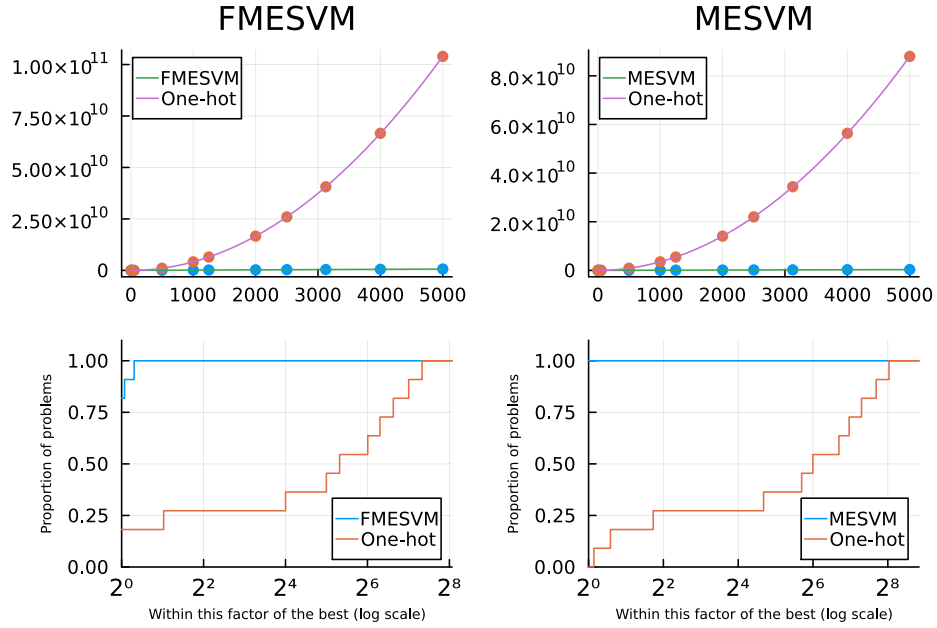


FIGURE 3. Memory comparison in Bytes.

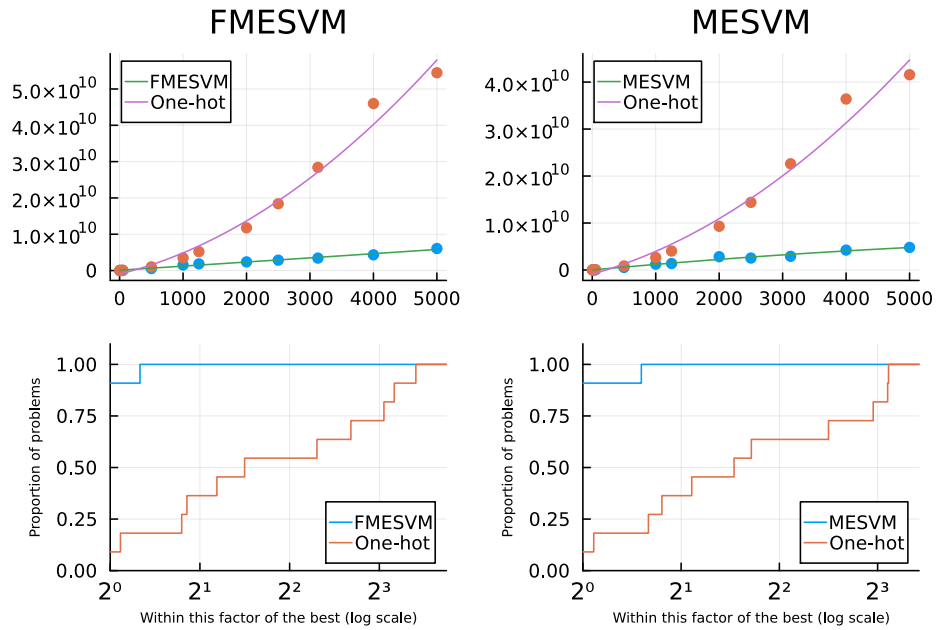


FIGURE 4. Time comparison in Microsecond.

As can be seen in Figures 3 and 4, our optimization models outperforms the one-hot encoding in time and memory. Hence, the approach is a more resource-friendly option, mainly for large datasets and complex models in both situations where the

datasets present unfairness and when they do not. While one-hot encoding can be applied to SVM, our model is a better choice.

The next section dives into numerical tests, demonstrating the efficacy of the proposed methodology.

4. SIMULATION STUDY

First we present the step-by-step strategy used to create the datasets and to conduct the numerical experiments. Using `Julia 1.9` (Bezanson et al. 2017) with the packages `Distributions` (Besançon et al. 2021), `DataFrames` (Bouchet-Valat and Kamiński 2023), `MLJ` (Blaom et al. 2019) and `MKL` (Corporation 2023). The following parameters are generated randomly. Their specific values will be determined at a later stage.

- *Number of points*: Number of points in the dataset;
- *β 's*: The fixed effects;
- *g 's*: The random effects with distribution $N(0, 2)$;
- *Data points*: The covariate vector associated with fixed effects with distribution $N(0, 1)$;
- *c* : Threshold from Fair Support Vector Machine;
- *seed*: Random seed used in the generation of data;
- *Train-Test split*: Approximately 1% of the dataset was used for the training set, and 99% for the test set. This percentage was due to the fact that we randomly selected 3 to 5 points from each cluster for the training set.

Then the labels of the synthetic dataset were computed using

$$m = \beta^T x + g \quad (3)$$

in tests where the dataset has random effects, and

$$m = \beta^T x \quad (4)$$

in the tests where the dataset has just fixed effects. Since $m \in [-\infty, \infty]$, we project the value to $[0, 1]$ using the function:

$$M = \frac{\exp(m)}{1 + \exp(m)}. \quad (5)$$

Finally, the true label of each point x is given by:

$$y = \text{Bernoulli}(M). \quad (6)$$

Thus, the datasets are ready.

Regarding the tests, the comparisons are made between the following optimization problems:

- (1) Mixed Effects Support Vector Machine (MESVM);
- (2) Fair Mixed Effects Support Vector Machine (FMESVM);
- (3) Support Vector Machine (SVM);
- (4) Fair Support Vector Machine (SVMF).

We then, compare the accuracy and the disparate impact of each method above. The test were conducted on a computer with an Intel Core i9-13900HX processor with a clock speed of 5.40 GHz, 64 GB of RAM, and Windows 11 operating system, with 64-bit architecture.

To compute accuracy, first we need compute the classifications. For this, we use expressions (3) - (6) with

$$\hat{y} = \begin{cases} 1 & \text{if } M \geq 0.5 \\ -1 & \text{if } M < 0.5 \end{cases}.$$

Hence given the true label of all points, we can distinguish them into four categories: true positive (TP) or true negative (TN) if the point is classified correctly in the positive or negative class, respectively, and false positive (FP) or false negative (FN) if the point is misclassified in the positive or negative class, respectively. Based on this, we can compute the accuracy, where a higher value indicates a better classification, as follows,

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1].$$

To compute the Disparate Impact of a specific sensitive feature s we use the following equation based on Radovanović et al. (2020),

$$di := \frac{|\{\ell : \hat{y}_\ell = 1, x_\ell \in \mathcal{S}_0\}|}{|\mathcal{S}_0|} \frac{|\mathcal{S}_1|}{|\{\ell : \hat{y}_\ell = 1, x_\ell \in \mathcal{S}_1\}|} \in [0, \infty).$$

Disparate Impact, as a metric, should ideally be equal to 1 to indicate fair classifications. Values greater or lower than 1 suggest the presence of unfairness. For instance, both $di = 2$ and $di = 0.5$ represent the same amount of discrimination, but in opposite directions. While the former case deviates further from the ideal value (1) compared to the latter, the di metric itself does not capture this distinction. To address this limitation and achieve a more nuanced metric, we use the minimum value between the di and its inverse $\frac{1}{di}$ as follows:

$$DI := \min(di, di^{-1}) \in [0, 1]. \quad (7)$$

The parameters for the Unfair cases are:

- β 's = $[-1.5, 0.4, 0.8, 0.5, 1.5]$;
- g 's: 100 clusters with $b_i \sim N(0, Q)$, with $i \in [1, 100]$;
- $c = 10^{-3}$;
- $K = 1$;
- $\lambda = 1$.

And for the fair cases are:

- β 's = $[-1, 1, 2, 1, 0.1]$;
- g 's: 100 clusters with $b_i \sim N(0, Q)$, with $i \in [1, 100]$;
- $c = 10^{-3}$;
- $K = 1$;
- $\lambda = 1$,

with Q being 2 for cases with random effect and 0 for cases without random effect. The β_0 is related to the intercept, and β_4 are the coefficient of the binary sensitive feature. In the unfair cases the coefficients were randomly selected using numbers between 0 and 1, with the exception of the sensitive feature where we assign a disproportionately high value to the coefficient relative to the other coefficients, thereby emphasizing the greater significance of the sensitive variable in predicting the labels. In other words, data points with the sensitive categories equal to 1 are more likely to be classified as positive. This practice results in a dataset that is inherently unfair, as needed to test our method. For all experiments, the matrix X was randomly generated from a multivariate normal distribution with zero mean and independent variables. In fair datasets, we substantially decrease the coefficient value of the sensitive feature, aiming to minimize its correlation with the labels.

The following figures present experimental results on four different datasets with 1000 samples each. The datasets are presented individually, and the changes between samples occur only in the training set. Each figure contains four corresponding images representing accuracy and disparate impact for all possible data. All figures were created using the `Plots` and `PlotlyJS` packages, developed by Christ et al. (2023).

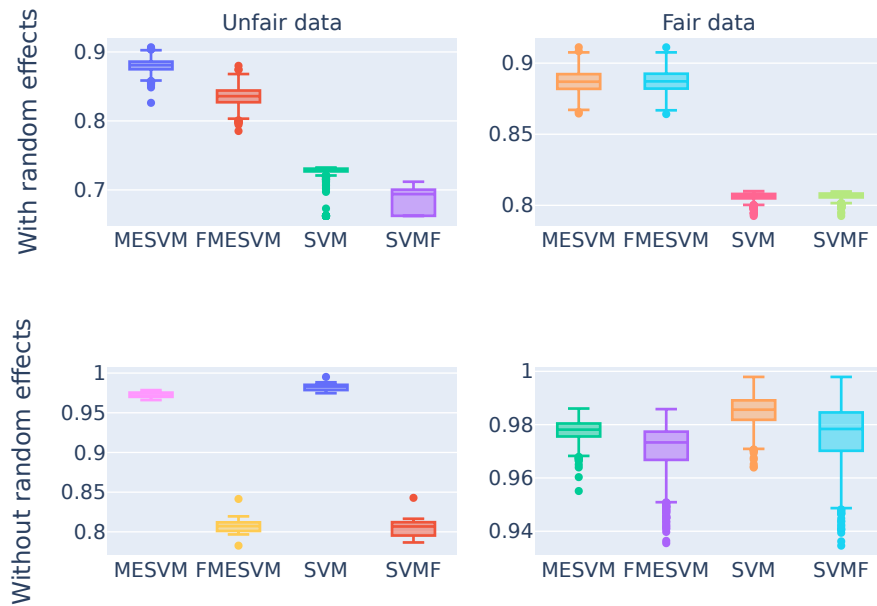


FIGURE 5. Accuracy.



FIGURE 6. Disparate Impact.

Figure 5 demonstrates that our proposed approach consistently outperformed alternative methods in scenarios with random effects. In settings without random effects, both methods achieved comparable accuracy, as anticipated. However, a slight decrease in accuracy was observed when introducing unfairness in the data. This is understandable, as the approach must balance addressing both random effects and unfairness simultaneously.

Figure 6 demonstrates that the approach consistently yields improved disparate impact metrics when applied to datasets containing inherent biases. Conversely, in scenarios where the underlying data exhibits no inherent bias, our approach produces equivalent outcomes, as there is no inherent disparity to mitigate.

5. APPLICATION

In this set of experiments, we do tests using the Adult dataset. To test the method’s efficiency, we created groups based on individuals age and marital status, and a sensitive feature, gender (Speicher et al. 2018). The Adult dataset is a famous tool in machine learning, where the goal is predicting whether the individuals earn more ($y = 1$) or less ($y = -1$) than 50000 USD annually. Were conducted 1000 samples with 0.5% of the data as the training set and the remaining data as the test set.

The features used in the classification process are the follows:

- Age: The age of the individual in years;
- Capital_gain: Capital gain in the previous year;
- Capital_loss: Capital loss in the previous year;
- Education: Highest level of education achieved by the individual;
- Education_num: A numeric form of the highest level of education achieved;
- Fnlwgt: An estimate of the number of individuals in the population with the same demographics as this individual;
- Hours_per_week: Hours worked per week;
- Marital_status: The marital status of the individual;
- Native_country: The native country of the individual;
- Occupation: The occupation of the individual;
- Race: The individual's race;
- Relationship: The individual's relationship status;
- Gender: The individual's gender;
- Workclass: The sector that the individual works in.

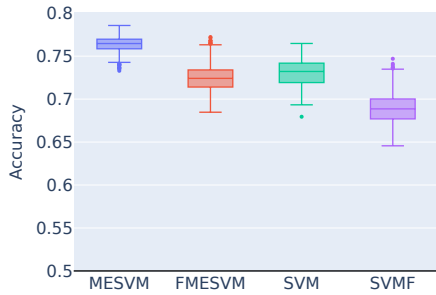


FIGURE 7. Accuracy.

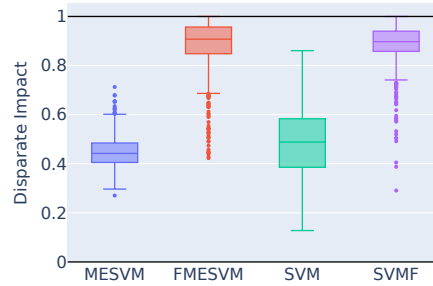


FIGURE 8. Disparate Impact.

As can be seen in Figure 7, in this set of experiments, we obtained a better accuracy in MESVM and FMESVM in comparison to regular SVM since this one not account for random effects.

Figure 8 show that we also obtained an improvement in the disparate impact on the Fair algorithms. Note that make sense since we have a unfair population.

Upon examining all the tests, we were able to observe an improvement in Disparate Impact in (100%) of the cases.

6. CONCLUSION

This study investigated a novel approach to mitigating disparate impact, a fairness issue in machine learning models, while simultaneously addressing mixed effects. We introduce a novel Fair Mixed Effects Support Vector Machine (FMESVM) algorithm that tackles both concerns cohesively, overcoming the limitations of existing methods often dedicated to separate problem solving. This integrated approach

tailors treatments to the specific demands of each issue, ensuring optimal performance.

Employing the widely respected Support Vector Machine (SVM) for binary classification, the FMESVM framework incorporates mixed effects within the SVM setting and deploys novel regularization techniques to effectively mitigate disparate impact. Extensive evaluation across diverse datasets and metrics demonstrates the success of our proposed method in demonstrably reducing disparate impact while maintaining or minimally compromising overall accuracy.

For comprehensive experimentation, we systematically explored all possible scenarios involving the two concerns: datasets exhibiting both, only unfairness with random effects, only fairness with random effects, and neither issue. This approach yielded expected results, with each combination directly impacting accuracy or disparate impact as predicted.

The FMESVM presents a significant advancement in fairness-aware machine learning by comprehensively addressing disparate impact and mixed effects through a unified framework. This paves the way for the development of more robust and ethical machine learning models with broader applicability.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the German Federal Ministry of Education and Research (BMBF) for this research project, as well as for the “OptimAgent Project”.

We would also like to express our sincere appreciation for the generous support provided by the German Research Foundation (DFG) within Research Training Group 2126 “Algorithmic Optimization”.

REFERENCES

- Agasisti, T, F Ieva, and A. M. Paganoni (2017). “Heterogeneity, school-effects and the North/South achievement gap in Italian secondary education: evidence from a three-level mixed model.” In: *Statistical Methods & Applications* 26, pp. 157–180.
- Aghaei, S, M. J. Azizi, and P Vayanos (2019). “Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making.” In: *CoRR* abs/1903.10598. arXiv: 1903.10598. URL: <http://arxiv.org/abs/1903.10598>.
- Barocas, S and A. D. Selbst (2016). “Big data’s disparate impact.” In: *Calif. L. Rev.* 104, p. 671.
- Becker, B and R Kohavi (1996). *Adult*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5XW20>.
- Berk, R, H Heidari, S Jabbari, M Joseph, M Kearns, J Morgenstern, S Neel, and A Roth (2017). “A convex framework for fair regression.” In: *arXiv preprint arXiv:1706.02409*.

- Besançon, M, T Papamarkou, D Anthoff, A Arslan, S Byrne, D Lin, and J Pearson (2021). “Distributions.jl: Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem.” In: *Journal of Statistical Software* 98.16, pp. 1–30. DOI: [10.18637/jss.v098.i16](https://doi.org/10.18637/jss.v098.i16).
- Bezanson, J, A Edelman, S Karpinski, and V. B. Shah (2017). “Julia: A fresh approach to numerical computing.” In: *SIAM review* 59.1, pp. 65–98. DOI: [10.1137/14100067](https://doi.org/10.1137/14100067).
- Blaom, A, F Kiraly, T Lienart, and S Vollmer (Nov. 2019). *alan-turing-institute/MLJ.jl: v0.5.3*. Version v0.5.3. DOI: [10.5281/zenodo.3541506](https://doi.org/10.5281/zenodo.3541506).
- Bouchet-Valat, M and B Kamiński (2023). “DataFrames.jl: Flexible and Fast Tabular Data in Julia.” In: *Journal of Statistical Software* 107.4, pp. 1–32. DOI: [10.18637/jss.v107.i04](https://doi.org/10.18637/jss.v107.i04).
- Cheng, Q, J Tezcan, and J Cheng (2014). “Confidence and prediction intervals for semiparametric mixed-effect least squares support vector machine.” In: *Pattern Recognition Letters* 40, pp. 88–95. DOI: <https://doi.org/10.1016/j.patrec.2013.12.010>.
- Christ, S, D Schwabeneder, C Rackauckas, M. K. Borregaard, and T Breloff (2023). “Plots.jl – a user extendable plotting API for the julia programming language.” In: DOI: <https://doi.org/10.5334/jors.431>.
- Corporation, I. (2023). *Intel Math Kernel Library*.
- Das, S, M Donini, J Gelman, K Haas, M Hardt, J Katzman, K Kenthapadi, P Larroy, P Yilmaz, and M. B. Zafar (2021). “Fairness measures for machine learning in finance.” In: *The Journal of Financial Data Science*.
- Do, H, P Putzel, A. S. Martin, P Smyth, and J Zhong (2022). “Fair generalized linear models with a convex penalty.” In: *International Conference on Machine Learning*. PMLR, pp. 5286–5308.
- Dolan, E. D. and J. J. Moré (2002). “Benchmarking Optimization Software with Performance Profiles.” In: *Math. Program.* 91, pp. 201–213. DOI: [10.1007/s101070100263](https://doi.org/10.1007/s101070100263).
- Domingos, P (2012). “A few useful things to know about machine learning.” In: *Communications of the ACM* 55.10, pp. 78–87.
- Fernandes, A, Solimun, and Nurjannah (Feb. 2022). “Computational Statistics with Dummy Variables.” In: DOI: [10.5772/intechopen.101460](https://doi.org/10.5772/intechopen.101460).
- Friedman, J, T Hastie, and R Tibshirani (2010). “Regularization paths for generalized linear models via coordinate descent.” In: *Journal of statistical software* 33.1, p. 1.
- Green, B (2018). “Fair risk assessments: A precarious approach for criminal justice reform.” In: *5th Workshop on fairness, accountability, and transparency in machine learning*, pp. 1–5.
- Greene, W. H. (1997). *Econometric Analysis*. Prentice-Hall international editions. Prentice Hall.

- Hsieh, C. J., K Chang, and C. J. Lin (Jan. 2008). “A dual coordinate descent method for large-scale linear SVM.” In: *Proceedings of the Twenty-fifth International Conference on Machine Learning*, pp. 1369–1398.
- Hu, S, Y Wang, C Drovandi, and T Cao (Sept. 2022). “Predictions of machine learning with mixed-effects in analyzing longitudinal data under model misspecification.” In: *Statistical Methods & Applications* 32, pp. 1–31. DOI: [10.1007/s10260-022-00658-x](https://doi.org/10.1007/s10260-022-00658-x).
- Luts, J, G Molenberghs, G Verbeke, S Huffel, and J Suykens (Mar. 2012). “A mixed effects least squares support vector machine model for classification of longitudinal data.” In: *Computational Statistics & Data Analysis* 56, pp. 611–628. DOI: [10.1016/j.csda.2011.09.008](https://doi.org/10.1016/j.csda.2011.09.008).
- Menon, A. K. and R. C. Williamson (2018). “The cost of fairness in binary classification.” In: *Conference on Fairness, accountability and transparency*. PMLR, pp. 107–118.
- Oberg, A. L. and D. W. Mahoney (2007). “Linear mixed effects models.” In: *Topics in biostatistics*, pp. 213–234.
- Olfat, M and A Aswani (2017). “Spectral Algorithms for Computing Fair Support Vector Machines.” In: *CoRR* abs/1710.05895. arXiv: [1710.05895](https://arxiv.org/abs/1710.05895). URL: <http://arxiv.org/abs/1710.05895>.
- Radovanović, S, A Petrović, B Delibašić, and M Suknović (2020). “Enforcing fairness in logistic regression algorithm.” In: *2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–7. DOI: [10.1109/INISTA49547.2020.9194676](https://doi.org/10.1109/INISTA49547.2020.9194676).
- Speicher, T, H Heidari, N Grgic-Hlaca, K. P. Gummadi, A Singla, A Weller, and M. B. Zafar (2018). “A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual & Group Unfairness via Inequality Indices.” In: *CoRR* abs/1807.00787. arXiv: [1807.00787](https://arxiv.org/abs/1807.00787).
- Srivastava, P, A Yaduvanshi, S Singh, T Islam, and M Gupta (Feb. 2016). “Support vector machines and generalized linear models for quantifying soil dehydrogenase activity in agro-forestry system of mid altitude central Himalaya.” In: *Environmental Earth Sciences* 75. DOI: [10.1007/s12665-015-5074-3](https://doi.org/10.1007/s12665-015-5074-3).
- Vapnik, V and A. Y. Chervonenkis (1964). “A class of algorithms for pattern recognition learning.” In: *Avtomat. i Telemekh* 25.6, pp. 937–945.
- Zafar, M. B., I Valera, M Gomez-Rodriguez, and K. P. Gummadi (2017). “Fairness Constraints: Mechanisms for Fair Classification.” In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by A. Singh and J. Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, pp. 962–970.
- (2019). “Fairness Constraints: A Flexible Approach for Fair Classification.” In: *Journal of Machine Learning Research* 20.75, pp. 1–42.

- Zhang, W, A Bifet, X Zhang, J. C. Weiss, and W Nejdl (2021). “FARF: A Fair and Adaptive Random Forests Classifier.” In: *CoRR* abs/2108.07403. arXiv: [2108.07403](https://arxiv.org/abs/2108.07403). URL: <https://arxiv.org/abs/2108.07403>.
- Zhao, H and G. J. Gordon (2022). “Inherent tradeoffs in learning fair representations.” In: *Journal of Machine Learning Research* 23.57, pp. 1–26.
- Zou, H and T Hastie (2005). “Regularization and variable selection via the elastic net.” In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2, pp. 301–320.

(J. P. Burgard, J. V. Pamplona) TRIER UNIVERSITY, DEPARTMENT OF ECONOMIC AND SOCIAL STATISTICS, UNIVERSITÄTSRING 15, 54296 TRIER, GERMANY
Email address: burgardj@uni-trier.de, pamplona@uni-trier.de

Paper 2

Fair Generalized Linear Mixed Models

Jan Pablo Burgard, João Vitor Pamplona

Preprint under review.

URL: <https://arxiv.org/abs/2405.09273>

FAIR GENERALIZED LINEAR MIXED MODELS

JAN PABLO BURGARD, JOÃO VITOR PAMPLONA

ABSTRACT. When using machine learning for automated classification, it is important to account for fairness in the classification. Fairness in machine learning aims to ensure that biases in the data and model inaccuracies do not lead to discriminatory decisions. E.g., classifications from fair machine learning models should not discriminate against sensitive variables such as sexual orientation and ethnicity. The training data often is obtained from social surveys. In social surveys, oftentimes the data collection process is a strata sampling, e.g. due to cost restrictions. In strata samples, the assumption of independence between the observation is not fulfilled. Hence, if the machine learning models do not account for the strata correlations, the results may be biased. Especially high is the bias in cases where the strata assignment is correlated to the variable of interest. We present in this paper an algorithm that can handle both problems simultaneously, and we demonstrate the impact of stratified sampling on the quality of fair machine learning classifications in a reproducible simulation study.

1. INTRODUCTION

With the advent of automatic decision-making, the need for fair decision-making algorithms is steadily rising. The automatic decision should comply to restriction based on societal values, such as non-discrimination of parts of the population. Machine learning algorithms, while offering efficiency, can inadvertently perpetuate bias in critical areas like loan approvals (Das et al. 2021) and criminal justice (Green 2018). In loan applications, factors like marital status can lead to unfair disadvantages for single individuals, while in criminal justice, algorithms might associate race with recidivism risk, leading to discriminatory sentencing despite individual circumstances. This highlights the need for fair and unbiased AI frameworks to ensure equal opportunities and outcomes for all.

The training data, used to learn the machines for the automatic decision-making, oftentimes comes from surveys. These surveys typically are drawn according to a sampling plan, and hence do not comply with the general assumption in machine learning, that each unit is sampled independently and with the same probability of inclusion. For a detailed discussion on survey methods including sampling strategies, see Lohr (2009).

Date: December 3, 2024.

2020 Mathematics Subject Classification. 90C90, 90-08, 68T99 .

Key words and phrases. Logistic Regression, Fair Machine Learning, Mixed Models.

The field of fair machine learning has thrived, with numerous research articles exploring approaches to mitigate bias in various algorithms. Notable examples include fair versions of Logistic and Linear Regression (Berk et al. 2017), Support Vector Machines (Olfat and Aswani 2018), Random Forests (Zhang et al. 2021), Decision Trees (Aghaei et al. 2019), and Generalized Linear Models (GLMs) (Do et al. 2022). These methods aim to address potential discrimination arising from historical data or algorithmic design, ensuring fairer outcomes for all individuals.

In this paper we propose a Generalized Mixed Model for fair classifications. We show how to estimate the model and evaluate its performance against the current model that does not take the possible clustering of the data into account. As far as we know, this has not been proposed before.

The paper is organized as follows: In Section 2 we establish the theoretical underpinnings of fair generalized linear mixed models and propose a strategy for solving them. In Section 3, we conduct a comprehensive evaluation of our proposed method’s effectiveness through various tests. Finally, in Section 4, we demonstrate the practical applicability of our algorithm by solving a real-world problem using the Bank marketing dataset (Moro et al. 2012). Our key findings and potential future directions are presented in Section 5.

2. FAIR GENERALIZED LINEAR MIXED MODELS

In recent years, there has been a growing interest in developing fair machine learning algorithms. Fairness is a complex concept, but it generally refers to the idea that algorithms should not discriminate against certain groups of people (Caton and Haas 2020). This is important because algorithms are increasingly being used to make decisions about people’s lives, such as whether to grant them a loan or admit them to college.

In this context, we have Generalized Linear Models (GLMs) that are a class of models that can be used to model a variety of response variables, including count, continuous and binary data that is the focus of this work. Following the same idea, but with some changes, we have the Generalized Linear Mixed Models that allow for the inclusion of random effects, which are random variables that capture the variability in the response variable due to the hierarchical data structure. GLMMs are a powerful tool for analyzing data that are non-normal and hierarchical. They are used in a wide variety of fields, including medicine and psychology (Bono et al. 2021; Casals et al. 2014) for example. However, GLMMs, like many other statistical models, can lead to unfair outcomes.

2.1. Fair Classifications. In classification algorithms, we need to find a function that predicts the label y given a feature vector $x \in \mathbb{R}^p$. This function is learned on a training set $D = (x^\ell, y_\ell)_{\ell=1}^N$. One typically way to do that is minimizing a loss function $L(\beta)$ over a training set that minimize the classification error in this set.

In the context of fairness in binary classification, each observation ℓ has an associated sensitive feature $s_\ell \in \{0, 1\}$, and the objective then becomes finding a

solution with good accuracy (AC) while it is also fair. The concept of fairness in machine learning can be seen from different metrics, here, we study the concept of disparate impact (DI) (Barocas and Selbst 2016). Other unfairness metrics for binary classifiers can be found in Zafar et al. (2019).

Disparate impact refers to a situation where the classification of a model disproportionately harm points with different sensitive feature values. That is, a classifier is considered fair with respect to disparate impact if the probability, of the model (\mathbb{P}_m), of its prediction remains constant across both values of the sensitive feature s , i.e.,

$$\mathbb{P}_m(\hat{y}_\ell = 1 | s_\ell = 0, X) = \mathbb{P}_m(\hat{y}_\ell = 1 | s_\ell = 1, X).$$

We can also say that an algorithm suffers from disparate impact if the decision-making process grants a disproportionately large fraction of beneficial outcomes to certain sensitive feature groups.

2.2. Generalized linear mixed model. Generalized linear mixed models are regarded as an extension of generalized linear models that effectively incorporate random effects. These random effects can come from a survey that has strata bias, for example. This section provides a brief explanation of how we can model the GLMMs as can be seen in Stroup (2012).

Let y_{ij} denote the label in the observation j in strata i , where $i \in [1, n] := \{1, \dots, n\}$ and $j \in [1, T_i]$, with T_i being the size of the strata i . These observations are collected in the vector $y_i^\top = (y_{i_1}, \dots, y_{i_{T_i}})$. Let $(x^{ij})^\top = (x_1^{ij}, \dots, x_p^{ij})$ represent the covariate vector associated with fixed effects, and $(z^{ij})^\top = (z_1^{ij}, \dots, z_n^{ij}) \in \mathbb{R}^n$ denote the covariate vector associated with the random effects $b_i \in [1, n]$ that follow a normal distribution with a covariance matrix $Q_b = \text{Blockdiag}(Q, \dots, Q) \in \mathbb{R}^{n \times n}$.

The generalized linear mixed model can be expressed as follows:

$$g(\mu_{ij}) = \beta_0 + \beta^\top x^{ij} + (z^{ij})^\top b_i. \quad (1)$$

Here, g represents a monotonic and continuously differentiable link function, $\mu_{ij} = E(y_{ij} | b_i, x^{ij}, z^{ij})$, β_0 is the intercept, β the fixed effects and b_i represents the strata-specific random effects.

We can represent Model (1) using matrix notation. Let $(X^i)^\top = [x^{i_1}, \dots, x^{i_{T_i}}] \in \mathbb{R}^{p \times T_i}$ denote the design matrix for the i -th strata, and $\tilde{\beta}^\top = (\beta_0, \beta^\top)$ represent the linear parameter vector, including the intercept. Let $\tilde{X}^i = [\mathbf{1}, X^i] \in \mathbb{R}^{(p+1) \times T_i}$ be the corresponding matrix, where $\mathbf{1}^\top = (1, \dots, 1) \in \mathbb{R}^{T_i}$. By grouping the observations within each strata, the model can be represented as:

$$g(\mu_i) = \tilde{X}^i \tilde{\beta} + Z^i b_i,$$

where $(Z^i)^\top = [z^{i_1}, \dots, z^{i_{T_i}}] \in \mathbb{R}^{n \times T_i}$. For all observations one obtains

$$g(\mu) = \tilde{X} \tilde{\beta} + Z b,$$

with $\tilde{X} = [\tilde{X}^1, \dots, \tilde{X}^n] \in \mathbb{R}^{N \times (p+1)}$ and a block-diagonal matrix $Z = [Z^1, \dots, Z^n] \in \mathbb{R}^{N \times n}$, considering, w.l.o.g., that the first T_1 points, of \tilde{X} belong to strata 1, the next T_2 points belong to strata 2, and so on, that is, there is an ordering, by strata, in the data.

For $r \in [1, p]$ and $i \in [1, n]$, let us introduce the notation $(x^i)_r^\top = (x_r^{i1}, \dots, x_r^{iT_i}) \in \mathbb{R}^{T_i}$ to represent the covariate vector of the r -th fixed effect in strata i . Furthermore, we define $x_r^\top = ((x^1)_r^\top, \dots, (x^n)_r^\top) \in \mathbb{R}^N$. Consequently, the r -th design matrix, which includes the intercept and solely the r -th covariate vector, can be expressed as:

$$X_r^i = [\mathbf{1}, x_r^i] \in \mathbb{R}^{T_i \times 2}$$

and

$$X_r = [\tilde{\mathbf{1}}, x_r] \in \mathbb{R}^{N \times 2},$$

with

$$\tilde{\mathbf{1}} = (1, \dots, 1) \in \mathbb{R}^N$$

representing the design matrices for stratas i and the entire sample, respectively. Within strata i , the predictor that exclusively contains the r -th covariate takes the form of $\eta_r^i = X_r^i \tilde{\beta}_r + Z^i b_i$, where $\tilde{\beta}_r^\top = (\beta_0, \beta_r) \in \mathbb{R}^2$. For the entire sample, we obtain:

$$\eta_r = X_r \tilde{\beta}_r + Z b$$

and

$$\eta^i = X^i \beta + Z^i b_i.$$

Ignoring the mixed effects we can state that the logistics regression is a special case of GLMs. In the next chapters, we will see this case and some of its particularities for the case in which we have unfair datasets.

2.3. Fair Logistic Regression. In classification algorithms that employ logistic regression that can be seen in Neter et al. (2004), a probabilistic model is used to link a feature vector x to the class labels $y \in \{0, 1\}$. The link function is:

$$p(\hat{y} = 1|x, \beta) = m_\beta(x) = \frac{1}{1 + e^{-\beta^\top x}},$$

where β is obtained by solving the maximum likelihood problem on the training set (D), that is, $\beta^* = \operatorname{argmax}_\beta \sum_{(x,y) \in D} \log p(y|x, \beta)$. Therefore, as we are working with a minimization problem, the corresponding loss function is defined as $-\sum_{(x,y) \in D} \log p(y|x, \beta)$, and the complete optimization problem is formulated as follows:

$$\min_{\beta} - \sum_{\ell=1}^N [y_\ell \log(m_\beta(x^\ell)) + (1 - y_\ell) \log(1 - m_\beta(x^\ell))] \quad (2)$$

$$\text{s.t. } \frac{1}{N} \sum_{\ell=1}^N (s_\ell - \bar{s})(\beta^\top x^\ell) \leq c \quad (3)$$

$$\frac{1}{N} \sum_{\ell=1}^N (s_\ell - \bar{s})(\beta^\top x^\ell) \geq -c, \quad (4)$$

with

$$\bar{s} = \frac{\sum_{\ell=1}^N s_\ell}{N} \quad (5)$$

being s the sensitive feature and $c \in \mathbb{R}^+$ is a threshold that controls the importance of fairness. However, if c is chosen to be very small, the problem focuses exclusively on fairness, resulting in low accuracy.

In objective function (2), we have the original logistic regression objective function. Constraint (3) and (4) guarantees the fairness in the classification. The construction and justification of these constraints can be found in Zafar et al. (2015).

Now, using the logit link function, we can model a more specifically case of the GLMM problem and make it fair.

2.4. Fair Generalized linear mixed model. The goal of this section is to build upon existing results an algorithm that effectively handles both fairness considerations and the presence of random effects. This is done adjusting the Problem (2) - (4) to ensure that the model's classifications are not biased considering different stratas. We will discuss two proposals to address this problem. First, following the same strategy as Burgard and Pamplona (2024), we encounter the following constrained optimization problem:

$$\min_{\beta, b} - \sum_{i=1}^n \sum_{j=1}^{T_i} [y_{ij} \log(m_{\beta, b}(x^{ij})) + (1 - y_{ij}) \log(1 - m_{\beta, b}(x^{ij}))] + \lambda \sum_{i=1}^n b_i^2 \quad (6)$$

$$\text{s.t. } \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{T_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + b_i) \leq c \quad (7)$$

$$\frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{T_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + b_i) \geq -c \quad (8)$$

with

$$m_{\beta, b}(x^{ij}) = \frac{1}{1 + e^{-(\beta^\top x^{ij} + b_i)}},$$

\bar{s} from Equation (5). We denote the unconstrained Problem (6) as Cluster-Regularized Logistic Regression (CRLR) and problem (6) - (8) as Fair Cluster-Regularized Logistic Regression (Fair CRLR).

The second proposal is an adaptation of the boosting method proposed by Tutz and Groll (2010). Boosting methods represent a powerful ensemble technique in machine learning, designed to sequentially combine weak learners into a strong predictive model. By iteratively fitting new models from the previous iteration. This

process results in a highly accurate and robust model capable of handling complex patterns within the data (Schapire et al. 1999). Common boosting algorithms include AdaBoost (Freund, Schapire, et al. 1996), Gradient Boosting (Friedman 2001), and XGBoost (Chen and Guestrin 2016).

Since the method proposed by Tutz and Groll (2010) is based on Newton’s method (Lange 2002), it is crucial that the optimization problem is unconstrained. With this in mind, we convert this problem into an unconstrained problem using Lagrange’s penalty, this strategy can be seen in Nocedal (2006). To do this, we fix $c = 0$, which means, Problem (6) - (8) has a unique constraint. Observe that this is not a problem since we can control the constraint with the Lagrange multiplier ρ allowing a penalized violation of it. So, we have the problem:

$$\min_{\beta, b} - \sum_{i=1}^n \sum_{j=1}^{T_i} [y_{ij} \log(m_{\beta, b}(x^{ij})) + (1 - y_{ij}) \log(1 - m_{\beta, b}(x^{ij}))] + \lambda \sum_{i=1}^n b_i^2 + \frac{\rho}{N} \|a^\top \delta\|_2^2 \quad (9)$$

with $\delta^\top = (\beta_0, \beta, b^\top) \in \mathbb{R}^{1+p+n}$ and

$$a^\top = \left[\sum_{i=1}^n \sum_{j=1}^{T_i} (s_{ij} - \bar{s}) [1 \quad (x^{ij})^\top] \quad \sum_{j=1}^{T_i} (s_{1j} - \bar{s}) \dots \sum_{j=1}^{T_i} (s_{n_j} - \bar{s}) \right] \in \mathbb{R}^{1 \times (1+p+n)}.$$

If we use this same strategy in Fair Logistic Regression, we can see that, by transforming the constraint into a penalization, it still respects the improvement in disparate impact, with results that are very similar to the original problem. This gives us another indication that the strategy works. The boxplots below were created using the same strategy that will be seen in Chapter 3.

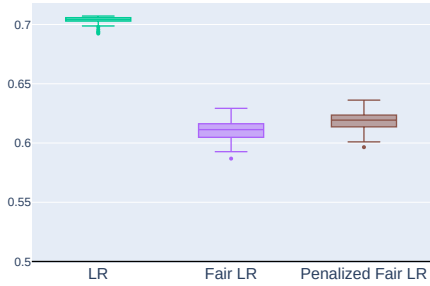


FIGURE 1. Accuracy.

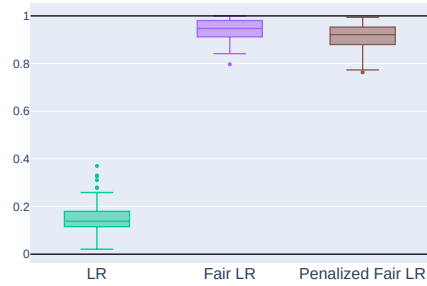


FIGURE 2. Disparate impact.

Similar to the propose of Tutz and Groll (2010) for solving the Problem (6), we do not explicitly solve the optimization problem (6) - (8). Instead, we propose an algorithm that, at each iteration, also updates the parameter λ , this stands in contrast to the traditional optimization problem where λ is treated as a fix parameter.

We can now update the components that we will utilize in the iterative process of Fair Generalized Linear Mixed Models (Fair GLMMs), a type of GLMM designed to

ensure fairness. In the following, we show in detail how to compute the components necessary for our algorithm based on the approach proposed by Tutz and Groll (2010).

For $r \in [1, p]$ and for $l \in [1, l_{max}]$ with l_{max} being the maximum number of iterations and the closed form for the pseudo Fisher matrix (FP)

$$FP_r(\delta^{(l)}) = A_r W_l A_r + K \in \mathbb{R}^{(n+2) \times (n+2)}, \quad (10)$$

with $A_r := [X_r, Z] \in \mathbb{R}^{N \times (n+2)}$, $K = \text{Blockdiag}(0, 0, Q^{-1}, \dots, Q^{-1}) \in \mathbb{R}^{(n+2) \times (n+2)}$ being a block diagonal penalty matrix with diagonal of two zeros corresponding to intercept and the r -th fixed effect and n times the matrix Q^{-1} , and given that we have a new objective function, we need to update the pseudo Fisher matrix. By the Equation (9), we have the penalization part being

$$\frac{\rho}{N} \|a^\top \delta\|_2^2. \quad (11)$$

As we can see in the maximization problem in Green and Silverman (1993), we can substitute the pseudo Fisher matrix with the negative of the Hessian matrix (\mathcal{H}). However, since we are working with a minimization problem, equality is automatically satisfied, i.e. $FP(\delta^{(l)}) = \mathcal{H}(\delta^{(l)})$. Hence, the Hessian of the penalization is:

$$\mathcal{H}\left(\frac{\rho}{N} \|a^\top \delta\|_2^2\right) = \frac{\rho}{N} a^\top a. \quad (12)$$

Then we can obtain the final Hessian (\mathcal{FH}) of the objective function (9) joining the Equations (10) and (12) and considering, w.l.o.g., $\rho = \frac{\rho}{N}$

$$\mathcal{FH}_r^{(l)} = A_r W_l A_r + K + \rho a_r^\top a_r, \quad (\text{FH})$$

with

$$W_l = w(\delta^{(l-1)}),$$

and

$$w(\delta) = D(\delta) \Sigma^{-1}(\delta) D(\delta)^\top \in \mathbb{R}^{N \times N},$$

where D is the derivative of the inverse of the link function. In our case we use the logit function, then $h(\eta^i) = g^{-1}(\eta^i) = e^{\eta^i} / (1 + e^{\eta^i})$, for $i \in [1, n]$ that is

$$\begin{aligned} D_i(\delta) &= \frac{\partial h(\eta^i)}{\partial \eta} \\ &= \frac{e^{\eta^i}}{(1 + e^{\eta^i})^2} \\ &= \frac{e^{\eta^i} + e^{2\eta^i} - e^{2\eta^i}}{(1 + e^{\eta^i})^2} \\ &= \frac{e^{\eta^i} (1 + e^{\eta^i}) - e^{2\eta^i}}{(1 + e^{\eta^i})^2} \\ &= \frac{(1 + e^{\eta^i})(e^{\eta^i} (1 + e^{\eta^i}) - e^{2\eta^i})}{(1 + e^{\eta^i})^3} \\ &= \frac{e^{\eta^i} (1 + e^{\eta^i})^2 - (1 + e^{\eta^i}) e^{2\eta^i}}{(1 + e^{\eta^i})^3} \end{aligned}$$

$$\begin{aligned}
&= \frac{e^{\eta^i}}{1 + e^{\eta^i}} - \frac{e^{2\eta^i}}{(1 + e^{\eta^i})^2} \\
&= \left(\frac{e^{\eta^i}}{1 + e^{\eta^i}} \right) \left(1 - \frac{e^{\eta^i}}{1 + e^{\eta^i}} \right) \\
&= h(\eta^i)(1 - h(\eta^i)).
\end{aligned} \tag{13}$$

Moreover, by Breslow and Clayton (1993),

$$\Sigma_i(\delta) = \text{cov}(y_i|\beta, b_i) = h(\eta^i)(1 - h(\eta^i)) \in \mathbb{R}^{T_i \times T_i}. \tag{14}$$

Combining (13) and (14) we obtain

$$D(\delta) = \Sigma(\delta),$$

being $D = \text{diag}(D_1, \dots, D_n)$, and then

$$\begin{aligned}
w(\delta) &= D(\delta)\Sigma^{-1}(\delta)D(\delta)^\top = D(\delta)D^{-1}(\delta)D(\delta)^\top \\
&= D(\delta)^\top \in \mathbb{R}^{N \times N}.
\end{aligned}$$

For the score function for $r \in [1, p]$, we have the closed form obtained by differentiating the objective function (6)

$$s_r(\delta^{(l)}) = A_r^\top W_l D_l^{-1}(y - \mu^{(l)}) - K \delta_r^{(l)} \in \mathbb{R}^{(n+2) \times 1} \tag{15}$$

being $\mu^{(l)} = h(\eta^{(l)}) \in \mathbb{R}^N$, $\delta_r^\top = (\beta_0, \beta_r, b^\top)$ and $D_l = D(\delta^{(l-1)})$ that come by differentiating the inverse of the logit function. Furthermore, the score function (6) can be interpreted as the negative of the gradient (∇), for the same reason as the Hessian, as it is calculated based on a maximization problem which we adapt here to a minimization problem. Therefore, the negative of the gradient of the penalization part (11) is:

$$-\nabla \left(\frac{\rho}{N} \|a^\top \delta\|_2^2 \right) = -2 \frac{\rho}{N} (a^\top a) \delta, \tag{16}$$

to obtain the final score function (\mathcal{FS}) of the objective function (9) we join the Equations (15) and (16) and considering, w.l.o.g., $\rho = 2 \frac{\rho}{N}$

$$\mathcal{FS}_r^{(l)} = s_r(\delta^{(l-1)}) - \rho a_r^\top (a_r \delta_r^{(l-1)}). \tag{FS}$$

For finding the best direction for the update we use the Bayesian Information Criterion (BIC) $\in \mathbb{R}^p$. The BIC is a popular model selection criterion for GLMMs for being relatively easy to calculate, and it has been shown to perform well in a variety of simulations as can be seen in Vrieze (2012):

$$BIC_r^{(l)} = -2\Omega(\mu_r^{(l)}) + 2\text{tr}(H_r^{(l)})\log(n),$$

with

$$\Omega(\mu_r^{(l)}) = \sum_{i=1}^n \sum_{j=1}^{T_i} y_{ij} \log(\mu_{ij}) + (1 - y_{ij})(1 - \log(\mu_{ij})) - \rho \|a^\top \delta_r^{(l)}\|_2^2,$$

that comes from the objective function, without considering the variance penalization of the random effects, as this is already considered in other stages of the

process, and the hat matrix corresponding to the l -th boosting step considering the r -th component,

$$H_r^{(l)} = I - (I - M_r^{(l)})(I - M_{l-1})(I - M_{l-2}) \dots (I - M_0) \in \mathbb{R}^{N \times N},$$

with M_k the matrix corresponding to the component that has been selected in the k -th iteration, for $k = 1, \dots, l-1$ being

$$M_r^{(l)} := \Sigma_l^{1/2} \tilde{H}_r^{(l)} \Sigma_l^{-1/2} \in \mathbb{R}^{N \times N},$$

and

$$M_0 := A_1(A_1^\top W_1 A_1 + K_1)A_1^\top W_1 \in \mathbb{R}^{N \times N},$$

being $\Sigma_l = \Sigma(\delta^{(l-1)})$, and an adaptation of the generalized ridge regression hat-matrix which considers the penalization of fairness constraints,

$$\tilde{H}_r^{(l)} = W_l^{1/2} A_r (A_r W_l A_r + K + \rho \|a^\top \delta_r^{(l)}\|_2^2)^{-1} A_r^\top W_l^{1/2} \in \mathbb{R}^{N \times N}.$$

Now, as can be seen in Harville (1977) we can update the covariance matrix $Q^{(l)}$ by

$$Q^{(l)} = \frac{1}{n} \sum_{i=1}^n (V_i^{(l)} + b_i^{(l)} (b_i^{(l)})^\top) \in \mathbb{R}.$$

In general, and in our case, the V_i are computed via the formula

$$V_i = F_i^{-1} + F_i^{-1} \tilde{F}_i^\top (\hat{F} - \sum_{i=1}^n \tilde{F}_i F_i^{-1} \tilde{F}_i^\top)^{-1} \tilde{F}_i F_i^{-1} \in \mathbb{R}$$

with

$$\begin{aligned} F_i &= (Z^i)^\top D_i(\delta) \Sigma_i(\delta)^{-1} D_i(\delta) Z^i + Q^{-1} \in \mathbb{R}, \\ \tilde{F}_i &= (X^i)^\top D_i(\delta) \Sigma_i(\delta)^{-1} D_i(\delta) Z^i \in \mathbb{R}^p, \end{aligned}$$

and

$$\hat{F} = \sum_{i=1}^n (X^i)^\top D_i(\delta) \Sigma_i(\delta)^{-1} D_i(\delta) X^i \in \mathbb{R}^{p \times p},$$

where \hat{F} , \tilde{F}_i and F_i are the elements of the pseudo Fisher matrix $FP(\delta)$ of the full model. For more detailed derivation see Gu et al. (2012).

Preliminary tests have shown that model (6) - (8) is an warm starting for algorithm 1. Consequently, we have the following initial parameters:

- $\beta_0^{(0)}$ from Fair CRLR;
- $\beta^{(0)}$ from Fair CRLR;
- $b^{(0)}$ from Fair CRLR;
- $\mu^{(0)} = 0 \in \mathbb{R}^N$;
- $\eta^{(0)} = 0 \in \mathbb{R}^N$;
- $Q^{(0)} = 2.0$.

Finally, we have all necessary components and motivations to propose an algorithm for solving the Fair Generalized Linear Mixed Model.

Algorithm 1 : Fair Generalized Linear Mixed Model	
Given: $\mu^{(0)}, \beta_0^{(0)}, \beta^{(0)}, b^{(0)}, \eta^{(0)}, l_{max}, Q^{(0)}$.	
Iteration:	
<p>(1) Refitting of residuals:</p> <p>For $l \in [1, l_{max}]$,</p> <p style="padding-left: 20px;">(i) Computation of parameters</p> <p style="padding-left: 20px;">For $r \in [1, p]$ the r-th Newton's method step has the form:</p> $\delta_r^{(l)} = (\mathcal{FH}_r^{(l-1)})^{-1}(\mathcal{FS}_r^{(l-1)})$ <p style="padding-left: 20px;">(ii) Selection step</p> <p style="padding-left: 20px;">Select from $r \in [1, p]$ the index j corresponding to the smallest $BIC_r^{(l)}$ and select the related $(\delta_j^{(l)})^\top = (\beta_0^*, \beta_j^*, (b^*)^\top)$.</p> <p style="padding-left: 20px;">(iii) Update</p> <p style="padding-left: 20px;">Set</p> $\beta_0^{(l)} = \beta_0^{(l-1)} + \beta_0^*$ <p style="padding-left: 20px;">and</p> $b^{(l)} = b^{(l-1)} + b^*$ <p style="padding-left: 20px;">and for $r \in [1, p]$ set</p> $\beta_r^{(l)} = \begin{cases} \beta_r^{(l-1)} & \text{if } r \neq j \\ \beta_r^{(l-1)} + \beta_r^* & \text{if } r = j \end{cases}$ <p style="padding-left: 20px;">with $A := [X, Z]$ update</p> $\eta = A\delta^{(l)}$	<p>(2) Computing of variance-covariance components:</p> $Q^{(l)} = \frac{1}{n} \sum_{i=1}^n (V_i^{(l)} + b_i^{(l)}(b_i^{(l)})^\top).$
until $Q^{(l)} = Q^{(l-1)}$.	

The following section presents various numerical tests demonstrating the efficacy of our proposed methods.

3. SIMULATION STUDY

In this section, we aim to present numerical tests to demonstrate the effectiveness of the proposed method.

First we present the step-by-step strategy used to create the datasets and to conduct the numerical experiments. Using the `Julia 1.9` (Bezanson et al. 2017) language with the packages `Distributions` (Besançon et al. 2021), `GLM` (Bates et al. 2022), `MixedModels` (Bates et al. 2023), `DataFrames` (Bouchet-Valat and Kamiński 2023), `MLJ` (Blaom et al. 2019) and `MKL` (Corporation 2023), we generate the following parameters:

- *Number of points*: Number of points in the dataset;
- β' s: The fixed effects;
- b' s: The random effects with distribution $N(0, Q)$, with covariance matrix Q (Used only when indicated);

- *Data points*: The covariate vector associated with fixed effects with distribution $N(0, 1)$;
- *c*: Threshold from Fair Logistic Regression;
- ρ : Lagrange multiplier;
- *seed*: Random seed used in the generation of data;
- *Train-Test split*: Approximately 0.4% of the dataset was used for the training set, and 99.6% for the test set. This percentage was due to the fact that we randomly selected 3 to 5 points from each strata for the training set.

The labels of the synthetic dataset were computed using

$$m = \frac{1}{1 + e^{-(\beta^T x + b)}} \quad (17)$$

in tests where the dataset has random effects, and

$$m = \frac{1}{1 + e^{-\beta^T x}} \quad (18)$$

in tests where the dataset has just fixed effects. Finally,

$$y = \text{Binomial}(1, m).$$

The comparisons are made by comparing the accuracy and the disparate impact between the Algorithms:

- (1) Generalized linear mixed model (GLMM);
- (2) Fair generalized linear mixed model (Fair GLMM);
- (3) Cluster-Regularized Logistic Regression (CRLR);
- (4) Fair Cluster-Regularized Logistic Regression (Fair CRLR),
- (5) Logistic Regression (LR);
- (6) Fair Logistic Regression (Fair LR).

The tests were conducted on a laptop with an Intel Core i9-13900HX processor with a clock speed of 5.40 GHz, 64 GB of RAM, and Windows 11 operating system, with 64-bit architecture.

To compute accuracy, first we need to compute the classifications using Equations (17) and (18) with,

$$\hat{y} = \begin{cases} 1 & \text{if } m \geq 0.5 \\ 0 & \text{if } m < 0.5 \end{cases}.$$

Given the true label of all points, we can distinguish them into four categories: true positive (TP) or true negative (TN) if the point is classified correctly in the positive or negative class, respectively, and false positive (FP) or false negative (FN) if the point is misclassified in the positive or negative class, respectively. Based on this, we can compute the accuracy, where a higher value indicates a better classification, as follows,

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1].$$

To compute the disparate impact of a specific sensitive feature s we use the following equation as can be seen in Radovanović et al. (2020),

$$DI = \frac{p(\hat{y}|s=1)}{p(\hat{y}|s=0)} \in [0, \infty).$$

Disparate impact, as a measure, should be equal to 1. This indicates that discrimination does not exist. Values greater or lower than 1 suggest that unwanted discrimination exists. However, $DI = 2$ and $DI = 0.5$ represent the same level of discrimination, although in the first case, the difference between the perfect value is 1, and in the latter case, it is 0.5. To avoid such situations, we use the minimum of DI and its inverse.

$$DI = \min\left(\frac{p(\hat{y}|s=1)}{p(\hat{y}|s=0)}, \frac{p(\hat{y}|s=0)}{p(\hat{y}|s=1)}\right) \in [0, 1].$$

In the following, we generate four different synthetic populations (scenarios) to compare the competing algorithms. For each synthetic data set 100 samples are drawn. The simulation results are discussed for each synthetic data set using 2 images that represent, respectively, the accuracy and the disparate impact.

All figures were created using the `Plots` and `PlotlyJS` packages, developed by Christ et al. (2023) and all hyperparameters were selected via cross-validation (Browne 2000). All tests can be reproduced, and the codes of all functions used can be found in [GitHub](#).

3.1. Unfair population with strata effect. Parameters of data generation:

- β 's = $[-2.0, 0.4, 0.8, 0.5, 3.0]$;
- b 's: 100 stratas with $b_i \sim N(0, 3.0)$, with $i \in [1, 100]$;
- $c = 0.1$;
- $\rho = 0.8$;
- $\lambda = 1$.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.89	0.88	0.89	0.90	0.91	0.01
Fair GLMM	0.82	0.82	0.82	0.83	0.83	0.01
CRLR	0.79	0.78	0.79	0.80	0.81	0.02
Fair CRLR	0.82	0.81	0.82	0.83	0.86	0.02
LR	0.68	0.68	0.68	0.68	0.68	0.01
Fair LR	0.64	0.63	0.64	0.64	0.66	0.01

TABLE 1. Accuracy.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.48	0.45	0.48	0.51	0.56	0.05
Fair GLMM	0.91	0.90	0.91	0.92	0.94	0.02
CRLR	0.18	0.14	0.18	0.22	0.27	0.06
Fair CRLR	0.66	0.63	0.66	0.70	0.75	0.05
LR	0.10	0.04	0.09	0.14	0.21	0.07
Fair LR	0.52	0.48	0.53	0.59	0.65	0.11

TABLE 2. Disparate impact

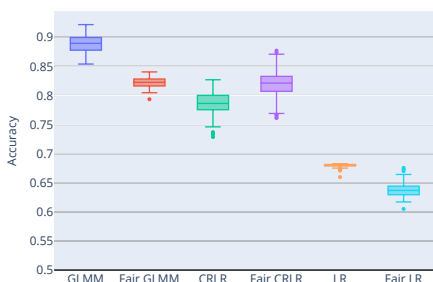


FIGURE 3. Accuracy.

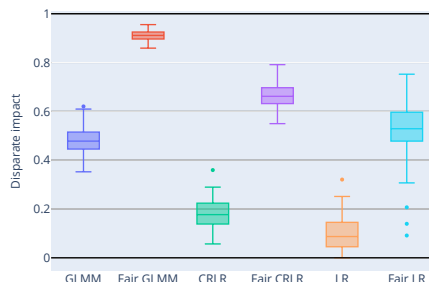


FIGURE 4. Disparate impact.

The results in Table 1 and Figure 3 demonstrate superior accuracy for GLMM, Fair GLMM, CRLR, and Fair CRLR in this experiment set. This is unsurprising, as logistic regression doesn't account for random effects. We can also see that the GLMM performs better on both metrics compared to the optimization problems.

Table 2 and Figure 4 show that we also obtained an improvement in the disparate impact on the Fair algorithms. Note that make sense since we have a unfair population.

3.2. Fair population with strata effect. Parameters of data generation:

- β 's = $[-0.1, 1, 1, 1, 0.1]$;
- b 's: 100 stratas with $b_i \sim N(0, 3.0)$, with $i \in [1, 100]$;
- $c = 0.1$;
- $\rho = 0.8$;
- $\lambda = 1$.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.87	0.86	0.89	0.90	0.90	0.06
Fair GLMM	0.87	0.87	0.88	0.88	0.89	0.02
CRLR	0.81	0.80	0.81	0.82	0.83	0.02
Fair CRLR	0.81	0.80	0.81	0.82	0.83	0.02
LR	0.66	0.66	0.66	0.66	0.66	0.01
Fair LR	0.66	0.66	0.66	0.66	0.66	0.01

TABLE 3. Accuracy.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.96	0.94	0.97	0.99	0.99	0.03
Fair GLMM	0.98	0.97	0.98	0.99	0.99	0.02
CRLR	0.91	0.86	0.92	0.96	0.98	0.07
Fair CRLR	0.92	0.88	0.93	0.96	0.98	0.05
LR	0.88	0.82	0.90	0.95	0.99	0.09
Fair LR	0.89	0.84	0.91	0.95	0.99	0.07

TABLE 4. Disparate impact

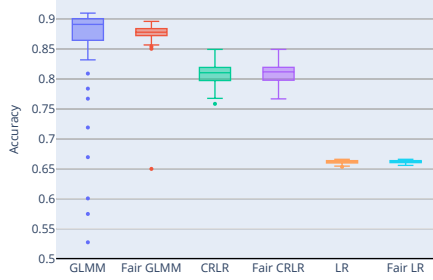


FIGURE 5. Accuracy.

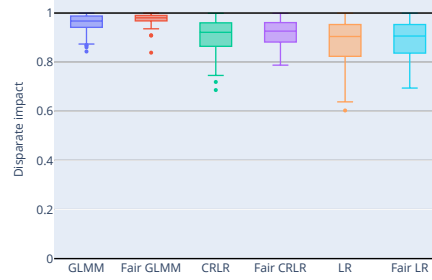


FIGURE 6. Disparate impact.

As can be seen in Table 3 and in the Figure 5, in this set of experiments, we obtained a better accuracy in GLMM, Fair GLMM, CRLR and Fair CRLR. This is expected since logistic regression does not account for random effects.

Table 4 and Figure 6 show that the disparate impact remains almost the same in all tests since we have a fair population.

3.3. Unfair population without strata effect. Parameters of data generation:

- β 's = $[-2.0, 0.4, 0.8, 0.5, 3.0]$;
- $\rho = 0.8$;
- $c = 0.1$;
- $\lambda = 1$.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.70	0.70	0.71	0.72	0.73	0.03
Fair GLMM	0.61	0.61	0.62	0.62	0.63	0.01
CRLR	0.79	0.78	0.79	0.79	0.79	0.01
Fair CRLR	0.68	0.67	0.68	0.69	0.70	0.01
LR	0.79	0.79	0.79	0.79	0.79	0.01
Fair LR	0.69	0.68	0.69	0.70	0.71	0.01

TABLE 5. Accuracy.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.17	0.12	0.15	0.17	0.25	0.15
Fair GLMM	0.73	0.70	0.74	0.78	0.82	0.07
CRLR	0.04	0.02	0.04	0.05	0.08	0.02
Fair CRLR	0.56	0.51	0.57	0.62	0.67	0.08
LR	0.04	0.02	0.03	0.05	0.07	0.02
Fair LR	0.52	0.47	0.53	0.59	0.64	0.09

TABLE 6. Disparate impact

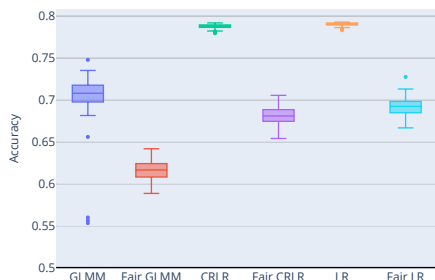


FIGURE 7. Accuracy.

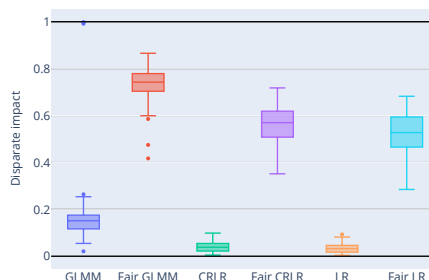


FIGURE 8. Disparate impact.

Our experiments, detailed in Table 5 and Figure 7, show a slight decrease in accuracy for GLMM algorithms. This is likely because GLMMs attempt to account for random effects, even when they are absent. This additional parameter estimation in GLMMs can lead to a reduction in accuracy compared to logistic regression optimization problems.

Table 6 and Figure 8 show that we obtained an improvement in the disparate impact on the Fair algorithms. Note that make sense since we have a unfair population.

3.4. Fair population without strata effect. Parameters of data generation:

- β 's = $[-0.1, 1, 1, 1, 0.1]$;
- $c = 0.1$;
- $\rho = 0.8$;
- $\lambda = 1$.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.63	0.61	0.63	0.67	0.68	0.04
Fair GLMM	0.67	0.67	0.68	0.68	0.69	0.02
CRLR	0.75	0.75	0.75	0.75	0.75	0.01
Fair CRLR	0.75	0.75	0.75	0.75	0.75	0.01
LR	0.75	0.75	0.75	0.75	0.76	0.01
Fair LR	0.75	0.75	0.75	0.75	0.76	0.01

TABLE 7. Accuracy.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.88	0.82	0.89	0.95	0.99	0.09
Fair GLMM	0.94	0.91	0.94	0.97	0.99	0.04
CRLR	0.90	0.85	0.90	0.95	0.99	0.07
Fair CRLR	0.91	0.86	0.91	0.95	0.99	0.06
LR	0.89	0.85	0.90	0.94	0.99	0.06
Fair LR	0.91	0.86	0.91	0.95	0.99	0.06

TABLE 8. Disparate impact

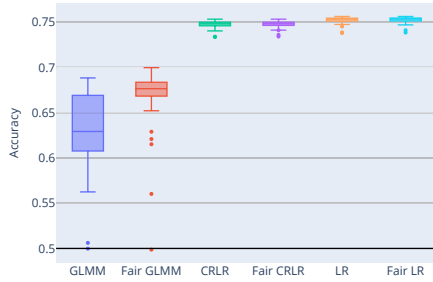


FIGURE 9. Accuracy.

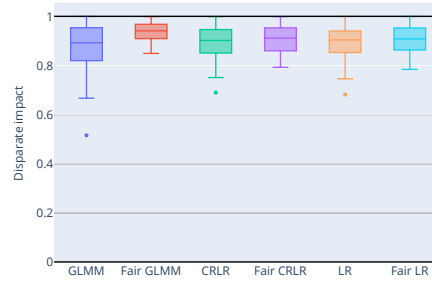


FIGURE 10. Disparate impact.

Our experiments, detailed in Table 7 and Figure 9, show a decrease in accuracy for GLMM algorithms for the same reason from Scenario 3.3.

Table 8 and Figure 10 show that we also have a very similar disparate impact in all algorithms. Note that make sense since we have a fair population.

4. APPLICATION

In this set of experiments, we test the Bank marketing dataset, which has a strata bias related to the duration of telephone calls, the longer the call duration, (calls with longer duration imply a higher probability of the label being 1), and a sensitive feature, which in this case is housing loan, the housing loan feature can be considered a sensitive feature because it is directly linked to injustice in its generation, as can be seen in Brooke (2023) and in Howell (2006). The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact was required with the same client, in order to determine if the product (bank term deposit) would be subscribed ($y = 1$) or not ($y = 0$). Were conducted 100 samples with 3.5% of the data as the training set and the remaining data as the test set. Since the application under study employs random effects, we conduct the comparisons using GLMM, the cluster-regularized logistic regression and the regular logistic regression.

The features used in the classification process are the follows:

- Age
- Job
- Marital status
- Education
- Has credit in default?
- Has housing loan?
- Has personal loan?
- Contact communication type
- Last contact month of year
- Last contact day of the week
- Number of contacts performed during this campaign and for this client
- Number of days that passed by after the client was last contacted from a previous campaign
- Number of contacts performed before this campaign and for this client
- Outcome of the previous marketing campaign
- Employment variation rate
- Consumer price index
- Consumer confidence index
- Euribor 3 month rate
- Number of employees

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.68	0.68	0.69	0.71	0.72	0.05
Fair GLMM	0.67	0.68	0.69	0.70	0.71	0.05
CRLR	0.65	0.64	0.65	0.66	0.68	0.01
Fair CRLR	0.65	0.64	0.65	0.66	0.68	0.01
LR	0.57	0.56	0.57	0.58	0.59	0.01
Fair LR	0.56	0.55	0.56	0.57	0.58	0.01

TABLE 9. Accuracy.

Algorithm	Mean	p25	Median	p75	p95	std
GLMM	0.77	0.69	0.78	0.85	0.99	0.12
Fair GLMM	0.87	0.83	0.86	0.90	0.99	0.06
CRLR	0.47	0.37	0.46	0.57	0.72	0.15
Fair CRLR	0.62	0.56	0.60	0.66	0.79	0.09
LR	0.36	0.24	0.31	0.45	0.71	0.18
Fair LR	0.50	0.38	0.48	0.60	0.77	0.15

TABLE 10. Disparate impact

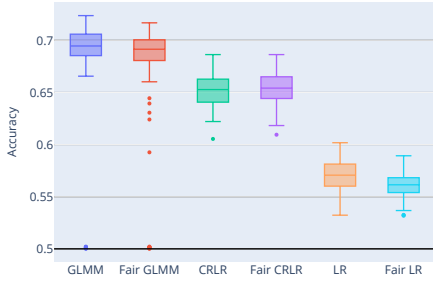


FIGURE 11. Accuracy.

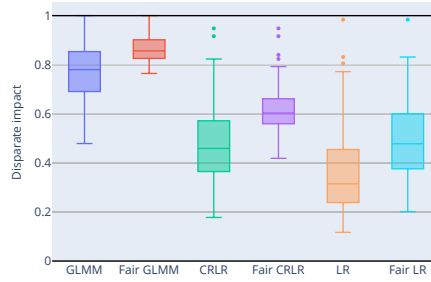


FIGURE 12. Disparate Impact.

We can see that the application is part of scenario 3.1. Thus, can be seen in Table 9 and in the Figure 11 results similar to those previously generated synthetically for the same scenario. That is, we obtained a better accuracy in models that can deal with heterogeneous populations.

Table 10 and Figure 12 show that we also obtained an improvement in the disparate impact on the Fair algorithms. Note that make sense since we have a unfair population.

4.1. Sensitivity analysis. The Lagrange multiplier strategy, using Karush-Kuhn-Tucker (KKT) conditions, is a powerful tool for analyzing the sensitivity of constrained optimization problems. To perform the sensitivity analysis of the constraints of problems, we use the strategy, described in Appendix A.

As seen in Section 2.3 fairness constraints operate within an interval $[-c, c]$. However, we set $c = 0$. Because this value in the constraint makes the optimization problem as fair as possible, and then, we can say that if we do not obtain a large perturbation in the objective function with $c = 0$, the constraint is useless for the problem. This is because, with other values of c , the perturbation would be even smaller.

The sensitivity analysis (SA) or, as it is also known, Lagrange multipliers of fair logistic regression problems is important for understanding how the fairness constraints affect the model's classifications. By analyzing the sensitivity of the

model’s predictions to changes in the fairness constraints, we can identify which constraints, among the constraints of sensitive features, are most important for achieving fairness. We can also refer to this value as the shadow price, in essence, the shadow price reflects the economic value of relaxing or tightening a constraint in an optimization problem. It represents the marginal impact on the objective function of making a small adjustment to a constraint, that is if the shadow price is P , it means that the value of the objective function increase by P if the constraint is relaxed as we can see in Smith (1987).

In this application, we consider Marital Status, Education, and Housing Loan as sensitive features. The values below refer to the Lagrange multiplier values and the disparate impact values found in the 100 tests performed on this application. The disparate impact improvement are compared to the regular logistic regression and the accuracy drop are compared with the regular GLMM.

Sensitivity Feature	SA	DI improve	AC drop
Housing	28.7	134.3%	7.5%
Marital Status	11.9	14.4%	< 1.0%
Education	12.5	16.6%	< 1.0%
Marital Status/Education	13.5/14.7	13.9%/18.5%	< 1.0%
Housing/Marital Status	30.9/14.4	86.5%/9.8%	7.9%
Housing/Education	29.3/13.8	82.5%/16.0%	7.9%

TABLE 11. Sensitivity analysis table.

As we can see in Table 11, the sensitive feature constraint that made the biggest difference in the objective function and, therefore, achieved a greater improvement in the Disparate Impact was the housing feature.

We can also note, that in this application simply accounting for the random effect without adding fairness constraints already significantly improves the disparate impact. As expected, adding the fairness constraints, yields even better disparate impact.

5. CONCLUSION

In this work, we proposed an algorithm for a fair generalized linear mixed model (GLMM) and a optimization model (CRLR) that allows for controlling the disparate impact of a sensitive feature. This way, a fair classification can be achieved even when the data has an inherent grouping structure. To our knowledge, this has not been proposed before.

We leverage simulations to showcase how our approach overcomes limitations in existing methods. It delivers superior results when group structures significantly impact classification accuracy or fairness. We also concluded when to use which approach, considering that if we have the information about random effects, the Fair GLMM performs better, otherwise the optimization model is also a good choice.

Furthermore, we applied our method to the Bank marketing dataset. Here, it effectively addressed random effects while mitigating disparate impact associated with the sensitive feature.

Additionally, we explore how KKT conditions can be used to assess feature sensitivity in fair logistic regression. This analysis helps identify the specific sensitive variable whose disparate impact we aimed to mitigate in the process.

To enhance the framework’s capabilities, future work could focus on incorporating additional fairness constraint options within the GLMM. Additionally, efforts to optimize the computational efficiency of the proposed algorithms would be beneficial.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the German Federal Ministry of Education and Research (BMBF) for this research project, as well as for the “OptimAgent Project”.

We would also like to express our sincere appreciation for the generous support provided by the German Research Foundation (DFG) within Research Training Group 2126 “Algorithmic Optimization”.

REFERENCES

- Aghaei, S, M. J. Azizi, and P Vayanos (2019). “Learning optimal and fair decision trees for non-discriminative decision-making.” In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01, pp. 1418–1426.
- Barocas, S and A. D. Selbst (2016). “Big data’s disparate impact.” In: *Calif. L. Rev.* 104, p. 671.
- Bates, D, P Alday, D Kleinschmidt, J. B. S. Calderón, L Zhan, A Noack, M Bouchet-Valat, A Arslan, T Kelman, A Baldassari, B Ehinger, D Karrasch, E Saba, J Quinn, M Hatherly, M Piibeleht, P. K. Mogensen, S Babayan, T Holy, Y. L. Gagnon, and Y Nazarathy (Nov. 2023). “JuliaStats/MixedModels.jl: v4.22.2.” Version v4.22.2. In: DOI: [10.5281/zenodo.10069987](https://doi.org/10.5281/zenodo.10069987).
- Bates, D, A Noack, S Kornblith, M Bouchet-Valat, M. K. Borregaard, A Arslan, J. M. White, D Kleinschmidt, G Lynch, I Dunning, P. K. Mogensen, S Lendle, D Aluthge, P Alday, J. B. S. Calderón, A Patnaik, B Born, B Setzler, C DuBois, J Quinn, O Slámečka, P Bastide, V. B. Shah, A Blaom, B König, B Kamiński, and C Caine (2022). “JuliaStats/GLM.jl: v1.8.0.” Version v1.8.0. In: DOI: [10.5281/zenodo.6580436](https://doi.org/10.5281/zenodo.6580436).
- Berk, R, H Heidari, S Jabbari, M Joseph, M Kearns, J Morgenstern, S Neel, and A Roth (2017). “A convex framework for fair regression.” In: *arXiv preprint arXiv:1706.02409*.
- Bertsekas, D. P., A Nedic, and A Ozdaglar (2003). *Convex analysis and optimization*. Vol. 1. Nashua: Athena Scientific.

- Besançon, M, T Papamarkou, D Anthoff, A Arslan, S Byrne, D Lin, and J Pearson (2021). “Distributions.jl: Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem.” In: *Journal of Statistical Software* 98.16, pp. 1–30. DOI: [10.18637/jss.v098.i16](https://doi.org/10.18637/jss.v098.i16).
- Bezanson, J, A Edelman, S Karpinski, and V. B. Shah (2017). “Julia: A fresh approach to numerical computing.” In: *SIAM review* 59.1, pp. 65–98. DOI: [10.1137/14100067](https://doi.org/10.1137/14100067).
- Blaom, A, F Kiraly, T Lienart, and S Vollmer (2019). “alan-turing-institute/MLJ.jl: v0.5.3.” Version v0.5.3. In: DOI: [10.5281/zenodo.3541506](https://doi.org/10.5281/zenodo.3541506).
- Bono, R, R Alarcón, and M. J. Blanca (2021). “Report Quality of Generalized Linear Mixed Models in Psychology: A Systematic Review.” In: *Frontiers in Psychology* 12. DOI: [10.3389/fpsyg.2021.666182](https://doi.org/10.3389/fpsyg.2021.666182).
- Bouchet-Valat, M and B Kamiński (2023). “DataFrames.jl: Flexible and Fast Tabular Data in Julia.” In: *Journal of Statistical Software* 107.4, pp. 1–32. DOI: [10.18637/jss.v107.i04](https://doi.org/10.18637/jss.v107.i04).
- Breslow, N. E. and D. G. Clayton (1993). “Approximate Inference in Generalized Linear Mixed Models.” In: *Journal of the American Statistical Association* 88.421, pp. 9–25.
- Brooke, E (2023). *Fair Housing Trends Report*. <https://nationalfairhousing.org/resource/2023-fair-housing-trends-report/>. Accessed: 2023-11-13.
- Browne, M. W. (2000). “Cross-validation methods.” In: *Journal of mathematical psychology* 44.1, pp. 108–132.
- Burgard, J. P. and J. V. Pamplona (2024). *Fair Mixed Effects Support Vector Machine*. arXiv: [2405.06433](https://arxiv.org/abs/2405.06433) [cs.LG].
- Casals, M, M Girabent-Farrés, and J. L. Carrasco (2014). “Methodological Quality and Reporting of Generalized Linear Mixed Models in Clinical Medicine (2000–2012): A Systematic Review.” In: *PLoS ONE* 9.
- Caton, S and C Haas (2020). “Fairness in machine learning: A survey.” In: *ACM Computing Surveys*.
- Chen, T and C Guestrin (2016). “Xgboost: A scalable tree boosting system.” In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.
- Christ, S, D Schwabeneder, C Rackauckas, M. K. Borregaard, and T Breloff (2023). “Plots.jl – a user extendable plotting API for the julia programming language.” In: DOI: <https://doi.org/10.5334/jors.431>.
- Corporation, I. (2023). “Intel Math Kernel Library.” In.
- Das, S, M Donini, J Gelman, K Haas, M Hardt, J Katzman, K Kenthapadi, P Larooy, P Yilmaz, and M. B. Zafar (2021). “Fairness measures for machine learning in finance.” In: *The Journal of Financial Data Science*.
- Do, H, P Putzel, A. S. Martin, P Smyth, and J Zhong (2022). “Fair generalized linear models with a convex penalty.” In: *International Conference on Machine Learning*. PMLR, pp. 5286–5308.

- Freund, Y, R. E. Schapire, et al. (1996). “Experiments with a new boosting algorithm.” In: *icml*. Vol. 96. Citeseer, pp. 148–156.
- Friedman, J. H. (2001). “Greedy function approximation: a gradient boosting machine.” In: *Annals of statistics*, pp. 1189–1232.
- Green, B (2018). “Fair” risk assessments: A precarious approach for criminal justice reform.” In: *5th Workshop on fairness, accountability, and transparency in machine learning*, pp. 1–5.
- Green, P. J. and B. W. Silverman (1993). *Nonparametric Regression and Generalized Linear Models: A roughness penalty approach*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Florida: CRC Press.
- Gu, Q, Z Li, and J Han (2012). “Generalized fisher score for feature selection.” In: *arXiv preprint arXiv:1202.3725*.
- Harville, D. A. (1977). “Maximum likelihood approaches to variance component estimation and to related problems.” In: *Journal of the American statistical association* 72.358, pp. 320–338.
- Hilbe, J. M. (2009). *Logistic regression models*. Florida: CRC press.
- Howell, B (2006). “Exploiting Race and Space: Concentrated Subprime Lending as Housing Discrimination.” In: *California Law Review* 94.1, pp. 101–147.
- Lange, K (2002). “Newton’s Method and Scoring.” In: *Mathematical and Statistical Methods for Genetic Analysis*. New York, NY: Springer New York, pp. 39–58. DOI: [10.1007/978-0-387-21750-5_3](https://doi.org/10.1007/978-0-387-21750-5_3).
- Lohr, S. L. (Feb. 2009). *Sampling : Design and Analysis*. 2nd ed. Florence, KY: Brooks/Cole.
- Moro, S, P Rita, and P Cortez (2012). *Bank Marketing*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5K306>.
- Neter, D. J., M. H. Kutner, and C. J. Nachtsheim (2004). *MP Applied Linear Regression Models-Revised Edition with Student CD*. Dubuque: McGraw-Hill Education.
- Nocedal, J (2006). “Penalty and Augmented Lagrangian Methods.” In: *Numerical Optimization*. New York: Springer New York, pp. 497–528.
- Olfat, M and A Aswani (2018). “Spectral algorithms for computing fair support vector machines.” In: *International conference on artificial intelligence and statistics*. PMLR, pp. 1933–1942.
- Radovanović, S, A Petrović, B Delibašić, and M Suknović (2020). “Enforcing fairness in logistic regression algorithm.” In: *2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–7. DOI: [10.1109/INISTA49547.2020.9194676](https://doi.org/10.1109/INISTA49547.2020.9194676).
- Schapire, R. E. et al. (1999). “A brief introduction to boosting.” In: *Ijcai*. Vol. 99. 999. Citeseer, pp. 1401–1406.
- Smith, A (1987). “Shadow price calculations in distorted economies.” In: *The Scandinavian Journal of Economics*, pp. 287–302.

- Stroup, W. W. (2012). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Chapman & Hall/CRC Texts in Statistical Science. Oxfordshire: Taylor & Francis.
- Tutz, G and A Groll (2010). “Generalized linear mixed models based on boosting.” In: *Statistical Modelling and Regression Structures: Festschrift in Honour of Ludwig Fahrmeir*, pp. 197–215.
- Vrieze, S. I. (2012). “Model selection and psychological theory: a discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).” In: *Psychological methods* 17.2, p. 228.
- Zafar, M, I Valera, M Rodriguez, and K. P. Gummadi (July 2015). “Fairness Constraints: A Mechanism for Fair Classification.” In.
- Zafar, M. B., I Valera, M Gomez-Rodriguez, and K. P. Gummadi (2019). “Fairness Constraints: A Flexible Approach for Fair Classification.” In: *Journal of Machine Learning Research* 20.75, pp. 1–42.
- Zhang, W, A Bifet, X Zhang, J. C. Weiss, and W Nejdil (2021). “Farf: A fair and adaptive random forests classifier.” In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer, pp. 245–256.

APPENDIX A. SENSITIVITY ANALYSIS

The generalization of the sensitivity analysis to multiple sensitive features is straightforward. We simply add a Lagrange multiplier for each sensitive feature. The sensitivity of the objective function can then be determined by the corresponding Lagrange multipliers ζ . Each ζ_k is the perturbation of the k -th sensitive feature, with $k = 1, \dots, K$, where K is the number of sensitive features that we want to use in the tests. And for each sensitive feature, we must use the corresponding s^k and \bar{s}^k .

The sensitivity analysis of fair logistic regression problems is important for understanding how the fairness constraints affect the model’s classifications. By analyzing the sensitivity of the model’s predictions to changes in the fairness constraints, we can identify which constraints are most important for achieving fairness.

The calculation of the sensitivity analysis equations for fair logistic regression problems, fixing $c = 0$, is done using the following optimization problem for a fixed $k \in [1, K]$:

$$\begin{aligned} \min_{\beta} \quad & - \sum_{\ell=1}^N [y_{\ell} \log(m_{\beta}(x^{\ell})) + (1 - y_{\ell}) \log(1 - m_{\beta}(x^{\ell}))] \\ \text{s.t.} \quad & \frac{1}{N} \sum_{\ell=1}^N (s_{\ell}^k - \bar{s}^k)(\beta^{\top} x^{\ell}) = 0 \end{aligned} \tag{19}$$

with $m_{\beta}(x^{\ell})$

$$m_{\beta}(x^{\ell}) = \frac{1}{1 + e^{-(\beta^{\top} x^{\ell})}}.$$

Calculating the Lagrangian for problem (19), we achieve:

$$\mathcal{L}(\beta, \zeta_k) = - \sum_{\ell=1}^N \left[y_{\ell} \log(m_{\beta}(x^{\ell})) + (1-y_{\ell}) \log(1-m_{\beta}(x^{\ell})) \right] + \zeta_k \left(\frac{1}{N} \sum_{\ell=1}^N (s_{\ell}^k - \bar{s}^k)(\beta^{\top} x^{\ell}) \right)$$

Now, calculating the partial derivative of the Lagrangian with respect to β as can be seen in Hilbe (2009), we have:

$$\nabla \mathcal{L}(\beta, \zeta_k) = - \sum_{\ell=1}^N \left(m_{\beta}(x^{\ell}) - y_{\ell} \right) x^{\ell} + \zeta_k \left(\sum_{\ell=1}^N (s_{\ell}^k - \bar{s}^k)(x^{\ell}) \right)$$

Using the Karush-Kuhn-Tucker (KKT) conditions, as we can see in Bertsekas et al. (2003), we find $\nabla \mathcal{L}(\beta, \zeta_k) = 0$, so:

$$- \sum_{\ell=1}^N \left(\frac{1}{1 + e^{-(\beta^{\top} x^{\ell})}} - y_{\ell} \right) x^{\ell} + \zeta_k \left(\sum_{\ell=1}^N (s_{\ell}^k - \bar{s}^k)(x^{\ell}) \right) = 0 \quad (20)$$

Since we already have the fixed effects β 's, the perturbations of all constraints can be obtained by solving the system (20). Since the solution found by solving the optimization problem is an optimal point, the KKT conditions are satisfied, so the system has a guaranteed solution.

For the Cluster-Regularized Logistic Regression, fixing $c = 0$, the sensitivity analysis is done using the following optimization problem for a fixed $k \in [1, K]$:

$$\begin{aligned} \min_{\beta, b} & - \sum_{i=1}^n \sum_{j=1}^{T_i} [y_{i_j} \log(m_{\beta, b}(x^{i_j})) + (1 - y_{i_j}) \log(1 - m_{\beta, b}(x^{i_j}))] + \lambda \sum_i b_i^2 \\ \text{s.t.} & \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{T_i} (s_{i_j}^k - \bar{s}^k)(\beta^{\top} x^{i_j} + b_i) = 0 \end{aligned} \quad (21)$$

with

$$m_{\beta, b}(x^{i_j}) = \frac{1}{1 + e^{-(\beta^{\top} x^{i_j} + b_i)}},$$

calculating the Lagrangian for problem (21), we achieve:

$$\begin{aligned} \mathcal{L}(\beta, b, \zeta_k) &= - \sum_{i=1}^n \sum_{j=1}^{T_i} [y_{i_j} \log(m_{\beta, b}(x^{i_j})) + (1 - y_{i_j}) \log(1 - m_{\beta, b}(x^{i_j}))] \\ &+ \lambda \sum_{i=1}^n b_i^2 + \zeta_k \left(\frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{T_i} (s_{i_j}^k - \bar{s}^k)(\beta^{\top} x^{i_j} + b_i) \right) \end{aligned} \quad (22)$$

Now, computing the partial derivative of the Lagrangian with respect to β , we have:

$$\begin{aligned} \nabla_{\beta} \mathcal{L}(\beta, b, \zeta_k) &= - \sum_{i=1}^n \sum_{j=1}^{T_i} \left[y_{i_j} \left(\frac{x^{i_j}}{1 + e^{(\beta^{\top} x^{i_j} + b_i)}} \right) - (1 - y_{i_j}) \left(\frac{x^{i_j}}{1 + e^{-(\beta^{\top} x^{i_j} + b_i)}} \right) \right] \\ &+ \zeta_k \left(\frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{T_i} (s_{i_j}^k - \bar{s}^k) x^{i_j} \right) \end{aligned} \quad (23)$$

and the partial derivative with respect to each b_i :

$$\begin{aligned} \nabla_{b_i} \mathcal{L}(\beta, b, \zeta_k) = & - \sum_{j=1}^{T_i} \left[y_{i_j} \left(\frac{1}{1 + e^{(\beta^\top x^{i_j} + b_i)}} \right) - (1 - y_{i_j}) \left(\frac{1}{1 + e^{-(\beta^\top x^{i_j} + b_i)}} \right) \right] \\ & + 2\lambda b_i + \zeta_k \left(\frac{1}{N} \sum_{j=1}^{T_i} (s_{i_j}^k - \bar{s}^k) \right) \end{aligned} \quad (24)$$

that is,

$$\nabla \mathcal{L}(\beta, b, \zeta_k) = \begin{bmatrix} \nabla_{\beta} \mathcal{L}(\beta, b, \zeta_k) \\ \nabla_b \mathcal{L}(\beta, b, \zeta_k) \end{bmatrix} = \begin{bmatrix} \nabla_{\beta} \mathcal{L}(\beta, b, \zeta_k) \\ \nabla_{b_1} \mathcal{L}(\beta, b_1, \zeta_k) \\ \vdots \\ \nabla_{b_i} \mathcal{L}(\beta, b_i, \zeta_k) \end{bmatrix}.$$

Using the Karush-Kuhn-Tucker (KKT) conditions, we find $\nabla \mathcal{L}(\beta, b, \zeta_k) = 0$ and since we already have the fixed effects β 's and the random effects b 's, the Lagrange multipliers of all constraints can be obtained by solving the system. Since the solution found by solving the optimization problem (6) - (8) is an optimal point, the KKT conditions are satisfied and the system has a guaranteed solution.

(J. P. Burgard, J. V. Pamplona) TRIER UNIVERSITY, DEPARTMENT OF ECONOMIC AND SOCIAL STATISTICS, UNIVERSITÄTSRING 15, 54296 TRIER, GERMANY

Email address: burgardj@uni-trier.de, pamplona@uni-trier.de

Paper 3

FairML: A Julia Package for Fair Classification

Jan Pablo Burgard, João Vitor Pamplona

Preprint under review.

URL: <https://arxiv.org/abs/2412.01585>

FAIRML: A Julia PACKAGE FOR FAIR CLASSIFICATION

JAN PABLO BURGARD, JOÃO VITOR PAMPLONA

ABSTRACT. In this paper, we propose `FairML.jl`, a `Julia` package providing a framework for fair classification in machine learning. In this framework, the fair learning process is divided into three stages. Each stage aims to reduce unfairness, such as disparate impact and disparate mistreatment, in the final prediction. For the preprocessing stage, we present a resampling method that addresses unfairness coming from data imbalances. The in-processing phase consist of a classification method. This can be either one coming from the `MLJ.jl` package, or a user defined one. For this phase, we incorporate fair ML methods that can handle unfairness to a certain degree through their optimization process. In the post-processing, we discuss the choice of the cut-off value for fair prediction. With simulations, we show the performance of the single phases and their combinations.

1. INTRODUCTION

The increase of automated decision-making necessitates the development of fair algorithms. These algorithms must adhere to societal values, particularly those that promote non-discrimination (Caton and Haas 2020). While machine learning can offer precise classifications, depending on the data situation it can also inadvertently perpetuate classification biases in crucial domains like loan approvals (Das et al. 2021) and criminal justice (Green 2018). For instance, loan approval algorithms may unfairly disadvantage single applicants by considering marital status. Similarly, criminal justice algorithms that associate race with recidivism risk can lead to discriminatory sentencing, neglecting individual circumstances. This underscores the critical need for fair classification frameworks to guarantee equal opportunities and outcomes, especially when applied within artificial intelligence application.

Driven by the growing concern of bias perpetuation by algorithms, the field of fair classification has seen a significant rise. Numerous research papers are now dedicated to exploring approaches that can mitigate bias across a wide range of algorithms. Notable examples include fair versions of logistic and linear regression (Berk et al. 2017), support vector machine (Olfat and Aswani 2017), random forests (Zhang et al. 2021), decision trees (Aghaei et al. 2019), and generalized linear models (GLMs) (Do et al. 2022). These methods are designed to promote fair and equitable outcomes for all individuals by reducing potential biases that may stem from historical data or algorithmic design choices.

Date: December 3, 2024.

Key words and phrases. Fair Machine Learning, Optimization; `Julia` language; Mixed Models.

Moreover, in machine learning, training data for automated decision-making algorithms often originates from surveys. These surveys are usually designed using a sampling plan, which can deviate from the common assumption that each data point is sampled independently and with an equal probability of inclusion. Disregarding this can introduce additional bias. To mitigate this issue, approaches that handle mixed effects were proposed. Some examples can be seen in applications like Psychology (Bono et al. 2021) and Medicine (Casals et al. 2014). For a detailed discussion on survey methods and sampling strategies, see Lohr (2009).

Packages for fair classification are already part of the literature, with versions available for Python (Jesus et al. 2024) and R (Scutari 2023). There also exists the `Fairness` package (Agrawal et al. 2020) in Julia, aiming to equalize accuracies across sensitive groups. Although these packages present several techniques, none of them consider mixed effects. Moreover, the package developed for R considers the fairness metrics statistical parity, equality of opportunity and individual fairness. Our proposal focuses more on disparate impact, disparate mistreatment, false positive rate equality and false negative rate equality. We choose these metrics because they can be adapted as constraints in the model. Besides that, our package considers fairness as constraints, solving the constrained optimization problems via solver while the Python package uses other algorithms such as boosting tree (Cruz et al. 2023) that penalizes unfairness. Additionally, our package handles mixed effects data.

The Julia programming language has been growing increasingly, especially in the field of machine learning. One reason is the availability of robust tools for optimization problems (Berman and Ginesin 2024). For this reason, a package for fair classification in Julia that takes into account an optimization problem adds value to the academic community.

This paper is organized as follows: In Section 2, we establish the theoretical underpinnings of fair classification. In Section 3, we present a novel resampling method for preprocessing data with the aim of reducing disparate impact. In Section 4, we introduce optimization problems, previously proposed in the literature, that address unfairness metrics. There we also adapt the optimization methods for data with mixed effects. In Section 5, we present cross-validation-based post-processing methods to determine an optimal cut-off value for the classification process. Finally, in Section 6, we conduct a comprehensive evaluation of our proposed package’s effectiveness through various tests. Our key findings and potential future directions are presented in Section 7.

2. MACHINE LEARNING FOR FAIR CLASSIFICATION

In machine learning, binary classification algorithms are used to estimate a specific classification $\hat{y} \in \{-1, 1\}$ for a new data point x based on a training set $\mathcal{D} = (x^\ell, y_\ell)_{\ell=1}^n$, with n being the number of points. For the point $x^\ell \in X = [x^1, \dots, x^n]$, if $y_\ell = 1$, we say that x^ℓ is in the positive class and if $y_\ell = -1$, x^ℓ

belongs to the negative class for each $\ell \in [1, n] := \{1, \dots, n\}$. Moreover, $x^\ell \in \mathbb{R}^{p+1}$, for each $\ell \in [1, n]$, due to the addition of an extra column with the value 1 as the data intercept.

When aiming for fairness in binary classification, we balance achieving good accuracy (AC) with ensuring fairness for observations ℓ based on their sensitive feature $s_\ell \in \{0, 1\}$, this is a standard approach in fair classification as stated by Zafar et al. (2017). In this work the set of sensitive features is represented by SF , being the name of the sensitive variables. While fairness in machine learning can be assessed through various metrics, in this paper we focus on disparate impact (DI) and disparate mistreatment (DM) that can be seen in Barocas and Selbst (2016) and Zafar et al. (2019), respectively. The main reason is that they are already adapted to constraints within an optimization model, as demonstrated in Zafar et al. (2017).

Considering the true labels and the predicted classifications of a supervised machine learning approach, we can categorize the data into four groups. A point is classified as true positive (TP) or true negative (TN) if its predicted class (positive or negative, respectively) matches its true label. Conversely, points are classified as false positives (FP) or false negatives (FN) if their predicted class differs from the true label. Based on this classification scheme, we can calculate accuracy, a metric where higher values indicate better classification performance. The formula for accuracy is as follows:

$$AC := \frac{TP + TN}{TP + TN + FP + FN} \in [0, 1].$$

Now, we present the fairness metrics.

Disparate Impact. Disparate impact refers to a situation where the probability under the prediction model (\mathbb{P}) is different conditional on the sensitive feature values. A classifier is considered fair with respect to disparate impact if the probability of the point being classified as positive is equal when conditioning on the sensitive feature s , i.e.,

$$\mathbb{P}(\hat{y}_\ell = 1 | s_\ell = 0) = \mathbb{P}(\hat{y}_\ell = 1 | s_\ell = 1).$$

To compute the disparate impact of a specific sensitive feature s consider:

$$\begin{aligned} \mathcal{S}_1 &= \{x^\ell : \ell \in [1, n], s_\ell = 1\}, & \mathcal{S}_0 &= \{x^\ell : \ell \in [1, n], s_\ell = 0\}, \\ \mathcal{P} &= \{x^\ell : \ell \in [1, n], y_\ell = 1\}, & \mathcal{N} &= \{x^\ell : \ell \in [1, n], y_\ell = -1\}, \\ \mathcal{D}_0^{\mathcal{P}} &= \mathcal{S}_0 \cap \mathcal{P}, & \mathcal{D}_0^{\mathcal{N}} &= \mathcal{S}_0 \cap \mathcal{N}, \\ \mathcal{D}_1^{\mathcal{P}} &= \mathcal{S}_1 \cap \mathcal{P}, & \mathcal{D}_1^{\mathcal{N}} &= \mathcal{S}_1 \cap \mathcal{N}. \end{aligned} \tag{1}$$

Let \mathcal{S}_0 and \mathcal{S}_1 be disjoint subsets of dataset X , where the sensitive feature of all points in each subset is 0 and 1, respectively. Further, let \mathcal{P} and \mathcal{N} be the subsets where the true labels of the training set \mathcal{D} are positive and negative, respectively.

Then, we have the following metric di , based on Radovanović et al. (2020):

$$\text{di} := \frac{|\{\ell : \hat{y}_\ell = 1, x_\ell \in \mathcal{S}_0\}|}{|\mathcal{S}_0|} \frac{|\mathcal{S}_1|}{|\{\ell : \hat{y}_\ell = 1, x_\ell \in \mathcal{S}_1\}|} \in [0, \infty).$$

Note that di is the ratio between the proportion of points in \mathcal{S}_0 classified as positive and the proportion of points in \mathcal{S}_1 classified as positive. Hence disparate impact, as a metric, should ideally be equal to 1 to indicate fair classifications. Values greater or lower than 1 suggest the presence of unfairness. For instance, both $\text{di} = 2$ and $\text{di} = 0.5$ represent the same amount of discrimination, but in opposite directions. To address this limitation and achieve a more nuanced metric, we use the minimum value between di and its inverse $\frac{1}{\text{di}}$. Furthermore, to align with the convention of other fairness metrics where a value closer to 0 indicates greater fairness (as will be show later), we redefine the DI as follows

$$\text{DI} := 1 - \min(\text{di}, \text{di}^{-1}) \in [0, 1]. \quad (2)$$

Hence, a value closer to 0 indicates better performance and a value closer to 1 indicates worse performance.

Disparate Mistreatment. Disparate mistreatment, also known as equalized odds (Hardt et al. 2016), is defined as the condition in which the misclassification rates for points with different values in the sensitive features are unequal. In other words, a classification is free of disparate mistreatment when the classification algorithm is equally likely to misclassify points in both positive and negative classes, regardless of their sensitive characteristics.

A classification is considered free of disparate mistreatment if the rate of false positives and false negatives is equal for both categories of a sensitive feature s . That is,

$$\mathbb{P}(\hat{y}_\ell = 1 | \ell \in \mathcal{D}_0^{\mathcal{N}}) = \mathbb{P}(\hat{y}_\ell = 1 | \ell \in \mathcal{D}_1^{\mathcal{N}})$$

and

$$\mathbb{P}(\hat{y}_\ell = -1 | \ell \in \mathcal{D}_0^{\mathcal{P}}) = \mathbb{P}(\hat{y}_\ell = -1 | \ell \in \mathcal{D}_1^{\mathcal{P}}).$$

To quantify the disparate mistreatment with respect to a specific sensitive feature s , we first establish the equations for the false positive rate (FPR) and false negative rate (FNR) metrics. The FPR metric is defined as the absolute value of the difference between the false positive rates of the categories defined by the sensitive feature s , as follows:

$$\begin{aligned} FPR &:= |FPR_{s=0} - FPR_{s=1}| \\ &= \left| \frac{FP_{s=0}}{FP_{s=0} + TN_{s=0}} - \frac{FP_{s=1}}{FP_{s=1} + TN_{s=1}} \right| \quad (\text{FPR}) \\ &= \left| \frac{|\{\ell : \hat{y}_\ell = 1, x_\ell \in \mathcal{D}_0^{\mathcal{N}}\}|}{|\mathcal{D}_0^{\mathcal{N}}|} - \frac{|\{\ell : \hat{y}_\ell = 1, x_\ell \in \mathcal{D}_1^{\mathcal{N}}\}|}{|\mathcal{D}_1^{\mathcal{N}}|} \right| \in [0, 1]. \end{aligned}$$

Similarly, the FNR is given by:

$$\begin{aligned}
 FNR &:= |FNR_{s=0} - FNR_{s=1}| \\
 &= \left| \frac{FN_{s=0}}{FN_{s=0} + TP_{s=0}} - \frac{FN_{s=1}}{FN_{s=1} + TP_{s=1}} \right| \tag{FNR} \\
 &= \left| \frac{|\{\ell : \hat{y}_\ell = -1, x^\ell \in \mathcal{D}_0^P\}|}{|\mathcal{D}_0^P|} - \frac{|\{\ell : \hat{y}_\ell = -1, x^\ell \in \mathcal{D}_1^P\}|}{|\mathcal{D}_1^P|} \right| \in [0, 1],
 \end{aligned}$$

Disparate mistreatment is the mean of both metrics above. Again, the lower the value, the fairer classification.

$$DM = \frac{FPR + FNR}{2} \in [0, 1]. \tag{DM}$$

With our fairness metrics at hand, we now present the strategy of our Julia package, FairML that employs a variety of optimization techniques and a resampling strategy to ensure fairness in classifications based on a user-specified sensitive attribute. The package operates under a three-step framework:

- (1) Preprocessing: This stage encompasses the implementation of functions that perform initial data manipulation aimed at enhancing fairness metrics;
- (2) In-processing: This stage constitutes the main part of the paper, where optimization problems are addressed with the aim of improving a specific fairness metric;
- (3) Post-processing: Following the previous stage, which outputs class membership probabilities, this phase is responsible for performing classification. It may or may not employ strategies to optimize a specific fairness metric in relation to accuracy.

While the theoretical underpinnings, construction, and explanation of each stage will be detailed in subsequent chapters, the package’s core functionality unifies all stages into a single, user-friendly interface:

Julia Code 1 Classification function.

```

1 classification = fair_pred(xtrain::DataFrame, ytrain::Vector{Union{Float64,
  ↪ Int64}}, newdata::DataFrame, inprocess::Function, SF::Array{String},
  ↪ preprocess::Function=id_pre, postprocess::Function=id_post, c::Real=0.1,
  ↪ R::Int64=1, seed::Int64=42, SFpre::String=SF, SFpost::String=SF)

```

Besides that, many datasets exhibit unexplained variation within groups or across different levels, more details can be seen in Section 4. Hence, in this package we also propose a classification function for this type of data:

Julia Code 2 Classification function for mixed models.

```

1 classifications = me_fair_pred(xtrain::DataFrame, ytrain::Vector,
  ↪ newdata::DataFrame, group_id_train::CategoricalVector,
  ↪ group_id_test::CategoricalVector, inprocess::Function,
  ↪ SF::Array{String}, postprocess::Function=id_post, c::Real=0.1,
  ↪ SFpost::String=SF)

```

Being:

- Input arguments:
 - (1) *xtrain*: The dataset that the labels are known (training set);
 - (2) *ytrain*: The labels of the dataset *xtrain*;
 - (3) *newdata*: The new dataset for which we want to obtain the *classifications*;
 - (4) *inprocess*: One of the several optimization problems available in this package or any machine learning classification method present in MLJ.jl package;
 - (5) *SF*: One or a set of sensitive features (variables names. E.g Sex, race...), that will act in the in-processing phase. If the algorithm come from the MLJ.jl package, no fair constraint are acting in this phase;
 - (6) *group_id_train*: Training set group category;
 - (7) *group_id_newdata*: New dataset group category.
- Optional argument:
 - (1) *preprocess*: A pre-processing function among the options available in this package, *id_pre()* by default;
 - (2) *postprocess*: A post-processing function among the options available in this package, *id_post()* by default;
 - (3) *c*: The threshold of the fair optimization problems, 0.1 by default;
 - (4) *R*: Number of iterations of the preprocessing phase, each time sampling differently using the resampling method, 1 by default;
 - (5) *seed*: For sample selection in *R*, 42 by default;
 - (6) *SFpre*: One sensitive features (variable name), that will act in the preprocessing phase, disabled by default;
 - (7) *SFpost*: One sensitive features (variable name), that will act in the post-processing phase, disabled by default.
- Output arguments:
 - (1) *classifications*: Classifications of the *newdata* points.

The classification function for mixed models ignores the preprocessing phase, as this phase tends to eliminate numerous data points, as discussed in 3. Such elimination can lead to empty groups, which is not permissible in the classification functions for mixed models.

It is essential to highlight that both the preprocessing and post-processing stages should be limited to handling a single sensitive feature each. Only the in-processing stage can handle with multiple sensitive features at the same time, creating multiples fairness constraints for the optimization problems. However, sensitive features can differ across the three phases with the aim to achieve fairness through various potential discrimination classes.

3. PREPROCESSING

Resampling methods can serve various purposes, as can be seen in Good (2013). In our case, the goal is to mitigate disparate impact or disparate mistreatment in the data. We achieve this by generating multiple datasets that exhibit less unfairness than the original. In this context, we developed a hybrid approach that combines an adapted undersampling technique with cross-validation to address this issue.

Undersampling (Mohammed et al. 2020) reduces the majority class, in the sensitive feature, to balance the dataset, while cross-validation (Blagus and Lusa 2015) provides a evaluation of the model by iteratively training and testing on different subsets. Similar approaches have been used for class-imbalanced data in Zughrat et al. (2014) and Jesus et al. (2024).

As indicated by Equation (2), regarding to disparate impact, our goal is to ensure equal representation of positive and negative labels across both categories of the sensitive features. To achieve this, we enforce this condition within the training set \mathcal{D} using the following strategy:

- (1) Separate the training data \mathcal{D} as in Equation (1);
- (2) Compute the size of the smallest among the four subsets:

$$J = \min(|\mathcal{D}_0^{\mathcal{N}}|, |\mathcal{D}_1^{\mathcal{N}}|, |\mathcal{D}_0^{\mathcal{P}}|, |\mathcal{D}_1^{\mathcal{P}}|).$$

- (3) For each subset do a random sampling with replacement, M as follows:

$$\begin{aligned} M_J^{\mathcal{D}_0^{\mathcal{N}}} &\subseteq \mathcal{D}_0^{\mathcal{N}}, \text{ with } |M_J^{\mathcal{D}_0^{\mathcal{N}}}| = J, & M_J^{\mathcal{D}_0^{\mathcal{P}}} &\subseteq \mathcal{D}_0^{\mathcal{P}}, \text{ with } |M_J^{\mathcal{D}_0^{\mathcal{P}}}| = J, \\ M_J^{\mathcal{D}_1^{\mathcal{N}}} &\subseteq \mathcal{D}_1^{\mathcal{N}}, \text{ with } |M_J^{\mathcal{D}_1^{\mathcal{N}}}| = J, & M_J^{\mathcal{D}_1^{\mathcal{P}}} &\subseteq \mathcal{D}_1^{\mathcal{P}}, \text{ with } |M_J^{\mathcal{D}_1^{\mathcal{P}}}| = J. \end{aligned}$$

- (4) Create the new training dataset:

$$\mathcal{D} = M_J^{\mathcal{D}_0^{\mathcal{N}}} \cup M_J^{\mathcal{D}_1^{\mathcal{N}}} \cup M_J^{\mathcal{D}_0^{\mathcal{P}}} \cup M_J^{\mathcal{D}_1^{\mathcal{P}}}.$$

Therefore, since there is no disproportionality of labels across different sensitive features categories, we expected to have a new dataset with less disparate impact than the previous one.

Observe that the generation of the new dataset is a random process. To account for the insecurity introduced by the random generation, we allow the user to define the number R of times this data set is to be generated. In the pre-processing phase, the best one is chosen as follows:

- (1) Do the preprocessing phase R times, generating R different datasets;

- (2) For each dataset:
 - (a) Calculate the coefficients using the in-processing phase;
 - (b) Compute the classifications on the full training set (before resampling);
 - (c) Use the classifications to calculate disparate impact or disparate mistreatment;
- (3) Select the classification with the best fairness metric value;
- (4) Use the coefficients from the best classification to calculate classifications on new data.

That is, from all the R calculated coefficients, this phase selects the one that generate the smallest disparate impact or disparate mistreatment on the full training set, and uses it to classify the points in the new dataset (input *newdata*).

While the algorithm was designed to address disparate impact, preliminary numerical tests have shown that it can also mitigate disparate treatment. This makes it a flexible tool, allowing the user to choose the specific focus.

The inputs and outputs of the preprocessing function (`di_pre`) are documented on the [package's GitHub page](#).

In the next section, we will explain the in-processing phase.

4. IN-PROCESSING

The main goal of the in-processing phase is to predict the probability of a new point being classified as 1 or -1 . This is achieved by finding the coefficients of a prediction model by solving an optimization problem. We propose several optimization problems that can improve the fairness metrics of disparate impact, false positive rate, false negative rate, and disparate mistreatment.

This paper mainly focuses on two methods for binary classification. The first approach is logistic regression (LR). Since in our data we have $y \in \{-1, 1\}$, we adapt, w.l.o.g., the logistic regression model (Neter et al. 2004).

$$\min_{\beta} - \sum_{\ell=1}^n \left[\left(\frac{y_{\ell} + 1}{2} \right) \log(m_{\beta}^{LR}(x^{\ell})) + \left(\frac{y_{\ell} - 1}{2} \right) \log(1 - m_{\beta}^{LR}(x^{\ell})) \right] \quad (\text{LR})$$

with the prediction function given by

$$m_{\beta}^{LR}(x) := \frac{1}{1 + e^{-\beta^{\top} x}}. \quad (3)$$

The second method is the standard Support Vector Machine (SVM), proposed by Vapnik and Chervonenkis (1964) and Hearst et al. (1998).

$$\begin{aligned} \min_{(\beta, \xi)} \quad & \frac{1}{2} \|\beta\|^2 + \mu \sum_{\ell=1}^n \xi_{\ell} \\ \text{s.t.} \quad & y_{\ell}(m_{\beta}^{SVM}(x_{\ell})) \geq 1 - \xi_{\ell}, \ell = 1, \dots, n \end{aligned} \quad (\text{SVM})$$

with the prediction function given by

$$m_{\beta}^{SVM}(x) := \beta^{\top} x. \quad (4)$$

As already mentioned, the first column of the matrix X should be a vector of ones, that is, the first entrance of $x^\ell, \forall \ell \in [1, n]$, is equal to 1. If this column does not exist, the functions of this package automatically add one. Note that in standard SVM implementations, an intercept term is typically not added to the data, but rather a so called bias is included in the problem constraints. Using the formulation of Hsieh et al. (2008), we can adjust it to include an intercept term being the first entry in β .

In problems (SVM) and (LR), fairness constraints can be added. Let us now present them, based on the formulations of Zafar et al. (2017).

Fairness Constraints for Disparate Impact. As stated in Expression (2), to ensure a classification is free from disparate impact, the conditional probabilities of a positive classification given the sensitive feature s should be equal. While achieving zero disparate impact is a desirable goal, it can potentially reduce the classification accuracy, as we have a trade-off between fairness and accuracy (Menon and Williamson 2018; Zhao and Gordon 2022). To address this trade-off, Zafar et al. (2017) suggest introduce a fairness threshold, denoted by $c \in \mathbb{R}^+$, which allows us to adjust the relative importance placed on fairness compared to accuracy. With this logic, we deduce the following constraints:

$$\begin{aligned} \frac{1}{n} \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) &\leq c \\ \frac{1}{n} \sum_{\ell=1}^n (s_\ell - \bar{s})(\beta^\top x^\ell) &\geq -c. \end{aligned} \tag{5}$$

A more detailed description of how disparate impact constraints are constructed is provided in Burgard and Pamplona (2024b). Note that these constraints take into account the inner product $\beta^\top x^\ell$, which is the main component in both prediction functions (4) and (3).

Fairness Constraints for Disparate Mistreatment. As previously discussed, in Section 2, the fairness metric disparate mistreatment aims to simultaneously equalize or approximate (depending on c) the false negative rate and false positive rate across the different categories of the sensitive feature.

We begin by considering the *FNR* constraint. A point is a false negative if $y_\ell = 1$ and $\beta^\top x^\ell < 0$, that is, if and only if

$$\min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) \tag{6}$$

is greater than zero, being β the coefficient. In fact, let us examine all four possibilities:

- (1) True Negative: $y_\ell = -1$ and $\beta^\top x^\ell < 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = 0$
- (2) False Positive: $y_\ell = -1$ and $\beta^\top x^\ell > 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = 0$
- (3) False Negative: $y_\ell = 1$ and $\beta^\top x^\ell < 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = \beta^\top x^\ell$
- (4) True Positive: $y_\ell = 1$ and $\beta^\top x^\ell > 0 \implies \min\left(0, \frac{1+y_\ell}{2} y_\ell \beta^\top x^\ell\right) = 0$

For this reason, Zafar et al. (2016) uses the Expression (6) to select the false negative points among all points. However, note that in the *FNR* constraint, we only need to care about the points that belong to \mathcal{P} , because for the point that belongs to \mathcal{N} the Expression (6) is always equal to 0. Since for a point $x^\ell \in \mathcal{P}$ we have $y_\ell = 1$, Expression (6) becomes $\min(0, \beta^\top x^\ell)$.

To obtain the same proportion of false negatives in both sensitive categories, the *FNR* constraints impose that the sums of the minimum between 0 and the inner products of the coefficient and a positive point are close to each other in each sensitive category, as follows:

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^\top x^\ell) \leq c \quad (7a)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{P}}} \min(0, \beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{P}}} \min(0, \beta^\top x^\ell) \geq -c \quad (7b)$$

For false positive points, we employ the same logic, however, replacing the Expression (6) with:

$$\min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell),$$

and hence

- (1) True Negative: $y_\ell = -1$ and $\beta^\top x^\ell < 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = 0$
- (2) False Positive: $y_\ell = -1$ and $\beta^\top x^\ell > 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = -\beta^\top x^\ell$
- (3) False Negative: $y_\ell = 1$ and $\beta^\top x^\ell < 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = 0$
- (4) True Positive: $y_\ell = 1$ and $\beta^\top x^\ell > 0 \implies \min(0, \frac{1-y_\ell}{2} y_\ell \beta^\top x^\ell) = 0$

That is, in the *FPR* constraints, we only need to care about the points that belong to \mathcal{N} . Similarly to the *FNR* constraints, the *FPR* constraints impose that the sums of the minimum between 0 and minus the inner products of the coefficient and a negative point are close to each other in each sensitive category. That is,

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) \leq c \quad (8a)$$

$$\frac{|\mathcal{S}_0|}{n} \sum_{x^\ell \in \mathcal{D}_1^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) - \frac{|\mathcal{S}_1|}{n} \sum_{x^\ell \in \mathcal{D}_0^{\mathcal{N}}} \min(0, -\beta^\top x^\ell) \geq -c \quad (8b)$$

Therefore, the Disparate Mistreatment constraints are a combination of Constraints (7a),(7b),(8a) and (8b).

Given the constraints we have presented, we can utilize the following problems in the in-processing phase:

- Logistic regression free of disparate impact;
- Logistic regression free of false negative rate;
- Logistic regression free of false positive rate;
- Logistic regression free of disparate mistreatment;
- Support vector machine free of disparate impact;

- Support vector machine free of false negative rate;
- Support vector machine free of false positive rate;
- Support vector machine free of disparate mistreatment.

Problems (LR), (SVM) and above do not deal with random effects, which can be happening in diverse application, like from medicine or psychology (Bono et al. 2021; Casals et al. 2014). However, these problems, like many other statistical models, can lead to unfair outcomes. In light of this, we propose a novel research area designated as fair machine classification for data with mixed effects (Burgard and Pamplona 2024a,b). We adapt well-established methods from the literature to address fair machine learning optimization problems in the presence of random effects.

Mixed Model. To capture the latent heterogeneity present in some types of data, which can encompasses cultural, demographic, biological, and behavioral aspects, it is imperative to incorporate random effects into the predictive model. Omitting these effects can lead to substantial bias in the classifications, compromising the accuracy and generalization of the results (Barili et al. 2018; Yang et al. 2014).

Let g being the random vector and g_i with $i \in [1, K]$, representing the group-specific random effect, with g following a normal distribution with mean zero. Consider Γ_i the size of the group i for each $i \in [1, K]$ and y_{ij} the label of $(x^{ij})^\top = (x_1^{ij}, \dots, x_p^{ij})$ with $j \in [1, \Gamma_i]$.

To ensure that in all of our problems we have $y \in \{-1, 1\}$, we adapt, w.l.o.g., the mixed effects logistic regression model as we did in (LR).

$$\min_{\beta, g} - \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \left[\left(\frac{y_{ij} + 1}{2} \right) \log(m_{\beta, g}^{LR}(x^{ij})) + \left(\frac{y_{ij} - 1}{2} \right) \log(1 - m_{\beta, g}^{LR}(x^{ij})) \right] + \lambda \sum_{i=1}^K g_i^2 \quad (\text{MELR})$$

with the prediction function given by

$$m_{\beta, g}^{LR}(x^{ij}) := \frac{1}{1 + e^{-(\beta^\top x^{ij} + g_i)}}, \quad (9)$$

and y_{ij} being the label in the observation j in group i and $j \in [1, \Gamma_i]$, and Γ_i the size of the group i . For a detailed explanation and a breakdown of the Mixed Effects Logistic Regression derivation, see Burgard and Pamplona (2024a). For the Mixed Effects Support Vector Machine, we consider the model present by Burgard and Pamplona (2024b):

$$\begin{aligned} \min_{(\beta, g, \xi)} \quad & \frac{1}{2} \|\beta\|^2 + \mu \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} \xi_{ij} + \lambda \sum_{i=1}^K g_i^2 \\ \text{s.t.} \quad & y_{ij}(m_{\beta, g}^{SVM}(x^{ij})) \geq 1 - \xi_{ij}, \quad i = [1, K], \quad j = [1, \Gamma_i] \end{aligned} \quad (\text{MESVM})$$

with the prediction function given by

$$m_{\beta, g}^{SVM}(x^{ij}) := \beta^\top x_{ij} + g_i. \quad (10)$$

In mixed models, all constraints previously constructed for regular models are adapted to account for the existence of the random effect. The construction logic

for these constraints is equivalent to the problems with only fixed effects, with an adaptation of the created subgroups as shown in (1) as follows:

$$\begin{aligned} \mathcal{S}_1^i &= \{x^{ij} : j \in [1, \Gamma_i], s_{ij} = 1\}, & \mathcal{S}_0^i &= \{x^{ij} : j \in [1, \Gamma_i], s_{ij} = 0\}, \\ \mathcal{P}^i &= \{x^{ij} : j \in [1, \Gamma_i], y_{ij} = 1\} & \mathcal{N}^i &= \{x^{ij} : j \in [1, \Gamma_i], y_{ij} = -1\}, \\ \mathcal{D}_0^{\mathcal{P}^i} &= \mathcal{S}_0^i \cap \mathcal{P}^i, & \mathcal{D}_0^{\mathcal{N}^i} &= \mathcal{S}_0^i \cap \mathcal{N}^i, \\ \mathcal{D}_1^{\mathcal{P}^i} &= \mathcal{S}_1^i \cap \mathcal{P}^i, & \mathcal{D}_1^{\mathcal{N}^i} &= \mathcal{S}_1^i \cap \mathcal{N}^i. \end{aligned}$$

Observe that each subset is created for each cluster $i \in [1, K]$.

Moreover, we need to modify the fairness constraints to account for random effects.

Disparate Impact. Following the same logic as presented before, but considering a group-to-group analysis, we have a similar construction for the disparate impact constraints in mixed models that can be seen in Burgard and Pamplona (2024b) and is given by:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) &\leq c, \\ \frac{1}{n} \sum_{i=1}^K \sum_{j=1}^{\Gamma_i} (s_{ij} - \bar{s})(\beta^\top x^{ij} + g_i) &\geq -c. \end{aligned}$$

Disparate Mistreatment. We now discuss the DM metric for mixed effects. For the *FNR* constraints, we adapt the Expression (6) to incorporate the random effects as follows:

$$\min \left(0, \frac{1 + y_{ij}}{2} y_{ij} (\beta^\top x^{ij} + g_i) \right).$$

As done for the regular models, we only need take care about the positive points. And, for these points, the expression above becomes $\min(0, \beta^\top x^{ij} + g_i)$.

On the other hand, for the *FPR* constraints, the selection of the false positive points is adapted to

$$\min \left(0, \frac{1 - y_{ij}}{2} y_{ij} (\beta^\top x^{ij} + g_i) \right).$$

Here we only need to take care about the negative points. And, for these points, the expression above becomes $\min(0, -\beta^\top x^{ij} - g_i)$. Combining all constraints yields the following set of constraints for a classification free of disparate mistreatment in mixed models:

$$\begin{aligned}
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) \leq c \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{P}^i}} \min(0, \beta^\top x^{ij} + g_i) \geq -c \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) \leq c \\
& \frac{|\mathcal{S}_0|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_1^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) - \frac{|\mathcal{S}_1|}{n} \sum_{i=1}^K \sum_{x^{ij} \in \mathcal{D}_0^{\mathcal{N}^i}} \min(0, -\beta^\top x^{ij} - g_i) \geq -c.
\end{aligned}$$

The first summation iterates over all groups, while the second summation iterates only over the desired points within each group.

Similarly to regular models, we can assign the constraints above to problems ([MELR](#)) and ([MESVM](#)), leading to 8 new additional optimization problems, which are:

- Mixed effects logistic regression free of disparate impact
- Mixed effects logistic regression free of false negative rate
- Mixed effects logistic regression free of false positive rate
- Mixed effects logistic regression free of disparate mistreatment
- Mixed effects support vector machine free of disparate impact
- Mixed effects support vector machine free of false negative rate
- Mixed effects support vector machine free of false positive rate
- Mixed effects support vector machine free of disparate mistreatment

Unlike regular models, mixed model algorithms cannot be replaced by MLJ models, as the latter are not suitable for this kind of problem.

It is worth to remember that all constraints, both for the regular model and the model that includes random effects, allow for the use of multiple sensitive features simultaneously.

The inputs and outputs of all in-processing functions are documented on the [package's GitHub page](#) and in the next section, we will explain the post-processing phase.

5. POST-PROCESSING

The post-processing phase implements an algorithm that seeks an optimal cut-off value for classification (Cheong et al. 2013; Ren et al. 2016). An approach that implements a similar strategy, but considering each sensitive group, can be seen in Jesus et al. (2024). In our approach, we consider the entire dataset to ensure that no particular sensitive group is at advantaged or disadvantaged.

Classifications are computed using the predicted probability values from both the training and testing sets obtained from the previous phase.

Given the predicted probabilities from both the training and new datasets obtained in the in-processing phase:

- (1) For each cut-off value v ranging from 0.01 to 0.99 (with an increment of 0.01), do:
 - Generate classifications for training set as follows: if the probability is greater or equal v , classify as positive, otherwise as negative;
 - Compute the accuracy (AC_v) and the desired fairness metric value (fm_v) for training set.
- (2) Select only the values of v that decrease at most 5% of the accuracy compared to the accuracy given by the cut-off value $v = 0.5$. Among them, select the best result using $B = \operatorname{argmax}_v(AC_v - fm_v)$;
- (3) Use the new cut-off value, B , for the test set (*newdata*) classification.

If the user does not wish to use this phase in the classification process, the cut-off value v will be 0.5 by default. The value of 5% was determined through preliminary tests which demonstrated that allowing a greater reduction in accuracy could misclassify a significant number of data points into a specific class.

This strategy can be employed with any fairness metric documented within the package.

It is crucial to remember that the post-processing phase only affects a single sensitive feature. Therefore, if multiple sensitive features are utilized during the in-processing phase, just one can be selected in the post-processing phase.

The post-processing phase can be used in regular and mixed effects algorithms. In the following section, we demonstrate the effectiveness of the proposed package using multiple numerical simulations. The inputs and outputs of all post-processing functions are documented on the [package's GitHub page](#).

6. NUMERICAL RESULTS

Here, we present several numerical results to validate the proposed method's efficacy. First, we present the step-by-step strategy used to create the synthetic datasets and to conduct the numerical experiments. The tests are run in `Julia 1.9` (Bezanson et al. 2017) with the packages `JuMP` (Lubin et al. 2023), `Ipopt` (Wächter and Biegler 2006), to solve the optimization problems, `Distributions` (Bezanson et al. 2021) and `DataFrames` (Bouchet-Valat and Kamiński 2023).

To create the synthetic data, we define the following parameters:

- *Number of points*: Number of points in the dataset;
- β 's: The fixed effects;
- g 's: The random effects with distribution $N(0, 3)$, if necessary;
- *Data points*: The covariate vector associated with fixed effects with distribution $N(0, 1)$;
- c : Threshold from fair constraints;
- *seed*: Random seed used in the generation of data;

- *Train-Test split*: Approximately 1% of the dataset was used for the training set, and 99% for the test set.

The classifications of the synthetic dataset, are computed using the predictions functions (3), (4), (9) and (10), depending on the problem being solved. The package also provides these synthetic dataset generation functions.

The tests were conducted on a laptop with an Intel Core i9-13900HX processor with a clock speed of 5.40 GHz, 64 GB of RAM, and Windows 11 operating system, with 64-bit architecture.

All figures were created using the `Plots` and `PlotlyJS` packages, developed by Christ et al. (2023) and all unspecified hyperparameters were obtained through cross-validation (Browne 2000).

6.1. Regular Models. The parameters for creating synthetic datasets are as follows:

- β 's = [-2.0; 0.4; 0.8; 0.5; 2.0]
- $c = 0.1$.

The β_0 is the intercept, and β_4 is the coefficient associated to the binary sensitive feature. In the unfair case, the coefficient was randomly selected using numbers between 0 and 1, except for β_0 and β_4 . The reason for this is that we assign a high value to β_4 , to give more importance to the sensitive variable in the label. In other words, data points with the sensitive categories equal to 1 are more likely to be classified as positive. This practice results in a dataset that is inherently unfair in terms of both disparate impact and disparate mistreatment, as needed to test our methods. For all experiments, the matrix X was randomly generated from a multivariate normal distribution with zero mean and independent variables. Using the generated coefficients, we employed Prediction function (3) to obtain labels for logistic regression tests and the prediction function in (4) for SVM tests.

In the numerical tests for regular models we consider these options of methods, all documented in Sections 3, 4 and 5:

- Three options of preprocessing methods: Identity, disparate impact with $R = 1$ and disparate impact with $R = 5$;
- Ten options of in-processing methods, logistic regression and SVM based ones;
- Three post-processing methods: Disparate Impact, Disparate Mistreatment and no post-processing.

This leads to a total of 90 scenarios with 100 simulation runs each. For each optimization problem we impose a time limit of 60 seconds in the in-processing stage. Only the most relevant results are shown here, the other ones can be found on [GitHub](#). For each numerical test, box plots were generated for 7 metrics. We now present the most noteworthy numerical results. Firstly, we will demonstrate the effectiveness of the preprocessing method proposed in this work.

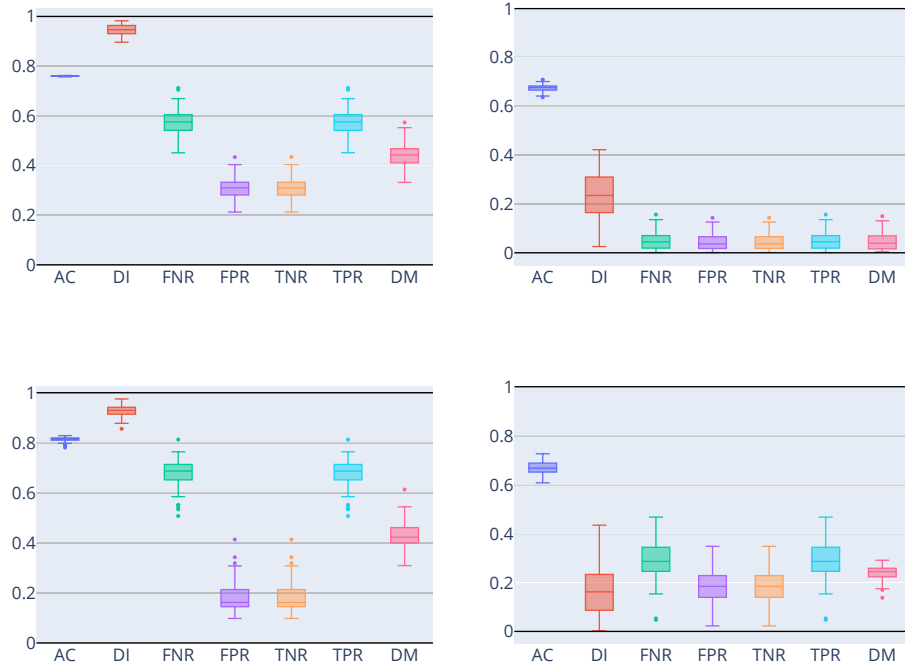


Figure 1. Preprocessing results: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Without preprocessing. Right: With preprocessing ($R=1$)

As can be seen in Figure 1 for both logistic regression and SVM, the proposed resampling method, significantly reduces the disparate impact. It is also worth noting that this leads to a decline in other fairness metrics as well. This implies in a decrease of accuracy, however, this is an anticipated outcome in the field of fair machine learning. Now, considering the same preprocessing but being executed multiple times:

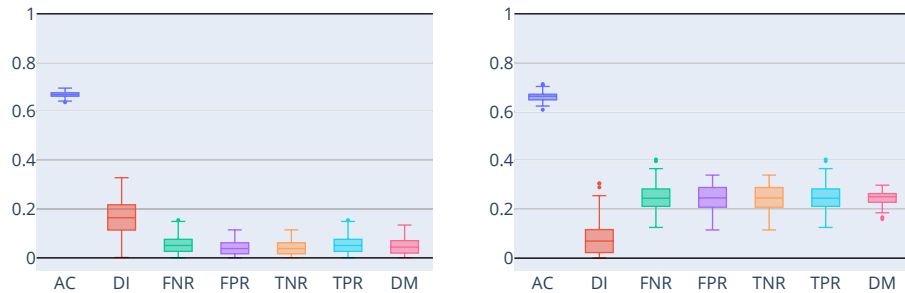


Figure 2. Preprocessing with multiple runs ($R=5$): Left: Logistic regression. Right: Support vector machine.

It can be observed from Figure 2 that repeating the resampling method and selecting the best solution is also an effective approach, in comparison to the right

side of the Figure 1, which is executed only once. Therefore, it is recommended when time is not an issue.

Henceforth, the following numerical tests focus on optimization problems during the in-processing phase.

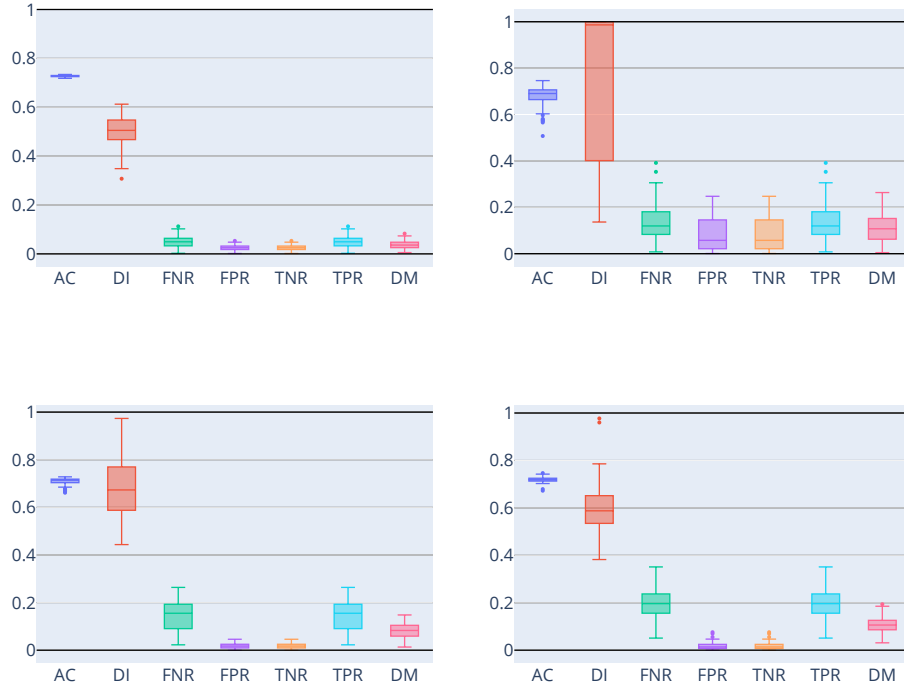


Figure 3. In-processing results: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Disparate impact. Right: Disparate mistreatment.

In this set of tests, we can verify that, when compared to tests without fairness constraints, in Figures 1, the fair optimization problems effectively reduced the fairness metrics they are designed to mitigate. I.e., when using the optimization problems with disparate impact constraints we have a decrease of DI. We can see similar results for disparate mistreatment.

Finally, we demonstrate the effectiveness of the post-processing phase, also proposed in this paper.

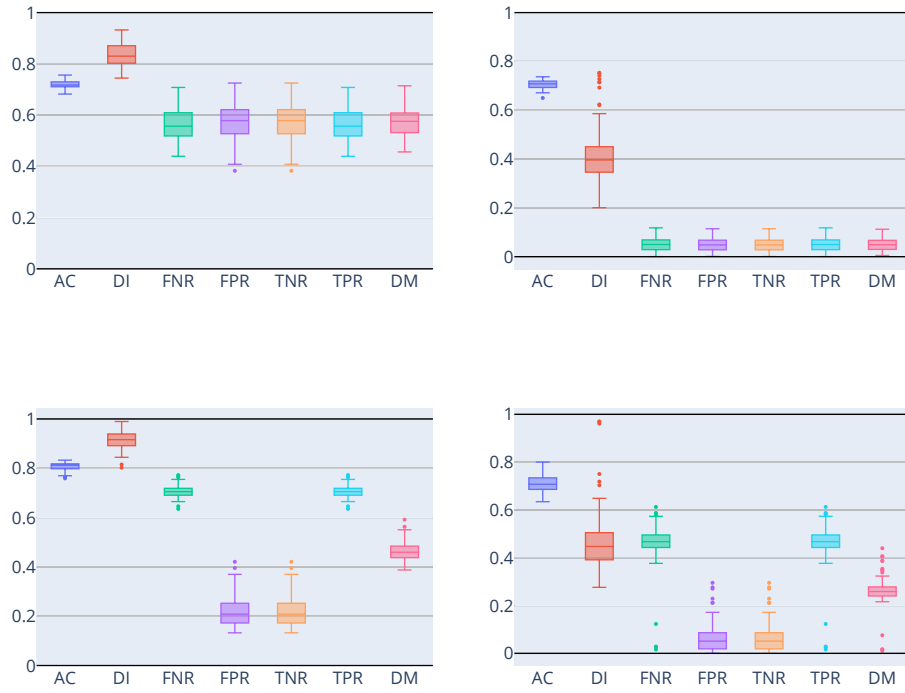
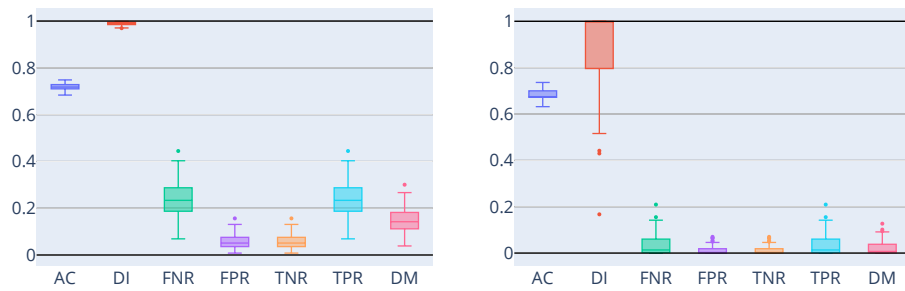


Figure 4. Post-processing results for disparate impact: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.



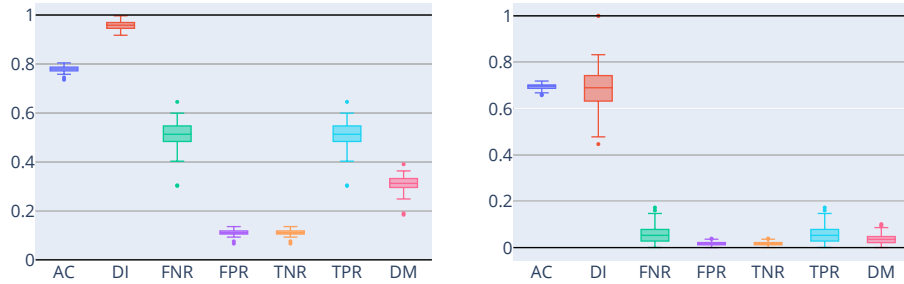


Figure 5. Post-processing results for disparate mistreatment: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.

The post-processing phase can be utilized independently, without the in-processing phase affecting the fair metrics, or both phases can be employed simultaneously. It can be observed, in the left side of Figures 4 and 5, that employing solely the post-processing phase leads to a slight improvement in the desired fairness metric without significantly compromising accuracy. Furthermore, it can be noted that utilizing both strategies in conjunction, as can be seen in the right side of Figures 4 and 5, yields superior outcomes compared to employing either in-processing or post-processing alone. Consequently, our recommendation is to utilize both strategies simultaneously.

6.2. Mixed Models. The parameters for creating synthetic datasets are as follows:

- β 's = $[-4.0; 0.4; 0.8; 0.5; 4.0]$;
- g 's: 100 groups with $b_i \sim N(0, 3.0)$, with $i \in [1, 100]$;
- $c = 0.1$.

In the numerical tests for mixed models we consider the following options of methods, documented in Sections 4 and 5:

- Ten options of in-processing methods, logistic regression and SVM based ones;
- Three post-processing methods: Disparate Impact, Disparate Mistreatment and no post-processing.

Hence, we have 30 scenarios, with 100 simulation runs each. For each optimization problem we impose a time limit of 60 seconds in the in-processing stage. As for regular models, we only present the most important results, the rest can be found on [GitHub](#). Since the mixed models strategy does not include a preprocessing phase, we will first examine the in-processing phase, where the optimization problems proposed in this work are solved.

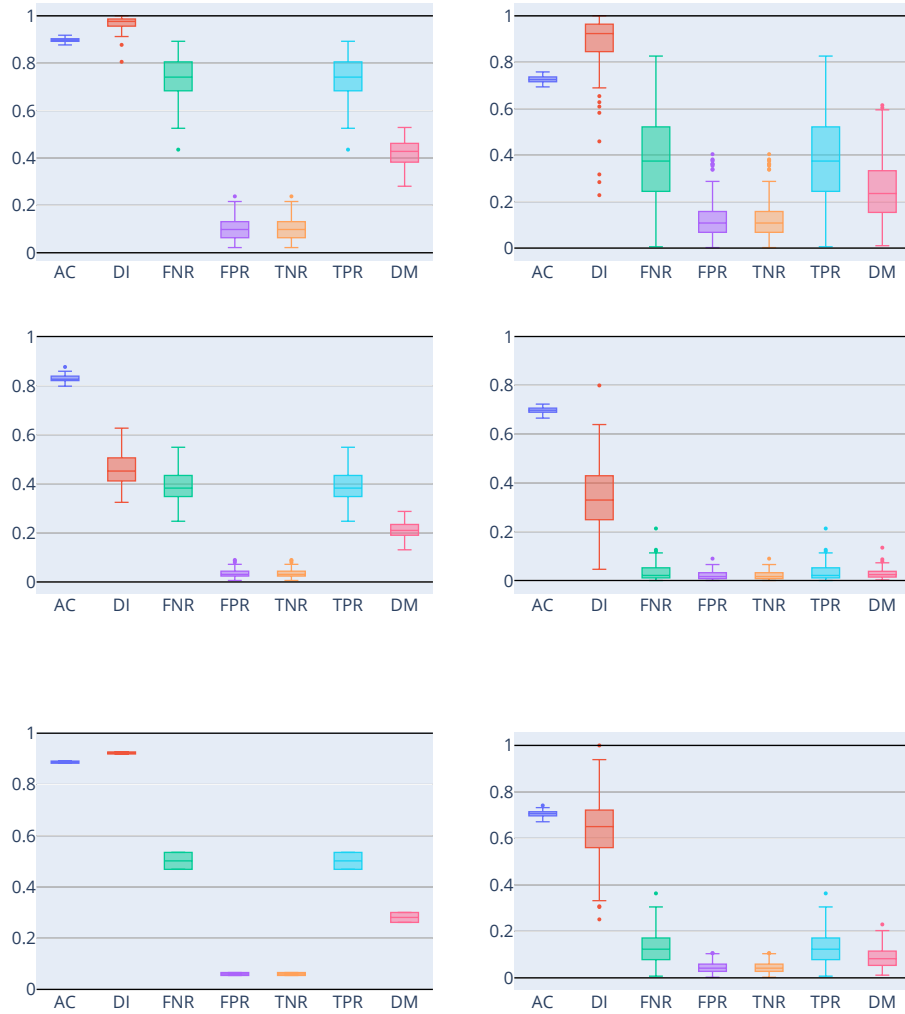


Figure 6. In-processing results in Mixed models: Left: Comparison for logistic regression. Right: Comparison for support vector machine. First row: No fairness constraints. Second row: Disparate impact constraints. Third row: Disparate mistreatment constraints.

Figure 6 confirms that fairness constraints in the optimization problems successfully improve the fairness metrics they were designed to address. That is, when incorporating disparate impact constraints into optimization problems, we observe a reduction in disparate impact. Similar results are evident for disparate mistreatment.

At last, we show the effectiveness of the post-processing step on the mixed models.

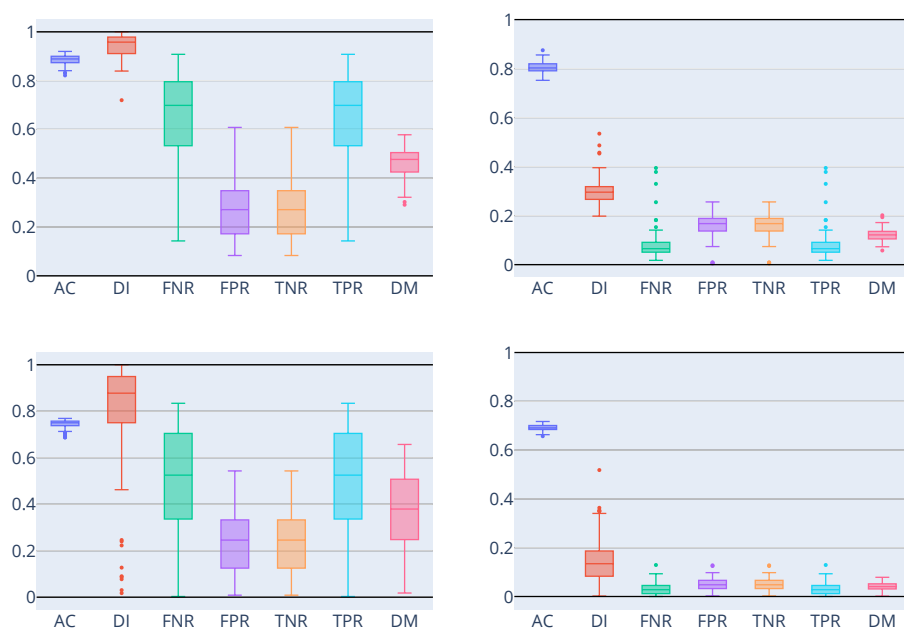
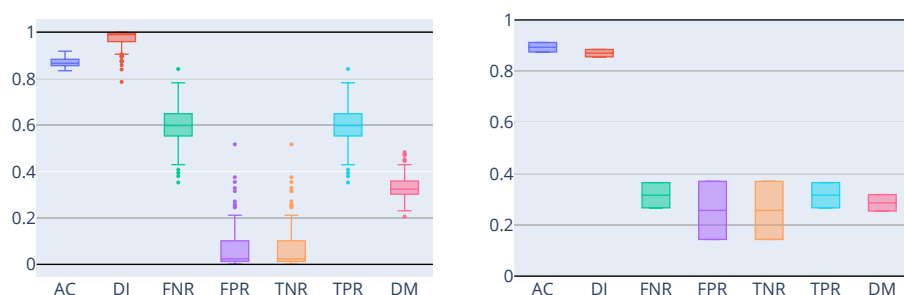


Figure 7. Post-processing results in mixed models for disparate impact: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.

Just as in post-processing tests for regular models, while the post-processing phase can function independently, its integration with the in-processing phase yields superior results. Consequently, we reiterate our recommendation to employ both phases simultaneously as can be seen in Figures 7 and 8.



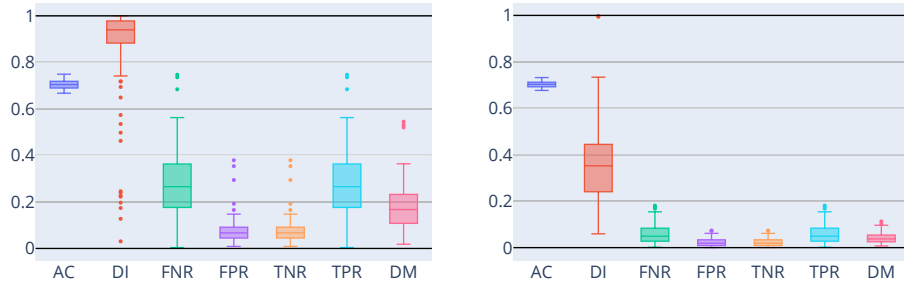


Figure 8. Post-processing results in mixed models for disparate mistreatment: First row: Comparison for logistic regression. Second row: Comparison for support vector machine. Left: Only post-processing. Right: In-processing and post-processing.

7. CONCLUSION

In this work we propose `FairML.jl` a `Julia` package that addresses fairness for classification in machine learning, offering a versatile tools to tackle unfairness at various stages of the classification process, providing users with more choices and control.

The first step, called preprocessing phase, employs a resampling method to mitigate disparate impact. This method utilizes a mixed strategy that combines undersampling and cross-validation.

In the in-processing phase, we extended the original optimization problems of support vector machine and logistic regression to address unfairness in the presence of group bias within the data. Specifically, we propose constrained optimization models that mitigate unfairness in heterogeneous populations. This phase also allows the utilization of any binary classifier from package `MLJ.jl` as learning tool.

Additionally, this paper proposes a post-processing method designed to identify a solution that improves the specified fairness metric, given by the user, without significantly compromising accuracy.

With simulations, we showcased how our approach reduces unfairness in the three phases. We also conducted some cross-phase combinations that can further enhance the final solutions.

To improve the framework’s capabilities, future work focuses on incorporating additional fairness metrics, and to adapt the phases to deal with multiclass classification.

ACKNOWLEDGEMENTS

The authors are grateful for the support of the German Federal Ministry of Education and Research (BMBF) for this research project, as well as for the “OptimAgent Project”.

We would also like to express our sincere appreciation for the generous support provided by the German Research Foundation (DFG) within Research Training Group 2126 “Algorithmic Optimization”.

REFERENCES

- Aghaei, S, M. J. Azizi, and P Vayanos (2019). “Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making.” In: *CoRR* abs/1903.10598. arXiv: [1903.10598](https://arxiv.org/abs/1903.10598). URL: <http://arxiv.org/abs/1903.10598>.
- Agrawal, A, J Chen, S Vollmer, and A Blaom (Aug. 2020). *ashryaagr/Fairness.jl*. Version v0.1.2. DOI: [10.5281/zenodo.3977197](https://doi.org/10.5281/zenodo.3977197). URL: <https://doi.org/10.5281/zenodo.3977197>.
- Barili, F, A Parolari, P. A. Kappetein, and N Freemantle (2018). “Statistical Primer: heterogeneity, random-or fixed-effects model analyses?” In: *Interactive cardiovascular and thoracic surgery* 27.3, pp. 317–321.
- Barocas, S and A. D. Selbst (2016). “Big data’s disparate impact.” In: *Calif. L. Rev.* 104, p. 671.
- Berk, R, H Heidari, S Jabbari, M Joseph, M Kearns, J Morgenstern, S Neel, and A Roth (2017). “A convex framework for fair regression.” In: *arXiv preprint arXiv:1706.02409*.
- Berman, E and J Ginesin (2024). “The State of Julia for Scientific Machine Learning.” In: *arXiv preprint arXiv:2410.10908*.
- Besançon, M, T Papamarkou, D Anthoff, A Arslan, S Byrne, D Lin, and J Pearson (2021). “Distributions.jl: Definition and Modeling of Probability Distributions in the JuliaStats Ecosystem.” In: *Journal of Statistical Software* 98.16, pp. 1–30. DOI: [10.18637/jss.v098.i16](https://doi.org/10.18637/jss.v098.i16).
- Bezanson, J, A Edelman, S Karpinski, and V. B. Shah (2017). “Julia: A fresh approach to numerical computing.” In: *SIAM review* 59.1, pp. 65–98. DOI: [10.1137/14100067](https://doi.org/10.1137/14100067).
- Blagus, R and L Lusa (2015). “Joint use of over-and under-sampling techniques and cross-validation for the development and assessment of prediction models.” In: *BMC bioinformatics* 16, pp. 1–10.
- Bono, R, R Alarcón, and M. J. Blanca (2021). “Report Quality of Generalized Linear Mixed Models in Psychology: A Systematic Review.” In: *Frontiers in Psychology* 12. DOI: [10.3389/fpsyg.2021.666182](https://doi.org/10.3389/fpsyg.2021.666182).
- Bouchet-Valat, M and B Kamiński (2023). “DataFrames.jl: Flexible and Fast Tabular Data in Julia.” In: *Journal of Statistical Software* 107.4, pp. 1–32. DOI: [10.18637/jss.v107.i04](https://doi.org/10.18637/jss.v107.i04).
- Browne, M. W. (2000). “Cross-validation methods.” In: *Journal of mathematical psychology* 44.1, pp. 108–132.
- Burgard, J. P. and J. V. Pamplona (2024a). *Fair Generalized Linear Mixed Models*. arXiv: [2405.09273](https://arxiv.org/abs/2405.09273) [cs.LG].

- Burgard, J. P. and J. V. Pamplona (2024b). *Fair Mixed Effects Support Vector Machine*. arXiv: [2405.06433](https://arxiv.org/abs/2405.06433) [cs.LG].
- Casals, M, M Girabent-Farrés, and J. L. Carrasco (2014). “Methodological Quality and Reporting of Generalized Linear Mixed Models in Clinical Medicine (2000–2012): A Systematic Review.” In: *PLoS ONE* 9.
- Caton, S and C Haas (2020). “Fairness in machine learning: A survey.” In: *ACM Computing Surveys*.
- Cheong, K. C., A. F. Yusoff, S. M. Ghazali, K. H. Lim, S Selvarajah, J Haniff, G. L. Khor, S Shahar, R. J. Abd, A. A. Zainuddin, et al. (2013). “Optimal BMI cut-off values for predicting diabetes, hypertension and hypercholesterolaemia in a multi-ethnic population.” In: *Public health nutrition* 16.3, pp. 453–459.
- Christ, S, D Schwabeneder, C Rackauckas, M. K. Borregaard, and T Breloff (2023). “Plots.jl – a user extendable plotting API for the julia programming language.” In: DOI: <https://doi.org/10.5334/jors.431>.
- Cruz, A. F., C Belém, S Jesus, J Bravo, P Saleiro, and P Bizarro (2023). *FairGBM: Gradient Boosting with Fairness Constraints*. arXiv: [2209.07850](https://arxiv.org/abs/2209.07850) [cs.LG]. URL: <https://arxiv.org/abs/2209.07850>.
- Das, S, M Donini, J Gelman, K Haas, M Hardt, J Katzman, K Kenthapadi, P Larroy, P Yilmaz, and M. B. Zafar (2021). “Fairness measures for machine learning in finance.” In: *The Journal of Financial Data Science*.
- Do, H, P Putzel, A. S. Martin, P Smyth, and J Zhong (2022). “Fair generalized linear models with a convex penalty.” In: *International Conference on Machine Learning*. PMLR, pp. 5286–5308.
- Good, P (2013). *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media.
- Green, B (2018). “Fair risk assessments: A precarious approach for criminal justice reform.” In: *5th Workshop on fairness, accountability, and transparency in machine learning*, pp. 1–5.
- Hardt, M, E Price, and N Srebro (2016). “Equality of opportunity in supervised learning.” In: *Advances in neural information processing systems* 29.
- Hearst, M. A., S. T. Dumais, E Osuna, J Platt, and B Scholkopf (1998). “Support vector machines.” In: *IEEE Intelligent Systems and their applications* 13.4, pp. 18–28.
- Hsieh, C. J., K Chang, and C. J. Lin (Jan. 2008). “A dual coordinate descent method for large-scale linear SVM.” In: *Proceedings of the Twenty-fifth International Conference on Machine Learning*, pp. 1369–1398.
- Jesus, S, P Saleiro, B. M. Jorge, R. P. Ribeiro, J Gama, P Bizarro, R Ghani, et al. (2024). “Aequitas Flow: Streamlining Fair ML Experimentation.” In: *arXiv preprint arXiv:2405.05809*.
- Lohr, S. L. (Feb. 2009). *Sampling : Design and Analysis*. 2nd ed. Florence, KY: Brooks/Cole.

- Lubin, M, O Dowson, J. D. Garcia, J Huchette, B Legat, and J. P. Vielma (2023). “JuMP 1.0: Recent improvements to a modeling language for mathematical optimization.” In: *Mathematical Programming Computation*. DOI: [10.1007/s12532-023-00239-3](https://doi.org/10.1007/s12532-023-00239-3).
- Menon, A. K. and R. C. Williamson (2018). “The cost of fairness in binary classification.” In: *Conference on Fairness, accountability and transparency*. PMLR, pp. 107–118.
- Mohammed, R, J Rawashdeh, and M Abdullah (2020). “Machine learning with oversampling and undersampling techniques: overview study and experimental results.” In: *2020 11th international conference on information and communication systems (ICICS)*. IEEE, pp. 243–248.
- Neter, D. J., M. H. Kutner, and C. J. Nachtsheim (2004). *MP Applied Linear Regression Models-Revised Edition with Student CD*. McGraw-Hill Education.
- Olfat, M and A Aswani (2017). “Spectral Algorithms for Computing Fair Support Vector Machines.” In: *CoRR* abs/1710.05895. arXiv: [1710.05895](https://arxiv.org/abs/1710.05895). URL: <http://arxiv.org/abs/1710.05895>.
- Radovanović, S, A Petrović, B Delibašić, and M Suknović (2020). “Enforcing fairness in logistic regression algorithm.” In: *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–7. DOI: [10.1109/INISTA49547.2020.9194676](https://doi.org/10.1109/INISTA49547.2020.9194676).
- Ren, Q, C Su, H Wang, Z Wang, W Du, and B Zhang (2016). “Prospective study of optimal obesity index cut-off values for predicting incidence of hypertension in 18–65-year-old Chinese adults.” In: *PloS one* 11.3, e0148140.
- Scutari, M (2023). *fairml: A Statistician’s Take on Fair Machine Learning Modelling*. arXiv: [2305.02009](https://arxiv.org/abs/2305.02009) [stat.ML]. URL: <https://arxiv.org/abs/2305.02009>.
- Vapnik, V and A. Y. Chervonenkis (1964). “A class of algorithms for pattern recognition learning.” In: *Avtomat. i Telemekh* 25.6, pp. 937–945.
- Wächter, A and L. T. Biegler (2006). “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.” In: *Mathematical programming* 106, pp. 25–57.
- Yang, J, N. A. Zaitlen, M. E. Goddard, P. M. Visscher, and A. L. Price (2014). “Advantages and pitfalls in the application of mixed-model association methods.” In: *Nature genetics* 46.2, pp. 100–106.
- Zafar, M, I Valera, M Rodriguez, and K. P. Gummadi (Oct. 2016). “Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment.” In.
- Zafar, M. B., I Valera, M Gomez-Rodriguez, and K. P. Gummadi (2017). “Fairness Constraints: Mechanisms for Fair Classification.” In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by A. Singh and J. Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, pp. 962–970.

- Zafar, M. B., I Valera, M Gomez-Rodriguez, and K. P. Gummadi (2019). “Fairness Constraints: A Flexible Approach for Fair Classification.” In: *Journal of Machine Learning Research* 20.75, pp. 1–42.
- Zhang, W, A Bifet, X Zhang, J. C. Weiss, and W Nejdl (2021). “FARF: A Fair and Adaptive Random Forests Classifier.” In: *CoRR* abs/2108.07403. arXiv: 2108.07403. URL: <https://arxiv.org/abs/2108.07403>.
- Zhao, H and G. J. Gordon (2022). “Inherent tradeoffs in learning fair representations.” In: *Journal of Machine Learning Research* 23.57, pp. 1–26.
- Zughrat, A, M Mahfouf, Y. Y. Yang, and S Thornton (2014). “Support vector machines for class imbalance rail data classification with bootstrapping-based over-sampling and under-sampling.” In: *IFAC Proceedings Volumes* 47.3, pp. 8756–8761.

(J. P. Burgard, J. V. Pamplona) TRIER UNIVERSITY, DEPARTMENT OF ECONOMIC AND SOCIAL STATISTICS, UNIVERSITÄTSRING 15, 54296 TRIER, GERMANY

Email address: burgardj@uni-trier.de, pamplona@uni-trier.de