Not So Fair: The Impact of Presumably Fair Machine Learning Models

Mackenzie Jorgensen
mackenzie.jorgensen@kcl.ac.uk
King's College London
London, UK

Hannah Richert
hrichert@uni-osnabrueck.de
Universität Osnabrück
Osnabrück, Germany

Elizabeth Black
elizabeth.black@kcl.ac.uk
King's College London
London, UK

Natalia Criado
ncriado@upv.es
Universidad Politècnica de València
València, Spain

Jose Such
jose.such@kcl.ac.uk
King's College London
London, UK

ABSTRACT

When bias mitigation methods are applied to make fairer machine learning models in fairness-related classification settings, there is an assumption that the disadvantaged group should be better off than if no mitigation method was applied. However, this is a potentially dangerous assumption because a "fair" model outcome does not automatically imply a positive impact for a disadvantaged individual—they could still be negatively impacted. Modeling and accounting for those impacts is key to ensure that mitigated models are not unintentionally harming individuals; we investigate if mitigated models can still negatively impact disadvantaged individuals and what conditions affect those impacts in a loan repayment example. Our results show that most mitigated models negatively impact disadvantaged group members in comparison to the unmitigated models. The domain-dependent impacts of model outcomes should help drive future bias mitigation method development.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Social and professional topics** → *User characteristics.*

KEYWORDS

fairness, impact, machine learning, synthetic data

ACM Reference Format:
Mackenzie Jorgensen, Hannah Richert, Elizabeth Black, Natalia Criado, and Jose Such. 2023. Not So Fair: The Impact of Presumably Fair Machine Learning Models. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES '23), August 8–10, 2023, Montréal, QC, Canada.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3600211.3604699

1 INTRODUCTION

The issue of algorithmic decision-making systems making harmful or discriminatory predictions is well-recognized (e.g., [7, 10, 15, 21, 28, 30, 31, 36, 37, 39, 40]). Algorithmic fairness research evolved in response to this, primarily focusing on optimizing for some fairness notion to prevent harm. Bias mitigation methods have been developed for different points along the Machine Learning (ML) pipeline and fairness constraints have operationalized fairness notions which are often dependent on conditional probabilities relating to model outcomes (e.g., [1, 2, 8, 9, 14, 17, 18, 41]). However, there is little consensus on when to use which bias mitigation method or constraint [11, 12, 15, 27].

Recent research has shown that "fair" outcomes and benefits for individuals are not always aligned, highlighting that algorithmic fairness sometimes falls short of its main goal of minimizing harm [13, 20, 25, 26, 35]. We call a model that has a bias mitigation method applied to it a mitigated model. Fairness disparity metrics measured after model training show how well a model satisfies a fairness constraint (e.g., Equality of Opportunity) that represents a fairness goal (e.g., groups should have equal true positive rates). Fairness constraints can aid in bias detection (with fairness metric disparities) and bias mitigation by constraining a model's training to satisfy a fairness goal. Sometimes fairness constraints, when used for bias mitigation, make an assumption that the positive class is beneficial. However, if that assumption is not valid, then applying the bias mitigation methods can result in fairer outcomes but worse potential impacts for individuals, especially for disadvantaged groups.

For instance, in the case of loan repayment, let us assume a bank developed a mitigated model that predicts an applicant's ability to repay the bank if given a loan. If the fairness constraint, Demographic Parity (DP), is used, then the selection rates (positive class rates) across the groups should be equal. This could result in a high false positive rate for the disadvantaged group. If an individual is falsely classified and expected to repay but defaults, because the DP constraint assumed a positive outcome was beneficial for them, then that positive outcome in fact had a negative impact on them. The example emphasizes our motivating problem that while fairness disparities for a mitigated model might be low, the individuals classified could still be negatively impacted which is a major issue.

Previous works began investigating how to quantify impact and what its relationship with fair decision-making is [13, 20, 24–26]. These works considered models such as a temporary labor market model [20], threshold optimization [26], causal models [24], agent-simulations [13], and multi-armed bandits [25]. They did not

AIES '23, August 8–10, 2023, Montréal, QC, Canada

ACM ISBN 979-8-4007-0231-0/23/08…$15.00
https://doi.org/10.1145/3600211.3604699

consider a model in a classic binary classification setting though. Since many bias mitigation methods that constrain model learning with fairness constraints apply to classification settings, these works could not consider how different mitigation methods could have affected their results. When fairness constraints were used in previous works, only one or two fairness constraints were considered. They also did not consider how the datasets themselves could have played a role in their impact results; for instance, what if the disadvantaged group was not in the minority?

In this paper, we investigate the question: when a false positive model outcome may have a negative impact, can mitigated models in a binary classification setting do more harm than good for the disadvantaged group? We explore this question through a case study of the loan repayment example aforementioned. We focus on the disadvantaged group because we aim to minimize negative impacts that that group experiences from supposedly "fair" models, while also considering the effect on the advantaged group. Our objectives that support us in answering our research question are: (1) quantifying the impact of different model outcomes, allowing us to explore specific cases where a positive outcome does not necessarily imply a positive impact, and (2) analyzing how different fairness constraints and dataset makeup relate to the impacts by group. We use DP and four other fairness constraints with several off-the-shelf ML models with different bias mitigation methods to apply the constraints to better understand the relationship between fairness constraint choice and impact.

In addition, we explore the effect of the dataset makeup on impacts too, motivated by this question: if we adjust the demographic group representation and increase the number of disadvantaged applicants who repay the bank, do we see a positive impact on the disadvantaged group? To begin to answer this in our experiments, we use synthetic datasets alongside a real-world dataset for comparison. We vary the synthetic datasets with two parameters: demographic group representation, which shows what data proportions are made up of disadvantaged and advantaged individuals, and repayment label composition by group, which shows if an individual repays or defaults if given a loan[1]—when we discuss dataset composition, we refer to these parameters.

Our results highlight that achieving good fairness disparity metric values and low negative impact results for the disadvantaged group are often in conflict with one another. As a result, the majority of the mitigated models tested actually leave the disadvantaged group worse off than the unmitigated models from an impact stand point. Also, the dataset composition did not have much of an effect on the impact results. The rest of the paper is structured as follows: the literature review in Section 2, the definitions in Section 3, the methodology for our work in Section 4, the experimental setup in Section 5, the results in Section 6, a discussion of the results in Section 7, and our conclusion in Section 8.

2 LITERATURE REVIEW

Our research is about algorithmic fairness in classification settings and impact considerations. The algorithmic fairness community has presented multiple fairness constraints for bias detection such as Equality of Opportunity (EOO) and Equalized Odds (EO) (e.g., [2, 9, 14, 18, 41]) and bias mitigation methods for mitigating unfairness at different stages along the ML pipeline (e.g., [1, 2, 8, 14, 17, 18, 22, 23, 28]). These methods were developed to answer a call for more safe and trusted algorithms, after algorithmic harms were highlighted in multiple domains from online ad delivery to hiring (e.g., [4, 6, 7, 31, 37]). The choice of bias mitigation methods and fairness constraints should be informed by the domain [16, 27].

While aiming to make ML models more "fair" is a valuable goal, scholars argue that we must look at the actual impacts from model outcomes to understand whether individuals were positively affected by the model and begin to explore how to quantify impact in different settings from labor market models to multi-armed bandits [13, 20, 24–26]. We describe the most relevant work to our's from this strand of research—Liu *et al.* coined the term "delayed impact" and conducted experiments using the loan repayment example to see if disadvantaged groups are better off in terms of delayed impact when optimizing for thresholds under fairness constraints [26]. We extend this work with the same example but instead of focusing on class probabilities, we focus on class labels. We also take this research further by using multiple ML models with bias mitigation methods to apply more fairness constraints than previously considered, by using synthetic datasets of varying compositions to test how that affects the impact results, and by modeling impact in different ways than before.

While some researchers generate data to make their data discrimination free or more fair from a causal lens [38, 43, 44], we generate synthetic datasets, not with a de-biasing goal, but to represent different dataset compositions from which models can learn and we can study their effects on impact. Friedler *et al.* conducted a comparative study of bias mitigation methods and found that these methods were sensitive to feature distributions in datasets [16], providing motivation for our consideration of dataset composition since we also use different mitigation methods. Zafar creates synthetic datasets with varying correlations between the sensitive feature and the label to analyze the relationship between the accuracy and discrimination [45]. Reddy *et al.* test numerous models with various synthetic dataset configurations to analyze the models' fairness performances [33]. Similarly to Reddy *et al.* and Zafar, we control demographic group representation and repayment label composition in our synthetic datasets. We do this to better understand the relationship between dataset composition and impacts from different mitigated models.

3 PRELIMINARIES

In this section, we outline the formalizations we use for our experiments and explain what we mean by impact.

3.1 Definitions

In our paper, we consider a binary supervised learning setting. A **dataset** is $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in X$ is an **instance**, $y_i \in \{0, 1\}$ is the **true label** of x_i, and n is the **number of samples in** X. D is split into two subsets, $D_{train} \subset D$ and $D_{test} \subset D$, such that $(D_{train} \cup D_{test}) = D$ and $(D_{train} \cap D_{test}) = \{\}$. To train a classifier model, the instances must undergo feature encoding where information is extracted from each instance into features

[1]The repayment label is what the model is trying to learn from the data.

that are categorical and/or numerical; each instance $x \in X$ is a k-dimensional **feature** vector $\langle f_1^x, \ldots, f_k^x \rangle$.

We are interested in problems where instances will contain, directly or indirectly, personal information about individuals. One such feature that we assume is in X is a **protected attribute** which is sensitive in nature (e.g., race or gender) [34, 42]. This attribute can be strongly associated with other features. The actual constructs that are considered a protective attribute depend on the domain and legal context. For any instance, $x = \langle f_1^x, \ldots, f_k^x \rangle$, we assume the first feature, f_1^x, is the protected attribute which can have values of $f_1^x \in \{0, 1\}$, where 0 represents a disadvantaged group and 1 represents an advantaged group.[2] We assume the disadvantaged group is underprivileged (often due to systemic power structures, inequity, and oppression) in comparison to the advantaged group which is privileged.

The protected attribute allows us to split instances into two groups (D_0 and D_1), where the subindex denotes the value of the protected attribute (e.g. $D_i = \{(\langle f_1^x, \ldots, f_k^x \rangle, y) | (\langle f_1^x, \ldots, f_k^x \rangle, y) \in D$ and $f_1^x = i\}$.) In addition, instances can also be split according to their label into D^1 and D^0, where the superindex denotes the value of the label (e.g. $D^i = \{(\langle f_1^x, \ldots, f_k^x \rangle, y) | (\langle f_1^x, \ldots, f_k^x \rangle, y) \in D$ and $y = i\}$.) Finally, the set D_j^i denotes the set of instances where the label takes value i and the protected attribute takes value j.

We define a **deterministic classifier** as a function, $h : X \longrightarrow \hat{Y}$, where $\hat{Y} = \{0, 1\}$. For any instance of x, $h(x)$ is the prediction returned from the classifier. The function h approximates a true function, representing the population, $t : X \longrightarrow Y$, where $Y = \{0, 1\}$ and for any instance x, $t(x)$ is the true label of x. We denote the prediction of a particular instance of x as $h(x) = \hat{y}_x$ and we denote the true label of a particular instance x as $t(x) = y_x$. The conditional probability that h, outputs a given prediction, $\hat{y}$, given a protected attribute, a, is denoted as $P(\hat{y}|a)$ where $\hat{y} \in \{0, 1\}$ and $a \in \{0, 1\}$. To analyze a classifier's performance, confusion matrices which show model outcomes are commonly used. The model outcomes are the True Positives (TP), False Positives (FP or Type I Error), True Negatives (TN), and False Negatives (FN or Type II Error), when looking at the predicted and true labels. Many fairness constraints can also be explained by TP, FP, TN, and FN [29].

3.2 Impact

We argue it is crucial to consider different ways that impact might relate to model outcomes.[3] If impact is not considered, then mitigated models could actually cause more harm to disadvantaged groups under certain conditions. The loan repayment example from before highlights this problem: where a fair outcome based on a DP-constrained model resulted in a FP applicant who was negatively impacted because they defaulted, since they were unable to repay the bank. There is a tension between benefits of certain model outcomes, like being granted a loan as a FP, and the actual impacts they have on individuals, defaulting on said loan.

In this paper, we assume that a classifier h's impact is a function of instances dependent on model predictions and true labels that outputs weights, i_h: $X \rightarrow W$, such that for any instance x, the weight w, returned by $i_h(x)$ depends on $\hat{y}_x$ and y_x and $w \in \mathbb{R}$.[4] The weight represents the impact of a model outcome for a given instance and can be deterministically or non-deterministically generated (according to some distribution like a Normal distribution). We note here, though, that when i_h provides non-deterministically generated weights as outputs we take liberties with the function, since a function must map every input value to a single output value. But, in this case, the same input could have different valued outputs because the output is dependent on the distribution. We define impact more specifically for our loan repayment example below in Section 5.4.

4 METHOD

We return to our research question: assuming that a false positive model outcome does not have a positive impact, can mitigated models negatively impact the disadvantaged group rather than positively impact them? We explore this question in a binary classification setting with a loan repayment example. The main objectives of this paper are to quantify the impact of model outcomes in different ways and to analyze the relationships between impact and dataset composition and impact and fairness constraints. To do this, we perform experiments, controlling for different variables like the datasets and their compositions, bias mitigation methods, fairness constraints, ML model choice, and impact functions to help us study impact. We provide a visualization of our experimental pipeline from a high level in Figure 1 and explain details more below.

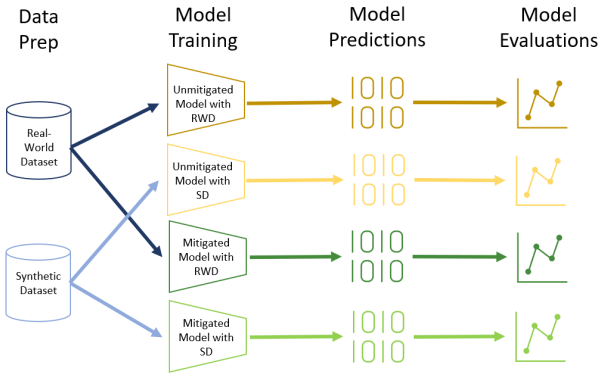

Figure 1: Our experimental pipeline follows a typical ML pipeline. We show how a real-world dataset (RWD) and one of our synthetic datasets (SD) would be pushed through the pipeline. For each dataset, we train an unmitigated model and multiple mitigated models (we only visualize one mitigated model here for simplicity purposes). The top two models are unmitigated models and the bottom two are mitigated models which would have a bias mitigation method applied with a chosen fairness constraint. Multiple runs would need to happen to go through all the different combinations of ML models, fairness constraints, and bias mitigation methods.

[2]Note our work can easily be extended to consider more than two labels for the protected attribute.
[3]We highlight that another way to think about impact is expected utility.
[4]Since we assume weights are real numbers, categorical weights can also be considered if transformed into numbers.

As mentioned before, we use multiple datasets so they are all funnelled through this pipeline multiple times to account for different ML model and bias mitigation method choices. For a real-world dataset, we transform the FICO scores dataset from over 300k TransUnion TransRisk scores from 2003 that was preprocessed by Hardt *et al.* into a tabular loan repayment dataset where each row corresponds to an individual loan applicant [18]. The FICO scores dataset as is contains a cumulative distribution function (CDF) providing the fraction of the racial group that falls below a given credit score and a probability mass function (PMF) showing the probability of an applicant repaying the bank given their race and credit score. We are interested in a tabular dataset so we can apply common bias mitigation methods with fairness constraints to study how they affect impact. We also assemble synthetic datasets with different demographic group representations and repayment label compositions; experiments with these synthetic datasets allow us to understand how the dataset composition affects the impact on different groups.

With a dataset for each experimental run, we then train off-the-shelf ML models that are mitigated during training using different reduction algorithms, our bias mitigation method of choice, that can use different fairness constraints each time (see Section 5.2 for details). By using reduction algorithms, we could simply change the constraint to be used for each experiment in an agnostic way. For each mitigated model, we run one reduction algorithm paired with one fairness constraint until we complete every combination. Models with no bias mitigation method applied during training we call unmitigated models. We train these mitigated and unmitigated models on the loan repayment dataset and the synthetic datasets.

After receiving the mitigated and unmitigated model predictions for all of our experiment runs, we evaluate the models. To do so, we calculate their model accuracy, fairness disparity metric, and impact results. We check the model performance and bias because we aim to develop well-performing and fair models. The fairness disparity metric results show us whether the mitigated model performs as well as the unmitigated model, how effective the bias mitigation method is at satisfying a fairness constraint, and whether the application of a particular mitigation method and constraint negatively impacts the disadvantaged group.

5 EXPERIMENTAL SETUP

In this section, we present the fairness constraints, ML models, bias mitigation methods, datasets, and impact functions that we use. We assume a white-box scenario where we have access to data, models, and model outputs. Recall that we consider a binary classification problem where a model predicts if a loan applicant will repay the bank if given a loan.

5.1 Fairness Constraints

We focus on group fairness which aims to identify what groups are at risk of being harmed [14]. Group fairness is defined in terms of constraints on a model called fairness constraints or parity constraints (we will use the former term). We explain the group fairness constraints considered for our experiments in Table 1 and refer to them primarily by the acronyms stated there.[5] These metrics were chosen because of their canonical nature within the algorithmic fairness domain and their availability in open-source fairness toolkits and libraries [3, 5]. Also, expert knowledge is not required to use them. All of our metrics are Bias Preserving which has an underlying assumption that the status quo is the baseline for equality across groups except for one which is Bias Transforming, DP, which assumes that protected groups, from an equality standpoint, start at different points [42]. To measure the level of fairness in a model, we take the fairness disparity metric value, telling us how well the model abides by a given fairness constraint.

5.2 ML Models and Reduction Algorithms

We utilize off-the-shelf ML models from *sklearn* to make our experiments easily replicable [32]. In our experiments, we use the following models: Decision Tree (DT), Gaussian Naive Bayes (GNB), Logistic Regression (LGR), and Gradient Boosted Trees (GBT) classifiers. We also chose these models because their fit functions had a sample weights parameter—this parameter is necessary for the reduction algorithms in *Fairlearn* [5]. For ML model performance, we consider accuracy as our metric of choice which is common to consider in the algorithmic fairness literature.

Microsoft's *Fairlearn* toolkit implements Agarwal *et al.*'s bias mitigation method which includes two reduction algorithms, Exponentiated Gradient and Grid Search; we use this mitigation method in our experiments. The reduction algorithms take the parameters: an already trained ML model and a fairness constraint, and then narrow the binary classification to weighted classification problems that focus on achieving strong performing models for certain classes. The algorithms' goal is to optimize the trade-off between the chosen fairness constraint and the model's accuracy. In the reduction algorithm, the fairness constraints are transformed into Lagrange multipliers. We encourage the reader to read Agarwal *et al.*'s paper for a more in-depth understanding of the reduction algorithms [1]. Reduction algorithms are versatile because they allow developers the choice in their ML model, unlike other fairness methods applied during training which are often model-specific.

5.3 Datasets

In our experiments, we have a dataset which represents the real-world and then we have eight synthetic datasets that represent different potential scenarios. For model training, we split each of the datasets into 70% train and 30% test sets. For testing the synthetic datasets, we use two different test sets. The first test set is the test set created when we split the synthetic dataset. The second test set matches the real-world dataset's composition. This real-world test set allows us to test how well the model trained on a synthetic dataset performs on a subset of the real-world's population.

5.3.1 Baseline Dataset. We transformed the FICO scores dataset from 2003 preprocessed by Hardt *et al.* into a tabular dataset that can be used in a binary classification setting which we call our

[5]Note that some metrics have different names in the literature so we try to clear up any confusion: DP is sometimes called Statistical Parity and Acceptance Rate. EO in previous literature is referred to as Disparate Mistreatment. EOO has a mathematical equivalent metric in False Negative Error Rate balance [9]. False Positive Rate Parity (FPRP) or False Positive Error Rate balance is also sometimes referred to as Predictive Equality and is linked to the True Negative Rate.

Table 1: The fairness constraints we consider in our experiments are listed and defined, where $y \in \{0, 1\}$.

Name	Expression
Demographic Parity (DP) [14]	$P(\hat{Y} = 1\|A = 0) = P(\hat{Y} = 1\|A = 1)$
Equalized Odds (EO) [18]	$P(\hat{Y} = 1\|Y = y, A = 0) = P(\hat{Y} = 1\|Y = y, A = 1)$
Equality of Opportunity (EOO) [18]	$P(\hat{Y} = 1\|Y = 1, A = 0) = P(\hat{Y} = 1\|Y = 1, A = 1)$
False Positive Rate Parity (FPRP) [9]	$P(\hat{Y} = 1\|Y = 0, A = 0) = P(\hat{Y} = 1\|Y = 0, A = 1)$
Error Rate Parity (ERP) [2]	$P(\hat{Y} = y\|Y \neq y, A = 0) = P(\hat{Y} = y\|Y \neq y, A = 1)$

baseline dataset [18]. The data is composed of FICO scores (for showing credit worthiness). We note that Liu *et al.* also used the same dataset for their impact experiments [26]. The credit scores ranged from 300 to 850 and the authors assumed the Black group as disadvantaged and the White group as advantaged.

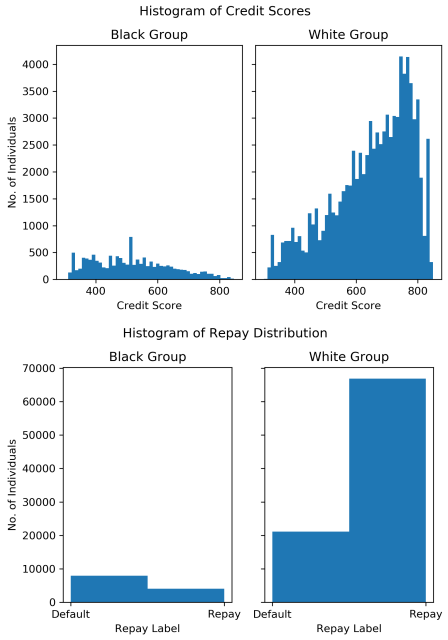

Figure 2: We show the baseline dataset composition for the credit scores and repayment labels by group.

For our baseline dataset, we generated 100k rows or 100k individuals based upon Liu *et al.*'s dataset composition such that the same demographic group representation, credit score distributions by race based on the CDF, and repayment label distributions based on the PMF are upheld. Our rationale behind that dataset composition is that it matches the real-world dataset. Each row or individual has two features: credit score and race which make up X. The dataset also contains labels, Y, for whether the loan could be repaid by the individual.[6] Our baseline dataset labels are generated from the PMFs for an individual repaying given their credit score. For visualizations of the dataset concerning the credit scores and repayment labels by group, see Figure 2. We use one algorithm to create our baseline dataset, similar to Liu *et al.*'s method [26]. The Algorithm 1 (see Appendix A.3) generates a dataset based on two parameters, demographic group representation and order of magnitude (for the dataset size), by using the CDF and PMF.

5.3.2 Synthetic Datasets. Since the baseline dataset is imbalanced considering the demographic group representation (12% Black and 88% White) and the disadvantaged group's repayment label composition (see the bottom left plot of Figure 2), we change those parameters when generating synthetic datasets. These synthetic datasets let us test how impact is affected by varying dataset compositions. We consider cases when the disadvantaged group is the majority, in the minority (matching the baseline dataset), and when the two groups have equal representation. We keep the credit score distributions the same and adjust the disadvantaged group's repayment label composition for only some runs so we can see the effect of the altered demographic distributions. In addition, by adjusting the disadvantaged group's repayment label composition, we oversample for instances where the disadvantaged group repays the bank so the models learn from a more balanced dataset. We do not adjust the advantaged group's repayment labels since we are primarily focused on minimizing harm to the disadvantaged group.

We use the following two ratios to generate synthetic datasets and Table 2 shows the ratios we have for all our datasets—the baseline dataset is included there as well. To generate these datasets, we extend Algorithm 1 to adjust the repayment label ratios for the disadvantaged group through Algorithm 2 and 3 in Appendix A.3.

Definition 5.1. The **demographic-ratio** is $R = r_0 : r_1$ such that $r_0 = |D_0|/|D|$ and $r_1 = |D_1|/|D|$.

Definition 5.2. The **label-ratio** is $L_a = |D_a^0| : |D_a^1|$ such that $|D_a^0|$ is the number of negative instances for a group $a \in 0, 1$ and $|D_a^1|$ is the number of positive instances for that same group.

Each scenario in Table 2 is categorized with a two-letter code such that the first letter represents the demographic-ratio and the second letter represents the disadvantaged group's label-ratio. A "0" means the ratio matches the baseline dataset, "i" means the ratio is imbalanced (not the same imbalanced ratio as the baseline however), and "b" means the ratio is balanced.[7] The "00" scenario represents the baseline dataset; while, as another example, the disadvantaged group is in the majority and the disadvantaged group's repayment label-ratio match the baseline's label-ratio in the "i0" scenario.

[6]For more information about how we transformed the initial FICO score dataset into a tabular dataset, see the GitHub: https://github.com/mjorgen1/explore-fair-impacts.

[7]The baseline dataset ratios are imbalanced as well but we did not force them to be.

Table 2: The dataset names and parameters used as constraints when generating the datasets for our experiements. Note that we only specify the disadvantaged group's label-ratio here, not the advantaged group's label-ratio which remains unchanged.

Dataset Name [Label]	Demographic-Ratio	Disadvantaged-Label-Ratio
Baseline [00]	0.12 : 0.88	0.66 : 0.34
Demo-Bal-Repay-Baseline [b0]	0.5 : 0.5	0.66 : 0.34
Demo-Imbal-Repay-Baseline [i0]	0.88 : 0.12	0.66 : 0.34
Demo-Baseline-Repay-Bal [0b]	0.12 : 0.88	0.5 : 0.5
Demo-Bal-Repay-Bal [bb]	0.5 : 0.5	0.5 : 0.5
Demo-Imbal-Repay-Bal [ib]	0.88 : 0.12	0.5 : 0.5
Demo-Baseline-Repay-Imbal [0i]	0.12 : 0.88	0.34 : 0.66
Demo-Bal-Repay-Imbal [bi]	0.5 : 0.5	0.34 : 0.66
Demo-Imbal-Repay-Imbal [ii]	0.88 : 0.12	0.34 : 0.66

5.4 Impact and Credit Scores

In our example, we assume that TPs and FPs are granted loans. The applicants who were classified as TNs or FNs would most likely no longer be followed up with by the bank after the rejection notification, so data on the actual impacts of those model outcomes are not available. As a result of this, we focus on the impact of the TP and FP model outcomes. We note that different model errors can lead to different impacts [19].

We take inspiration from Liu *et al.*'s focus on predatory lending for our impact measurement [26]. The credit score of applicants is a key feature in our example. We assume that the change in credit score is related to the model outcome so we use that feature in our impact calculations. For any instance, $x = \langle f_1^x, \ldots, f_k^x \rangle$, we assume that the second feature, $f_2^x \in [300, 850]$, is the credit score for an applicant. We define a set $S = \{f_2^x\}$, where s_i holds the credit score for applicant x_i.

5.4.1 Deterministically Generated Weights. The deterministically generated weights reflect the credit score change values used in Liu *et al.*'s experiments such that the weight, $w = \{75, -150\}$, depends on if an applicant is a TP or FP respectively [26]. If a sample, x_i, is a TP, meaning the applicant is correctly predicted to repay the bank, then the s_i is increased by 75 points; if that applicant is deemed an FP, meaning they are incorrectly predicted to repay the bank, then the s_i is decreased by -150 points. For all of our datasets, we use these weights when calculating credit score changes.

Table 3: The mean, μ, and standard deviation, σ, values for generating the non-deterministically generated weights from Normal probability distributions.

Name	μ_{TP}	σ_{TP}	μ_{FP}	σ_{FP}
Benchmark	75	15	-150	15
Equal	100	15	-100	15
Benchmark-Swap	150	15	-75	15

5.4.2 Non-Deterministically Generated Weights. We also conduct experiments with the baseline dataset using non-deterministically generated weights for the impact function. We generate these weights for w through a Normal probability distribution (see Table 3). We use the deterministically generated weights as means for the Benchmark distributions for comparison purposes. We also consider two other scenarios where FP and TP model outcomes have opposite but equal valued effects, the Equal distributions, and where the TP is weighed even more heavily than a FP, the Benchmark-Swap distributions. The standard deviations were chosen by taking into account the empirical rule for Normal distributions and the limit of the credit score range since we wanted updated credit scores to abide by their constraints. We argue that using non-deterministically generated weights is a potentially better modeling of reality since the applicants' credit scores could drop or increase at different scales.

5.4.3 Measuring Average Impact. Now that we have defined our weight generation, we define average impact for this problem as the difference in credit scores after the predictions, $\hat{Y}$. We calculate the average impact by group for all of our experiments.

Definition 5.3. Classifier h's **average impact** on group a is:

$$\bar{I}_a = \frac{1}{|D_a|} \cdot \sum_{i \in D_a} s_i + w$$

6 RESULTS

We present our results below and remind the reader that the mitigated models are those that had a reduction algorithm with a fairness constraint applied. When providing the fairness disparity metric values for the unmitigated and mitigated models, we show all four ML model results. When we present impact results, these are calculated by taking the average impact by group which we define in Section 5.4. For the impact findings, we only ran experiments with a Decision Tree (DT) model. We chose the DT model because it generated more stable results than a Gaussian Naive Bayes (GNB) model and produced comparable results to Logistic Regression (LGR) and Gradient Boosted Trees (GBT) models when trained on our datasets (see Table 4 and 5, and Table 6 in the Appendix A.2). The demographic groups are differentiated by the labels: disadvantaged, Black, or "0" and advantaged, White, or "1."

6.1 Baseline Dataset Results

First, we analyze how unfair the unmitigated models are when trained on the baseline dataset and check how well the bias mitigation methods minimized that unfairness in the mitigated models. Table 4 displays the fairness disparity metric values for the unmitigated models and Table 5 shows how well the bias mitigation methods mitigated bias according to the fairness constraints applied. The smaller the fairness disparity metric value, the closer the model satisfies a fairness constraint. If a fairness disparity metric value is 0, a model is satisfying the fairness constraint completely. All mitigated models perform well by reducing the fairness disparity metric for the applied fairness constraint and exhibit similar results. For the model accuracy results, see Table 6 in the Appendix A.2—the unmitigated DT, LGR, and GBT models all reach 88% accuracy and that performance only dropped between 1% and 4% for the mitigated models which shows that even with bias mitigation methods applied the models still perform reasonably well.

Table 4: Values of the fairness disparity metrics for our unmitigated models when trained on the baseline dataset, where the rows are the ML models tested and the columns are the fairness constraints considered.

Classifier	DP	EO	EOO	FPRP	ERP
DT	49.40	23.77	23.77	20.44	4.45
GNB	82.55	96.4	96.4	38.09	21.62
LGR	49.03	22.1	21.38	22.1	3.79
GBT	46.23	18.6	18.6	18.27	4.16

Table 5: Values of the fairness disparity metrics for our mitigated models when trained on the baseline dataset, where the rows are the ML models tested and the columns are the fairness constraints applied with the Exponentiated Gradient reduction algorithm and measured for the disparity metric.

Classifier	DP	EO	EOO	FPRP	ERP
DT	0.45	3.77	1.88	0.34	1.16
GNB	0.85	2.18	1.18	0.29	0.1
LGR	0.83	2.49	0.9	0.59	0.3
GBT	0.65	2.84	1.41	0.05	1.44

We used Exponentiated Gradient for our reduction algorithm of choice for the remainder of our results after comparing the results with Grid Search. Exponentiated Gradient (see Table 5) was more effective than Grid Search (see Table 7 in Appendix A.2) at decreasing the fairness disparity metric values from the unmitigated model fairness disparity metric values (see Table 4). Since we are interested in how different mitigated models impact groups, we chose the reduction algorithm for our experiments that gave stronger fairness results.

Before we can check if our credit score distributions from mitigated models are statistically significant in comparison to the credit score distributions from unmitigated models, we must test if those distributions are Normal. We check if the updated credit score distributions for the baseline dataset with deterministically generated weights are Normal distributions by using the Kolmogorov-Smirnov test. Then, with our not-Normal updated credit score distributions, we use Mann-Whitney tests to look at discrepancies between the updated credit scores and unmitigated versus mitigated models. This analysis tells us if there are statistically significant changes to credit score distributions when using bias mitigation methods.

6.1.1 Impact with Deterministic Weights. By considering the impact for the disadvantaged group (see the top plot of Figure 3), we highlight that, even though we have an improvement in fairness (as shown in Table 4 and 5), the disadvantaged group the majority of the time experiences a negative impact. For all models, the worst impact occurs when DP is the fairness constraint. The few models that positively impact the disadvantaged group are the unmitigated and ERP-constrained models. Besides the ERP-constrained model, none of the mitigated model results could exceed the unmitigated positive impact. The lower plot of Figure 3 shows that the advantaged group always experiences a high positive impact across all mitigated models.

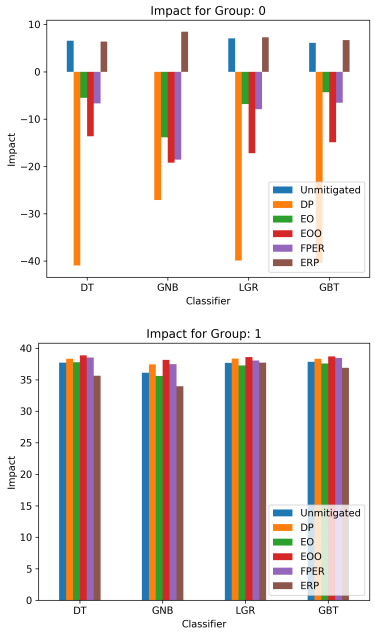

Figure 3: Impact for all classifier and fairness constraints when using the baseline dataset and when weights are deterministically generated.

We examine the statistical significance of how the impact on an individual affects the credit scores by demographic group; we compare the updated credit score distributions from each mitigated model (for all ML models) with the unmitigated model for each group. We update the credit scores based on the model outcomes given the deterministically generated weights. The updated credit score distributions for ERP-constrained models by demographic group are not statistically significant from the unmitigated model

for the disadvantaged group but this is most likely because the results are similar (see Figure 7 in the Appendix A.1). For all models with DP, EO, EOO, and FPRP as the fairness constraint, the change in the score distributions of the disadvantaged group is statistically significant. At the same time, the score distributions from the mitigated models for the advantaged group are not statistically significant, except for the ERP-constrained GNB model.

6.1.2 Impact with Non-deterministic Weights. Figure 4 displays the impact results for our groups with non-deterministically generated weights for impact when using a DT model. When we lessen the weight of an FP applicant and increase it for a TP applicant, the advantaged and disadvantaged groups impact increases, unsurprisingly. However, the DP-constrained model still has the lowest impact in comparison to other constraints for the disadvantaged group for all impact setups with non-deterministic weights.

When we compare the mitigated DT model impact results from Figure 3 with the Benchmark distribution impact results from Figure 4, we see that the results match up such that DP-constrained model has the lowest impact, followed by EOO, FPRP, and EO-constrained models. Similarly, the impact results for the ERP and unmitigated models are aligned. For the advantaged group, the results also are aligned such that the mitigated models do not change the advantaged group's impact much at all. We see the same statistically significant results as discussed in Section 6.1.1 and these results can be found in Figure 8 in the Appendix A.1.

When the TP and FP impacts have equal weight for the Equal distribution (see Figure 4), only the DP- constrained model leads to a negative impact for the disadvantaged group. However, only the ERP-constrained model impact matches the unmitigated model impact for the disadvantaged group while the other four fairness constraints result in a worse impact. We find that the statistically significant results for the disadvantaged group from the Equal distribution match the Benchmark distribution results; the advantaged group results match as before too except for two more significant results from the ERP-constrained DT and EO-constrained GNB models. These results can be seen in Figure 9 in Appendix A.1.

When TPs are weighed twice as heavily as FPs in the Benchmark-Swap distribution (see Figure 4) we see less impact variation amongst the models for the disadvantaged group, with the DP-constrained model as an exception as shown in Figure 10 in Appendix A.1. The statistical significance tests vary more with this setup, except for the disadvantaged group's results for constrained DT, LGR, and GBT models which remain the same and, similar to the Equal distribution results, the EO-constrained GNB model result is significant for the advantaged group. The GNB results are different such that only the ERP and DP results are statistically significant. When ERP constrains all the models, the advantaged group has statistically significant changes to their credit scores.

6.2 Synthetic Dataset Results

We trained an unmitigated DT model and mitigated DT models on our synthetic datasets. For each of our synthetic datasets, we tested the models with two test sets—one that matched the training set for the synthetic dataset and then one that matched the baseline dataset. The latter test set allows us to see how the model trained on

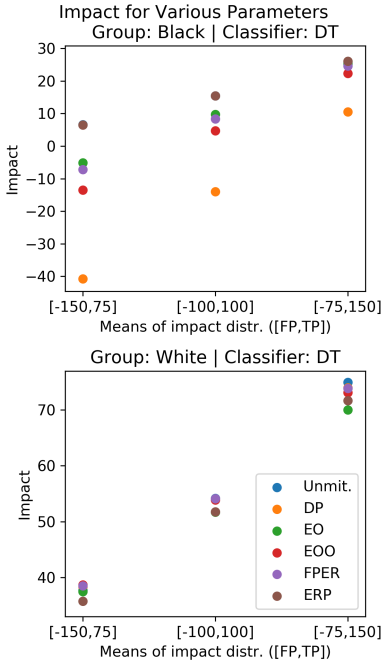

Figure 4: Impact for all fairness constraints for the three non-deterministically generated weight distributions when a DT model was trained on the baseline dataset. Recall that the Benchmark distribution impact results are on the left [-150,75], the Equal distribution impact results are in the middle [-100,100], and the Benchmark-Swap distribution impact results are on the right [-75,100].

synthetic data would work on a test set that matches the real-world. We provide the impact results depending on what test set was used.

No matter what test set was used, the impact of the advantaged group, when having trained models on the synthetic datasets, behaves identically (see bottom plots of Figure 5 and 6). For the rest of this section, we focus on the disadvantaged group's results. For the best impact-performing models for the disadvantaged group, we point to the unmitigated and ERP-constrained models in the top plots of Figures 5 and 6.

As a reminder for our dataset configuration notation (see Section 5.3.2), each scenario is labeled with a two-letter code, where the first letter represents the demographic-ratio and the second letter represents the disadvantaged group's label-ratio. For the values, a "0" says the ratio matches the baseline dataset, "i" says the ratio is imbalanced (but not the same imbalanced ratio as the baseline), and "b" says the ratio is balanced.

6.2.1 Train and Test Sets Have Equal Composition. When increasing the disadvantaged group's representation, we see little effect on the group's impact, see the top plot of Figure 5. When we only change the demographic-ratio and increase it for the disadvantaged group (see scenarios "b0" and "i0" in the top plot of Figure 5), we see an increase in impact for the disadvantaged group (except for the unmitigated and ERP-constrained model results which remain consistent) in comparison to scenario "00" and then all the impact

results converge when the disadvantaged group is in the majority ("i0"). In comparison, in Figure 5, we show that there is less impact variance but the impact improves when the disadvantaged group is more likely to repay (see scenarios "0b," "bb," "ib," "0i," "bi," and "ii") when we compare to the "00" scenario (when they are more likely not to repay the bank).

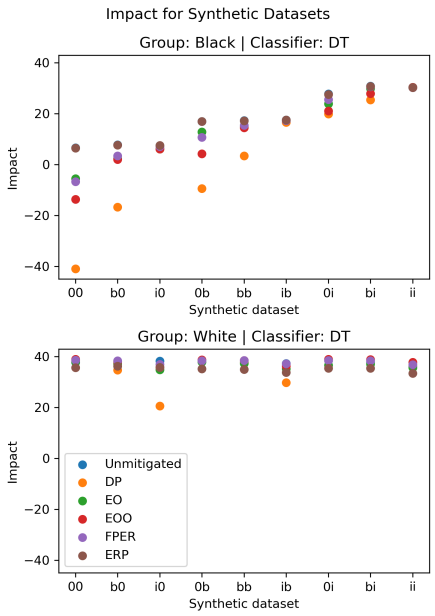

Figure 5: Impact with deterministically generated weights for all synthetic datasets with a test set with equal composition to the training set.

6.2.2 Test Set Matches Baseline Composition. When the disadvantaged group is in the minority (see scenarios "00," "0b," and "0i") in Figure 6, we have the most variance between the impact results. Of the disadvantaged group results in Figure 6, we see the two worst impact results from the DP and EOO-constrained models which align with the worst impact results for the disadvantaged group in Figure 5, when we are not testing with the baseline test set. When increasing the disadvantaged group representation in the synthetic datasets, the impact does increase in comparison to the baseline for EO, EOO, DP, and FPRP-constrained models until it converges (see scenarios "b0," "i0," "bb," "ib," "bi," and "ii") with other model results when the disadvantaged group is in the majority in the top plot of Figure 6. The other two model impact results for the unmitigated and ERP-constrained models show little changes and are consistent as seen in the top plot of Figure 6. Constrastingly to the top plot of Figure 5, where we saw an upward trend for the impact, when we test with the baseline in the top plot of Figure 6, we find the impact stagnating and changing little in comparison to scenario "00."

7 DISCUSSION

Mitigated models can do more harm than good. Our results demonstrate that the bias mitigation methods successfully mitigated unfairness in our loan repayment example. However, the

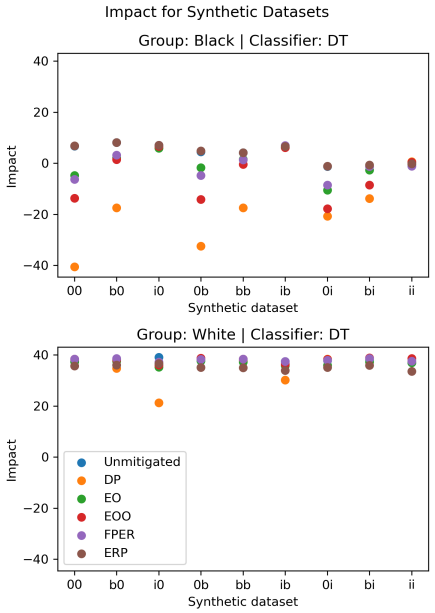

Figure 6: Impact with deterministically generated weights for all synthetic datasets with a test set with the baseline dataset composition.

results also demonstrate a trade-off between optimizing for fairness disparities and impact when choosing a fairness constraint. The problem with this trade-off is that mitigated models sometimes do more harm than unmitigated models and we saw that the disadvantaged group experienced negative impacts the majority of the time when mitigated models were used. We find ERP-constrained and unmitigated models outperforming other models with respect to the disadvantaged group's impact (see top plots of Figure 5 and Figure 6), while the DP-constrained and EOO-constrained models have the lowest impact results.

Balanced datasets may not solve inequalities. Our results also show that the disadvantaged group does not necessarily benefit when a model learns from a synthetic training set that does not match the real-world population's composition. In most cases, they will be treated similarly or worse than if they were being classified by an unmitigated model. When using test sets with the same composition as the training sets, the disadvantaged group tends to see an increase in impacts as they increase their label and demographic-ratios. However, when the test sets match the baseline data (representing the real-world), we see the impacts mostly stagnating or dropping. These results emphasize that imbalanced demographic group and label data should not be assumed to be a problem. We acknowledge though a limitation in our experiments is that the data the models are trained on only includes two features. We leave further impact investigations with datasets that include larger feature sets for future work.

Impact is a key factor not usually accounted for. The weights of the harms and benefits that make up the impact function play a vital role in the impact results and its interpretation. We

argue that impact results can be used to assist practitioners in deciding which fairness constraint to pick. They can decide what the appropriate trade-offs for fairness disparities and impact are (since they can have contradicting best fairness results) when optimizing for their model results. With that said, when impact is a key consideration and certain conditions hold, fairness constraints might not be sufficient. Impact-driven constraints or methods should be developed that consider the weights of different model outcomes and not only the model outcomes like many fairness notions do. Potential future work can also consider how to represent impact when there is not a clear feature (like credit score in our case) that is related to model outcomes.

8 CONCLUSION

In this paper, we assumed that a false positive model outcome has a negative impact and investigated if, in that case, mitigated models benefit the disadvantaged group or further harm them. To explore this, we used the loan repayment example and tested how fairness constraints and dataset composition affect the impacts on demographic groups. Our experiments, in the case of our loan repayment example under certain conditions, showcased that impact was worsened for the disadvantaged group the majority of the time when testing supposedly "fair" models. We highlight though that impact highly depends upon the context. Our key finding is that there is a trade-off between fairness constraints and impact.

We argue that including notions of impact while testing mitigated models before they are deployed is crucial. Testing these models with those impacts can aid practitioners in choosing the fairness constraint that matters most for their use case. Lastly, we emphasize that decreases in fairness disparity metric values in mitigated models do not necessarily equate to decreases in negative impacts on the disadvantaged group.

ACKNOWLEDGMENTS

This work was supported by UK Research and Innovation [grant number EP/S023356/1] in the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence. The second author was funded by the DAAD RISE Worldwide 2022 program to complete research with the first author over summer 2022 in London. We would like to thank the reviewers for their constructive feedback and Maria Stoica and Julia Barnett for their helpful feedback.

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. 2018. A Reductions Approach to Fair Classification. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 60–69.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning*. fairmlbook.org. http://www.fairmlbook.org.

[3] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilović, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang. 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development* 63, 4/5 (2019), 4:1–4:15.

[4] Ruha Benjamin. 2019. *Race After Technology: Abolitionist Tools for the New Jim Code*. Polity, Medford, MA.

[5] Sarah Bird, Miroslav Dudík, Hanna Wallach, and Kathleen Walker. 2020. *Fairlearn: A toolkit for assessing and improving fairness in AI*. Technical Report. Microsoft. 6 pages.

[6] Miranda Bogen and Aaron Rieke. 2018. *Help Wanted: An Examination of Hiring Algorithms, Equity, and Bias*. Technical Report.

[7] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research, Vol. 81)*, Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, 77–91.

[8] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building Classifiers with Independency Constraints. In *2009 IEEE International Conference on Data Mining Workshops*. 13–18.

[9] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163.

[10] Kate Crawford. 2017. The Trouble with Bias. NIPS Keynote.

[11] Natalia Criado, Xavier Ferrer, and Jose Such. 2021. Attesting Digital Discrimination Using Norms. *International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI)* 6, 5 (2021), 16–23.

[12] Natalia Criado and Jose Such. 2019. Digital Discrimination. In *Algorithmic Regulation*. Oxford University Press.

[13] Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, D. Sculley, and Yoni Halpern. 2020. Fairness is Not Static: Deeper Understanding of Long Term Fairness via Simulation Studies *(FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 525–534.

[14] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. 214–226.

[15] Xavier Ferrer, Tom van Nuenen, Jose Such, Mark Cote, and Natalia Criado. 2021. Bias and Discrimination in AI: a cross-disciplinary perspective. *IEEE Technology and Society* 20, 2 (2021), 72–80.

[16] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2021. The (Im)Possibility of Fairness: Different Value Systems Require Different Mechanisms for Fair Decision Making. *Commun. ACM* 64, 4 (2021), 136–143.

[17] Sara Hajian and Josep Domingo-Ferrer. 2013. A Methodology for Direct and Indirect Discrimination Prevention in Data Mining. *IEEE Transactions on Knowledge and Data Engineering* 25, 7 (2013), 1445–1459.

[18] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc.

[19] Deborah Hellman. 2020. Measuring Algorithmic Fairness. *Virginia Law Review* 106 (2020).

[20] Lily Hu and Yiling Chen. 2018. A Short-Term Intervention for Long-Term Fairness in the Labor Market. In *Proceedings of the 2018 World Wide Web Conference* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1389–1398.

[21] Ben Hutchinson and Margaret Mitchell. 2019. 50 Years of Test (Un)fairness: Lessons for Machine Learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 49–58.

[22] Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. 2010. Discrimination Aware Decision Tree Learning. In *2010 IEEE International Conference on Data Mining*. IEEE, 869–874.

[23] Faisal Kamiran, Indrė Žliobaitė, and Toon Calders. 2013. Quantifying Explainable Discrimination and Removing Illegal Discrimination in Automated Decision Making. *Knowledge and information systems* 35, 3 (2013), 613–644.

[24] Matt Kusner, Chris Russell, Joshua Loftus, and Ricardo Silva. 2019. Making Decisions that Reduce Discriminatory Impacts. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 3591–3600.

[25] David Lindner, Hoda Heidari, and Andreas Krause. 2021. Addressing the Long-term Impact of ML Decisions via Policy Regret. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 537–544. Main Track.

[26] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. 2018. Delayed Impact of Fair Machine Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm, Sweden, 3150–3158.

[27] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* 54, 6, Article 115 (2021), 35 pages.

[28] Jacob Metcalf, Emanuel Moss, et al. 2019. Owning ethics: Corporate logics, silicon valley, and the institutionalization of ethics. *Social Research: An International Quarterly* 86, 2 (2019), 449–476.

[29] Arvind Narayanan. 2018. 21 Fairness Definitions and Their Politics. FAT* 2018 Tutorial.

[30] Ziad Obermeyer and Sendhil Mullainathan. 2019. Dissecting Racial Bias in an Algorithm That Guides Health Decisions for 70 Million People. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA)

(FAT '19)*. Association for Computing Machinery, New York, NY, USA, 89.

[31] Cathy O'Neil. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Crown Publishing Group, USA.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[33] Charan Reddy, Deepak Sharma, Soroush Mehri, Adriana Romero Soriano, Samira Shabanian, and Sina Honari. 2021. Benchmarking Bias Mitigation Algorithms in Representation Learning through Fairness Metrics. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, J. Vanschoren and S. Yeung (Eds.), Vol. 1.

[34] Willy E. Rice. 1996. Race, Gender, "Redlining," and the Discriminatory Access to Loans, Credit, and Insurance: An Historical and Empirical Analysis of Consumers Who Sued Lenders and Insurers in Federal and State Courts, 1950-1995. *San Diego Law Review* 33 (1996).

[35] Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. 2019. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 59–68.

[36] Jose Such. 2017. Privacy and Autonomous Systems. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 4761–4767.

[37] Latanya Sweeney. 2013. Discrimination in Online Ad Delivery: Google Ads, Black Names and White Names, Racial Discrimination, and Click Advertising. *Queue* 11, 3 (Mar 2013), 10–29.

[38] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. 2021. DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks. In *Conference on Neural Information Processing Systems(NeurIPS) 2021.*

[39] Tom van Nuenen, Xavier Ferrer, Jose Such, and Mark Cote. 2020. Transparency for Whom? Assessing Discriminatory Artificial Intelligence. *IEEE Computer* 53 (2020), 36–44.

[40] Tom van Nuenen, Jose Such, and Mark Cote. 2022. Intersectional Experiences of Unfair Treatment Caused by Automated Computational Systems. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (2022), 1–30.

[41] Sahil Verma and Julia Rubin. 2018. Fairness Definitions Explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*. IEEE, 1–7.

[42] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2021. Bias Preservation in Machine Learning: The Legality of Fairness Metrics Under EU Non-Discrimination Law. *West Virginia Law Review* 123, 3 (2021).

[43] Depeng Xu, Yongkai Wu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2019. Achieving Causal Fairness through Generative Adversarial Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19.* International Joint Conferences on Artificial Intelligence Organization, 1452–1458.

[44] Depeng Xu, Shuhan Yuan, Lu Zhang, and Xintao Wu. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. In *2018 IEEE International Conference on Big Data (Big Data)*. 570–575.

[45] Muhammad Bilal Zafar. 2019. *Discrimination in Algorithmic Decision Making: From Principles to Measures and Mechanisms.* Ph. D. Dissertation.

A EXTENDED REASONING AND RESULTS

A.1 Credit Score Change Statistical Significance Results

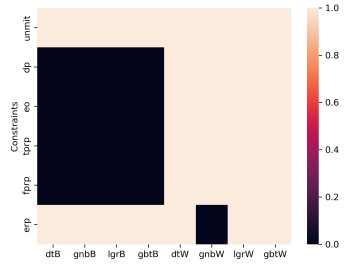

Figure 7: The credit score distribution statistical significance results when using deterministically generated weights.

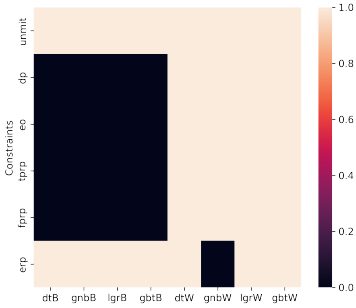

Figure 8: The credit score distribution statistical significance results when using the Benchmark distribution for non-deterministically generated weights.

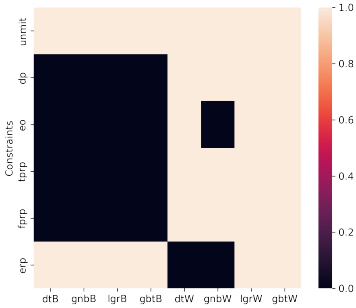

Figure 9: The credit score distribution statistical significance results when using the Equal distribution for non-deterministically generated weights.

We include the statistical significance results for the updated credit score distributions as a result of weights generated for impact from mitigated models in comparison to unmitigated models

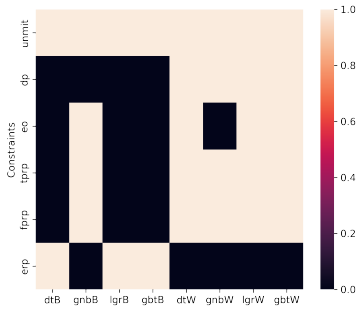

Figure 10: The credit score distribution statistical significance results when using the Benchmark-Swap distribution for non-deterministically generated weights.

("unmit" as referred to in the Figures). In each of the Figures, they showcase the results of the MWU-Tests, which compare credit score distributions from each mitigated model with the unmitigated score distribution for each model and by protected attribute. We note that in the Figures TPRP (True Positive Rate Parity) refers to Equality of Opportunity (EOO). The "B" or "W" added to the ML model acronym on the x-axis represents if it was a Black or White group distribution. The y-axis shows the fairness constraint used. If the field is dark in color, the result of the MWU-test is significant ($p < 0.05$), which means that the credit score distributions tested are significantly different. We give the results for all ML models when we have deterministically generated weights for impact in Figure 7 and when we have non-deterministically generated weights in Figures 8, 9, and 10. Deeper analysis of these results are in the main body of the paper.

A.2 Model Performance and Reduction Algorithm Results

Table 6 shows the model performance of our ML models. The accuracy of the unmitigated model (without any fairness constraint) trained with the GNB classifier is lower than the accuracy of the other three unmitigated models, which all have an accuracy of 88%. The mitigated models all have relatively similar accuracies.

Table 6: Model accuracy (in %) for all classifiers (by row) when trained on the baseline dataset. The column, "Unmit," shows the results of the unmitigated models and the columns to the right of that column specifies the fairness constraints applied to the mitigated models with the Exponentiated Gradient reduction algorithm for those results.

Classifier	Unmit	DP	EO	EOO	FPRP	ERP
DT	88.18	84.66	85.36	86.59	87.41	85.29
GNB	85.67	81.49	83.96	86.24	87.16	84.92
LGR	88.23	84.66	84.54	86.44	87.42	84.95
GBT	88.22	84.70	85.16	86.58	87.45	85.42

In our paper, we chose to use Exponentiated Gradient for our extended experiments with the synthetic datasets as our reduction

Table 7: The values of the fairness disparity metrics for our mitigated models (with Grid Search applied using the fairness constraints along the columns) for all four classifiers (rows) when trained on the baseline dataset.

Classifier	DP	EO	EOO	FPRP	ERP
DT	28.17	22.79	7.46	20.82	5.26
GNB	82.42	44.22	0.63	38.21	1.27
LGR	27.84	24.81	5.53	22.02	5.21
GBT	28.17	21.54	5.9	19.28	5.32

algorithm over the Grid Search algorithm. We present the Grid Search fairness disparity metric results with the baseline dataset to showcase why Exponentiated Gradient was the stronger algorithm for mitigating unfairness. We include Table 7 that covers the fairness disparity results after Grid Search mitigated unfairness in our different ML models. If compared with Table 5, we clearly see that Exponentiated Gradient outperforms Grid Search when dropping the fairness disparity metric values in comparison to the unmitigated model results in Table 4.

A.3 Dataset Generation Algorithms

The algorithms we used for generating our datasets are below. Algorithm 1 generates a tabular dataset from the original loan repayment dataset, depending on demographic ratio and order of magnitude. In the algorithms, when "concat" is used, we refer to the method concatenate which happens by row ("row") or by column ("col") and combines arrays into one array. Algorithm 2 generates the subset of data with the chosen ratios (demographic-ratio and label-ratio) that we vary and Algorithm 3 is the overall sampling loop connecting Algorithm 1 and 2, ensuring that we generate a dataset with the desired ratios and size.

The time complexity and space complexity of our algorithms are $O(n)$, highlighting that as the dataset size increases so does the running time and storage space. In our algorithms, we focus on the generation of one key feature for X, but the algorithms could, potentially, be used to sample more features; also, other features could be generated separately. With these algorithms, we assume we have access to a true label distribution and a feature distribution for a key feature. However, this might not be the case.

Algorithm 1 Create baseline dataset

Require: $f_1(x) \leftarrow P(X \leq x)$ {Cumulative distribution function for items},
$f_2(x) \leftarrow P(X = x)$ {Probability mass function for the label likelihoods},
oom {Order of magnitude for dataset creation},
$(r_0, r_1) \leftarrow R$ {Demographic group ratio},
$choices(items, probabilities, samples_{num})$ {Function for generating samples},
$randint(start, stop)$ {Function for generating random numbers}

$samples\_num_0 \leftarrow oom \times r_0$ {Initialize variables for number of samples by group}
$samples\_num_1 \leftarrow oom \times r_1$
$items_0 \leftarrow f_1.values_0$ {Collect the values from the CDF functions for each group}
$items_1 \leftarrow f_1.values_1$
$probs_0 \leftarrow f_2(items_0)$ {Collect the probabilities from the PMF functions for each group}
$probs_1 \leftarrow f_2(items_1)$
$samples_0 \leftarrow choices(items_0, probs_0, samples\_num_0)$ {Generate the samples for the groups}
$samples_1 \leftarrow choices(items_1, probs_1, samples\_num_1)$
$samples_{disadv} \leftarrow array([0] \times samples\_num_0)$
$samples_{adv} \leftarrow array([1] \times samples\_num_1)$
$D \leftarrow shuffle([concat_{col}[concat_{row}samples_0 \& samples_1]$ {Combine the arrays}
$\&[concat_{row}samples_{disadv} \& samples_{adv}]$ & $[concat_{row}probs_0 \& probs_1]])$
$labels \leftarrow []$, $index \leftarrow 0$ {Initialize array for labels and integer variable for index}
for $index < |D|$ **do**
  $rand\_num \leftarrow randint(0, 1000)/10$ {Initialize a random integer variable}
  if $rand\_num > D[index][2]$ **then**
    $label.append(0)$ {If true, assign negative class label}
  else
    $labels.append(1)$ {Else, assign a positive class label}
  end if
end for
$D \leftarrow concat_{col}[D \& labels]$, $D \leftarrow D.remove_{col}(2)$ {Add labels to D and drop probabilities}
return D

Algorithm 2 Generate subset with defined ratios

Require: D {Whole data set $[x_{group,label} \in D$ denote a sample with a $group, label \in (0, 1)]$},
$|S|$ {Size of our desired synthetic subset $S \subseteq D$}, R, L_0, L_1

for $group \in (0, 1)$ and $label \in (0, 1)$ **do**
  $|S_{group,label}| = d_{group} * l_{group,label} * |S|$ {Compute the number of samples}
end for
for $group \in (0, 1)$ and $label \in (0, 1)$ **do**
  if $|S_{group,label}| < |D_{group,label}|$ **then**
    $|S_{new}| = |D_{group,label}|/(d_{group} * l_{group,label})$ {Adjust the set size}
    for $group \in (0, 1)$ and $label \in (0, 1)$ **do**
      $|S_{group,label}| = d_{group} * l_{group,label} * |S_{new}|$ {Adjust the number of samples}
    end for
  end if
end for
for $group \in (0, 1)$ and $label \in (0, 1)$ **do**
  $S_{group,label} \in S = \{x_{group,label} \in D; |S_{group,label}|\}$ {Sample the desired amount of $x \in D$}
end for
return S

Algorithm 3 Generation of subset loop

Require: $|S|_{desired}, |D|, R, L_0, L_1, P(X \leq x)$ {Cumulative distribution function for items}, $P(X = x)$ {Probability mass function for the label likelihoods}

$D \leftarrow createBaselineSyntheticSet(P(X \leq x), P(X = x), |D|, R)$ {Algorithm 1}
$S \leftarrow generatesSubsetWithDefinedRatios(D, |S|_{desired}, R, L_0, L_1)$ {Algorithm 2}
while $|S| < |S|_{desired}$ **do**
  $D_{new} \leftarrow createBaselineSyntheticSet(P(X \leq x), P(X = x), |D|, R)$ {Algorithm 1}
  $D \leftarrow concat_{row}[D\&D_{new}]$
  $S \leftarrow generatesSubsetWithDefinedRatios(D, |S|_{desired}, R, L_0, L_1)$ {Algorithm 2}
end while
return S