Delayed Impact Optimization: Mitigating Bias in Deep Learning

Master's Thesis

Submitted by:
Tim Kessel (604307)
for acquiring the degree of

Master of Science (M.Sc.)
in Business Administration

at the School of Business and Economics
at Humboldt University Berlin

Submitted to:
Prof. Dr. Stefan Lessmann
Chair of Information Systems

Berlin, 04.07.2022

Abstract

Well-trained machine learning models are prone to reproducing unwanted bias present in the training data. Established fairness processors aim at mitigating this bias by modifying the predictions to satisfy certain fairness criteria. These criteria treat fairness as a static condition and hence, many fairness processors produce fair predictions regarding the decision process itself but do not necessarily increase the long-term well-being in dynamic scenarios like credit scoring. This paper presents a novel technique designed to directly optimize the long-term fairness in these scenarios based on the concept of delayed impact proposed by Liu et al. (2018) by dynamically maximizing the delayed impact in the unprivileged of two sensitive groups. It then investigates the effect of this processor on other fairness criteria, model performance, and the estimated profit for different hyperparameters. The method is theoretically applicable to different types of predictive models and when applied to simple neural networks trained on simulated and real-world data, it demonstrates the ability to significantly reduce unwanted bias in the predictions. However, when compared to another fairness processor, namely adversarial debiasing, it falls behind in terms of both cost-effectiveness and consistency of the results.

Contents

1 Introduction **1**

2 Theoretical Background **3**

2.1 Fairness in Machine Learning 3

2.2 The Profit-Fairness Trade-Off 4

2.3 Fairness Criteria 5

2.3.1 Independence - Disparate Impact 5

2.3.2 Separation - Equalized Odds 6

2.3.3 Sufficiency - Predictive Parity 7

2.4 Delayed Impact 7

2.5 Fairness Processors 10

2.5.1 Types of Fairness Processors 10

2.5.2 Adversarial Debiasing 10

3 Delayed Impact Optimization **11**

3.1 Rationale 11

3.2 Implementation of the Penalty 12

3.3 Alternative Implementations of the Loss Function 14

4 Methodology **15**

4.1 Metrics 15

4.2 Models 17

4.3 Data 18

5 Results & Discussion **19**

5.1 Performance and Fairness Analysis 19

5.1.1 Delayed Impact Optimization - Simulated Data 19

5.1.2 Delayed Impact Optimization - Give Me Some Credit 22

5.1.3 Adversarial Debiasing - Simulated Data 25

5.1.4 Adversarial Debiasing - Give Me Some Credit 27

5.2 Pareto Frontiers 30

5.2.1 Delayed Impact Optimization vs. Adversarial Debiasing (Simulated Data) 30

5.2.2 Delayed Impact Optimization vs. Adversarial Debiasing (Give Me Some Credit) 32

6 Conclusion **33**

References **37**

Appendices **38**
A Alternative Loss Functions 38
B Pareto Frontiers 40

Declaration of Academic Honesty **42**

1 Introduction

Machine learning enables us to do things that would have been unthinkable a few decades ago. We effortlessly translate whole documents in seconds, our cars can drive themselves, and witnessing a three-year-old interact with Amazon's Alexa must exceed George Orwell's worst nightmares. Achieving these capabilities requires vast amounts of training data. Natural language processing would not be possible without millions of human-written pages collected from the internet, Tesla gathers the data of around five million human-driven kilometers per day (Musk, 2016) and people interacting with more than 100 million Alexa devices train Amazon's algorithms every day (Al-Heeti, 2019). What all these people write, do, and say greatly influences what the translators, cars, and smart home devices will write, do and say. And whatever patterns exist in the training data, they are likely to show up in the outcome, too. While this is generally desired and necessary, it also leads to unintended outcomes whenever the training data is biased in an undesirable way, which happens more frequently than one might think.

'Tay' was a chatbot released by Microsoft in 2016 to interact with users on Twitter. Only 16 hours after its initial release, it had to be taken down due to racist, misanthropic, and nazi-sympathizing behavior that it had learned from interactions with human chat partners (Wakefield, 2016). This is one of the more striking but less consequential examples of biased machine learning. Unfortunately, other instances of biased machine learning can have potentially life-changing consequences, like 'COMPAS', a tool used by US courts to assess the probability of a defendant becoming a recidivist. Research showed that black defendants were almost twice as likely to be classified as high risk after not re-offending for two years than white defendants and that black individuals were mistakenly labeled as high risk when they actually did not re-offend twice as often as their white counterparts (Larson et al., 2016). This can make the difference between having to live in prison for years and being released on probation, continuing to live a relatively normal life.

Whenever influential decisions are made or supported by machine learning algorithms, bias can be extremely harmful. Thus, developing and refining techniques that can remove or at least decrease it is a necessity that has been disregarded for too long. Only a few years ago, algorithmic fairness used to be a niche topic, that only some scientific papers spent effort on. But with fairness and equality moving more into focus in everyday life and algorithmic decision-making becoming more and more ubiquitous, the focus in the field of computer science started to shift, as can be seen in Figure 1.

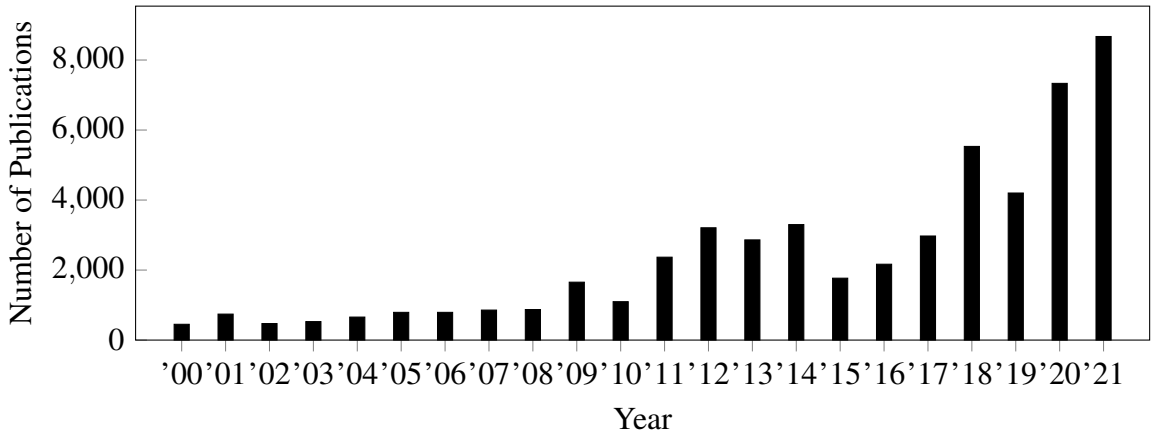

Figure 1: Articles on Algorithmic Fairness - Data retrieved from "Dimensions AI" (2022)

The topic of fair decision-making itself is not new, as it has already been studied more than half a century ago in contexts like educational testing (Darlington, 1971), but with ever more sophisticated models being built and machine learning becoming more readily available, the need for methods that achieve not only accurate but also fair results continues to increase.

One of the more consequential decisions, already handed over to machine learning, takes place in the lending industry. Whether or not a person receives a loan is no longer a decision made by an employee. Instead, an algorithm predicts the probability of customers defaulting based on their idiosyncratic characteristics, and only those whose probability falls under a certain threshold, i.e., are labeled as low risk, receive money. And while not receiving money when you need it can be a problem, receiving money and not being able to repay it can be even more harmful, as it leads to a worse credit rating and even more financial distress in the long run. Liu et al. (2018) analyze this issue with what they call the delayed impact, which they define as the change in credit scores following predictions made by different fairness-controlled models and investigate differences between privileged and unprivileged groups. They show that fair decisions, governed to satisfy different fairness criteria, do not necessarily lead to fair long-term outcomes, which are intuitively at least as important as fair decisions. Until now, to the best knowledge of the author, there have been no efforts to directly optimize the delayed impact of lending decisions which leaves a gap that this work attempts to close. As we will see below in more detail, it is in the best interest of lending institutions to not only invest in more accurate models but to also find models that can achieve the fairest predictions at the lowest possible cost.

The goal of this work is to develop and test a new method for achieving fair results that is based on the concept of delayed impact, which extends the view of fairness in credit scoring from one point in time (fair decision) to a more long-term perspective (fair outcomes). In summary, the research question this paper strives to answer is whether a penalty that punishes low and rewards high delayed impact that is added to the loss function for the unprivileged of two groups can lead to an increase in fairness and if so, how it affects profit and other fairness metrics.

This thesis is structured as follows: Section 2 gives an overview of fairness in machine learning and places it into the context of credit scoring, offers different fairness definitions, and presents how unwanted bias can be mitigated through the use of fairness processors. Thereafter, Section 3 explicates the idea and implementation of delayed impact optimization and shows alternative loss functions that have been discarded in the development stage. Next, the subsequent experiments, the used data, and calculated metrics are explained in Section 4, before the results are depicted and analyzed in Section 5. Finally, Section 6 summarizes the findings, reiterates any shortcomings, and gives an outlook on potential future research on the topic.

2 Theoretical Background

This section outlines the concept of fairness in machine learning and its implications in the context of credit scoring, examines different fairness definitions, and presents established approaches to mitigating unwanted bias in predictive modeling.

2.1 Fairness in Machine Learning

Bias in machine learning can be found in almost any application from word embedding (Bolukbasi et al., 2016; Caliskan et al., 2017) to predictive policing (Lum & Isaac, 2016), and typically, it represents bias already present in the training data (Chouldechova & Roth, 2018). If we don't want a model to treat people differently depending on their gender, ethnic group, or age, then why don't we just remove this information from the training data? This is a reasonable question that might arise when thinking about fair machine learning. Unfortunately, it showed that this is ineffective since especially in large feature spaces, protected attributes are usually redundant and if they are removed, the model will find a redundant encoding of these attributes based on the remaining features (Barocas et al., 2019).

In the fairness literature, these protected features are often referred to as sensitive attributes. The groups that can be separated based on these attributes are therefore referred to as sensitive groups. In this paper, the sensitive group which benefits from the bias is also called the privileged group, in contrast to the unprivileged group that is put at a disadvantage. These terms are not meant to imply intent, i.e., someone deliberately treating groups differently, but latent bias that stems from the training data and is present in the predictions. Note, that this terminology refers to the outcome of an unconstrained model with the original data. It is possible that the originally unprivileged group becomes privileged through the use of fairness-increasing methods and vice versa. However, thinking of a privileged and an unprivileged group appears to be more intuitively understandable and is thus deliberately used in all text throughout this paper instead of the more mathematical depiction of sensitive groups as $x_{s=0}$ and $x_{s=1}$. Since descriptions like "positive outcome" are often ambiguous, meaning for example either the binary feature for "default" being equal to 1, or on the contrary, the desirable outcome of not

defaulting, it is important to clarify this terminology. In this paper, outcomes referred to as positive will always imply that the target variable equals 1. With all target variables in this paper signifying default or late payment, a positive outcome always signals the inability to repay a loan in time. When it comes to the individual, the terms "desirable" and "undesirable" will be used commonly: it is desirable for an individual to repay a loan in time and it is undesirable to default. The same applies to the lending institution. Furthermore, the terms target and (true) label are used interchangeably when referring to the dependent variable.

2.2 The Profit-Fairness Trade-Off

The main goal of any for-profit business is to maximize profit. This statement is inherently logical and agreed upon so widely by both economists and business professionals that it can be considered an axiom and has been a fundamental assumption since the beginnings of modern economic theory (De Scitovszky, 1943; Smith, 1776). At the same time, maximizing fairness in machine learning tasks with any of the established fairness processors available today comes at the cost of a more or less pronounced decrease in predictive accuracy (Friedler et al., 2019) and consequently a decrease in profit. Whilst this trade-off appears to not be inevitable at least in theory (Dutta et al., 2020), it generally still makes restricted models with fairness optimization undesirable for profit-oriented businesses compared to their unconstrained counterparts. Thus, the question arises, why a company would be interested in the development of new fairness processors. The answer to this question lies in regulation. Especially in fields, where machine learning can yield large benefits for institutions or society but comes with far-reaching consequences for the individual, like medicine (Rajkomar et al., 2018) or criminal risk assessment (Larson et al., 2016), regulation becoming stricter clearly is a possibility and since the economic situation of an individual has a significant influence on its overall well-being (Ferrer-i-Carbonell, 2005), the same is to be said for the financial industry. The basis for specific regulation in the European Union is already established in Article 13 of the “Treaty establishing the European Community” (2002) that enables combating discrimination, and “Council Directive 2000/43/EC” (2000), which states that "specific action in the field of discrimination [...] should [...] cover [...] access to and supply of goods and services". These minimum standards in EU law form a lower limit that the member states cannot go below, however, they are free to implement stricter national laws. Therefore, it is in the best interest of financial institutions to have methods at hand as early as possible that can fulfill possible fairness requirements while at the same time keeping the profit reduction as small as possible. And at least in theory, the definition of fairness as equal delayed impact fuels hopes for a less pronounced profit-fairness trade-off since it appears to be well aligned with the objectives of lending institutions: default events do not only worsen the economic situation of the individual, they also decrease the institution’s profit, whereas successful and timely repayment benefit both customer and institution. Thus, maximizing the delayed impact does not automatically contradict maximizing profits.

2.3 Fairness Criteria

A plethora of mathematical definitions for fairness have been proposed, many of which are similar to each other or can be derived from one another. These so-called fairness criteria generally fall into one of three groups: group fairness, individual fairness, and counterfactual fairness. All fairness criteria used in this work belong to the former group. The three established criteria - independence, separation, and sufficiency - are chosen to be used in this work, because they relate to or combine the requirements of many other criteria that have been proposed (Barocas et al., 2019).

2.3.1 Independence - Disparate Impact

For the criterion of independence to be satisfied, the sensitive attribute needs to be statistically independent of the prediction. If we define $y = 0$ and $y = 1$ as the desirable and undesirable outcome and $x_s = 0$ and $x_s = 1$ as the unprivileged and privileged class of the sensitive attribute, this requirement is satisfied if the following statement holds:

$$Pr(y = 0|x_s = 0) = Pr(y = 0|x_s = 1) \tag{1}$$

To measure the degree to which this criterion is satisfied, we can calculate the ratio of both sides of the statement, which is called disparate impact and can be written as:

$$\text{Disparate Impact} = \frac{Pr(y = 0|x_s = 0)}{Pr(y = 0|x_s = 1)} \tag{2}$$

Going back to the US Supreme Court ruling of *Griggs v. Duke Power Co.* (1975), which ruled a hiring decision illegal if it results in disparate impact for different ethnic groups, it can be interpreted as measuring possibly unintended discriminatory outcomes, not to be confused with intended discriminatory practices. While the US Supreme Court explicitly does not provide a formula for disparate impact analysis (*Watson v. Fort Worth Bank & Trust*, 1988) and only specifies it as "a racial pattern significantly different from that of the pool of applicants" (*Albemarle Paper Co. v. Moody*, 1975), the US Equal Employment Opportunity Commission continued to define it as "a substantially different rate of selection in hiring, promotion or other employment decision which works to the disadvantage of members of a race, sex or ethnic group" and proposed the "80%" rule of thumb, stating that disparate impact of less than 80% can be seen as problematic.

When referring to probability predictions and a decision based on a cutoff probability τ, disparate impact can also be written as:

$$\text{Disparate Impact} = \frac{Pr(\hat{y} < \tau|x_s = 0)}{Pr(\hat{y} < \tau|x_s = 1)} \tag{3}$$

This metric, although originally brought forward in the context of hiring decisions, can also be applied to other decision-making processes like credit scoring. Since for the independence criterion that is measured through delayed impact to be fulfilled, prediction and sensitive attribute need to be entirely independent, disparate impact is one of the stricter metrics used in this paper. Other terms for the same principle include demographic parity and statistical parity.

2.3.2 Separation - Equalized Odds

For the separation criterion to be satisfied, the predictions need to be conditionally independent of the sensitive attribute, given the target value. In other words, the true positive rate and the false positive rate need to be the same for both groups (Barocas et al., 2019). Formally, this applies if the following two equations hold:

$$Pr(\hat{y} > \tau | y = 0, x_s = 0) = Pr(\hat{y} > \tau | y = 0, x_s = 1) \tag{4}$$

$$Pr(\hat{y} > \tau | y = 1, x_s = 0) = Pr(\hat{y} > \tau | y = 1, x_s = 1) \tag{5}$$

Compared to the disparate impact measure, separation is easier to satisfy, as the required independence of prediction and sensitive attribute is conditional given the target. It can be argued that this is more realistic in a credit scoring context because the sensitivity will often be correlated with the target variable. For example, even if socially undesirable, the average income of an unprivileged group might actually be lower than that of a privileged group, leading to an actual higher default rate in this group. The separation criterion acknowledges this bias, whereas independence and hence the disparate impact metric do not. To measure the degree to which separation is satisfied, a criterion suggested by Kozodoi et al. (2022) is used that computes the average absolute difference between the group-wise false positive rate (FPR) and true positive rate (TPR), with the distinction that instead of the absolute difference, the signed value is used. The metric is thus defined as follows:

$$\text{Separation} = \frac{1}{2}((\text{FPR}_{x_s=0} - \text{FPR}_{x_s=1}) + (\text{TPR}_{x_s=0} - \text{TPR}_{x_s=1})) \tag{6}$$

Note, that for ease of understanding, the calculated metrics are named identically to the criteria they are based on. However, the metrics measure to which degree the criteria are satisfied, and they are not identical to them.

If the criterion of separation was satisfied, the average odds difference would be 0, due to equalized group-wise FPR and TPR. Using or not using the absolute difference is an arbitrary decision, as it only changes the sign of negative outcomes. Not using the absolute difference gives us slightly more information, as we can theoretically also interpret the sign. However, since $TPR = 1 - FNR$, where FNR is the false negative rate, the metric could be rewritten as:

$$\text{Separation} = \frac{1}{2}\left(\text{FPR}_{x_s=0} - \text{FPR}_{x_s=1}\right) + \left(\left(1 - \text{FNR}_{x_s=0}\right) - \left(1 - \text{FNR}_{x_s=1}\right)\right) \tag{7}$$

Thus, values above zero indicate that the FPR in the unprivileged group is higher than in the privileged group or that the FNR in the unprivileged group is lower than in the privileged group. These opposing responses make it hard to interpret the sign of the metric in terms of which of the groups is at a disadvantage. Using the group-wise difference of the FPR and FNR instead would be equally appropriate to show the degree to which the criterion of separation is satisfied, while at the same time making the sign interpretable, as values above zero would imply that both error rates are higher in the unprivileged group, while values below zero would imply that they are lower in the unprivileged group. This could then be interpreted as unfair treatment of the previously privileged group.

2.3.3 Sufficiency - Predictive Parity

Barocas et al. (2019) define predictions as sufficient, if the probability of correctly classifying a positive outcome, also referred to as the sensitivity, is the same for both sensitive groups. Therefore, following the notation of this paper, sufficiency is satisfied if the following equation holds true:

$$Pr(y = 1|\hat{y} > \tau, x_s = 0) = Pr(y = 1|\hat{y} > \tau, x_s = 1) \tag{8}$$

The further away the outcome is from this balance, the less fair the model is concerning this criterion. The metric to measure the degree to which sufficiency is satisfied is therefore defined as the difference in the sensitivity for both groups as proposed by Kozodoi et al. (2022):

$$\text{Sufficiency} = Pr(y = 1|\hat{y} > \tau, x_s = 0) - Pr(y = 1|\hat{y} > \tau, x_s = 1) \tag{9}$$

It is also worth noting that independence and sufficiency are mutually exclusive if the target variable and the sensitive attribute are not independent. If bias is present in the data this is usually the case, which is why the proposition of mutual exclusivity can be considered to be true for all following experiments.

2.4 Delayed Impact

As Liu et al. (2018) rightfully state, machine learning is commonly utilized in rather static settings. Often, the data represents only one point in time and the objectives are static as well. The consequences of the decisions made in the process, on the other hand, are far from static. In the context of credit scoring, for example, the decision for or against granting a loan to a customer only marks the beginning of a potentially significant change in both short- and long-term well-being. Think of a customer that applied for a loan. In case the lending institution

rejects said application, the customer's situation remains unchanged. The individual might try again with another institution and their creditworthiness has been assessed but not influenced. If, in contrast, a loan is granted, the implications for the customer can be significantly more severe. Two basic scenarios can arise from this situation: either, the customer fully repays the loan in time, in which case the personal credit score increases, or some form of default or late payment happens, in which case the credit score decreases.

Most common fairness criteria disregard these implications and define fairness very narrowly: if someone applies for a loan, the desired outcome is to receive the loan. If the chances of receiving the loan are equally distributed among different groups, the decision-making process is fair. But that is only part of the truth. Most definitions focus on the equality of outcomes, which is by no means wrong, but it has been debated for decades whether equality of outcomes or equality of opportunities is what one should endeavor to achieve (Roemer, 1993; Roemer & Trannoy, 2016). Answering the philosophical debate of whether we should desire everybody to receive the same outcome or if we want what is often referred to as a level playing field, exceeds the scope of this work. But if a scenario in which the difference in the economic situation of two sensitive groups increases over time can still be evaluated as fair, just because their loan applications were accepted and rejected in a similar manner, that at least raises questions about the scope of standard fairness criteria.

It is in part for this reason, that Liu et al. (2018) propose a method that evaluates fairness metrics by looking into the subsequent changes following the credit decision in a one-step feedback model. Their method originally is not meant to be a way of measuring fairness by itself but instead to evaluate the long-term effects of different fairness criteria. To do that, they introduce what they call an "outcome curve". This curve plots the changes in average group credit scores ($\Delta\mu$) against the possible acceptance rates between zero and one as can be seen in Figure 2.

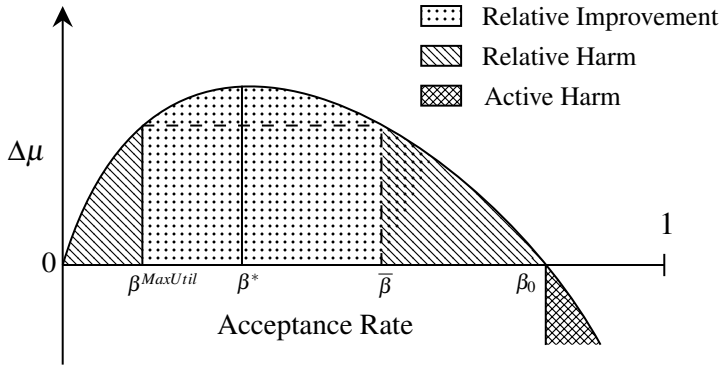

Figure 2: Outcome Curve - Adapted from Liu et al. (2018)

On the horizontal axis, different values are visible: $\beta^{MaxUtil}$ marks the acceptance rate that maximizes the institution's utility and would be chosen by it in an unconstrained optimization scenario. β^* maximizes the mean change in credit scores. Acceptance rate $\bar{\beta}$ leads to the same change in average credit scores as $\beta^{MaxUtil}$ and β_0 depicts an acceptance rate at which the mean scores remain unchanged. Note, that this is just an illustrative example to show all possible developments. Outcome curves can also be strictly positive or negative and don't necessarily match this depiction. The patterns indicate three possible developments: relative harm, i.e., the av-

erage credit score increases, but less so than in the case of a utility-maximizing acceptance rate, relative improvement, i.e., the average credit score increases more than in the case of a utility-maximizing selection rate, and active harm, i.e., the average credit score decreases.

The idea behind delayed impact optimization as proposed in this work is to not use the concept of delayed impact to evaluate other fairness metrics but instead to incorporate maximization of the delayed impact itself into the learning process by adding a constraint to the loss function. However, this requires a modification of the decision-making process: usually, one of the main choices a lending institution has to make in its lending policy is the default probability threshold at or below which a loan is granted, and thereby the acceptance rate, i.e. the percentage of applicants who receive a loan. Incorporating this reality into the calculation of a single fairness metric is possible by computing the delayed impact for as many selection rates as possible, as Liu et al. (2018) did for their outcome curves. Doing so for a given set of predicted default probabilities is computationally inexpensive. Unfortunately, once this procedure is incorporated into the loss function, it adds this computational effort each time the loss needs to be calculated throughout the training process. This slows down the already time-consuming training process even further, which is why the threshold used in the loss function remains fixed throughout the following experiments. What arises from this procedure is that the acceptance rate is measured as an outcome instead of it being an input decision. In this regard, the results do not accurately reflect a real-world scenario, in which a lending institution determines the acceptance rate and then changes the cutoff probability accordingly. All metric functions in this work that incorporate a cutoff follow this modification and use the same fixed cutoff as the models. This sacrifice is determined necessary to ensure consistency.

The cutoff for receiving a loan is chosen so that the predictions of the unrestrained model result in an acceptance rate of around 40%, which resembles the average probability to receive a credit card, one of the most common forms of consumer credit in the US (Bureau of Consumer Financial Protection, 2019). This is done for each data set individually as depicted in Figure 3.

In future research, fixing the acceptance rate instead of the cutoff value in both the loss and the metric calculations would be a logical next step, given sufficient computational capacity is available. Note, that the cutoff value is determined based on predictions made by a model that is trained using only one seed for random number generation. Since, as we will see below, the delayed impact optimization models are trained several times with different seeds to receive an average performance, acceptance rates in the experiments without a penalty do not necessarily correspond with the acceptance rates established when determining the cutoff.

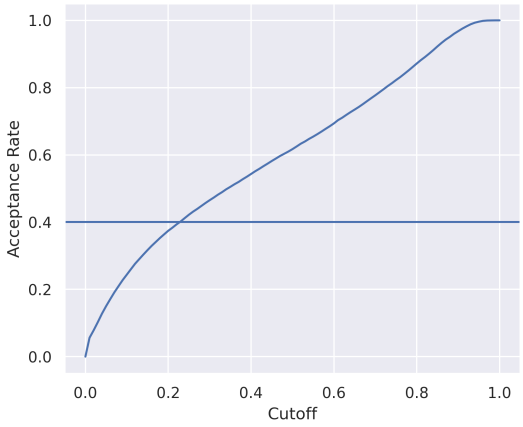

Figure 3: Determining a Static Cutoff (Simulated Data)

2.5 Fairness Processors

Methods that attempt to mitigate bias in machine learning and hence increase fairness are often referred to as fairness processors. This subsection gives a brief overview of the different types of fairness processors and explains adversarial debiasing in more detail, as it serves as a comparison in this work.

2.5.1 Types of Fairness Processors

In general, fairness processors fall into one of three categories, depending on which part of the modeling pipeline they target: pre-processors, in-processors, and post-processors.

As the name suggests, pre-processing denotes manipulating the training data before it is handed to the model and can therefore be applied to any type of model. Algorithms that belong to this group include disparate impact remover, which aims at changing feature values to increase group fairness without changing intra-group rank ordering (Feldman et al., 2015), and reweighing, which weighs observations differently, depending on their group-label-combination to achieve fair predictions (Kamiran & Calders, 2012).

In-processors, on the other hand, influence the training process itself. Many algorithms fall into this category, in which generally some constraint or regularization term is added to the training objective. The processor proposed in this work belongs to the group of in-processors since it adds a penalty to the optimization, extending the optimization goal from sole accuracy to a combination of accuracy and fairness. Other well-known in-processors include prejudice remover, which adds a fairness-aware regularization term to decrease discrimination (Kamishima et al., 2012), and adversarial debiasing (Zhang et al., 2018), which will act as a comparison for the method proposed in this work and is explained more thoroughly in the subsequent chapter.

Finally, there are post-processors that alter the predictions produced by a model. Like pre-processing, this yields the advantage that any model can be used to make predictions in the first place. Post-processors include reject option classification, a technique that changes the prediction to the desirable label for the unprivileged group and the undesirable label for the privileged group, respectively, in the cases with the highest uncertainty (Kamiran et al., 2012), and equalized odds post-processing, which creates and solves a linear program that determines which labels need to be changed to achieve equalized odds (Hardt et al., 2016).

2.5.2 Adversarial Debiasing

Adversarial debiasing was first introduced by Zhang et al. (2018) as an in-processor, capable of mitigating bias in neural networks. It is based on the work of Goodfellow et al. (2014), who proposed a framework consisting of multiple networks with divergent goals, in which one network tries to 'deceive' the other. They suggest the easily understandable analogy of

counterfeiters and the police: counterfeiters (a generative model) produce counterfeit currency, while the police (an adversarial network) try to differentiate real from counterfeit currency. This competition fuels both sides' improvement, leading to counterfeit currency becoming less and less distinguishable from real currency.

Zhang et al. (2018) take this principle and apply it to fairness optimization, making differentiation between sensitive groups the task of an adversarial network. The harder it is for the adversary to correctly predict the sensitive attribute given the outputs of the predictive network, the lower the adversary loss becomes. The adversary loss gradients then influence the weights of the predictive network, such that the network cannot change in ways that would aid the adversary in predicting the sensitive attribute. How strong this influence is, is controlled via the tunable hyperparameter α, called the adversary loss weight, which is used in the following experiments similarly to the penalty weight for delayed impact optimization, which will be thoroughly explained in the next section. This method has been chosen for comparison, because, like delayed impact optimization, it is a model agnostic in-processor, that has one major tunable hyperparameter which determines how much the training is influenced and can therefore easily be tested and visualized similarly.

3 Delayed Impact Optimization

This section exemplifies the idea of delayed impact optimization with its theoretical advantages and disadvantages, goes into detail on the implementation, and states alternative implementations that have been discarded.

3.1 Rationale

As stated above, the idea behind delayed impact is to step away from viewing fairness as something that is measured at one point in time and towards a concept that is closer to what one might call long-term fairness. For an illustrative analogy, imagine a group of people, half of which cannot swim, that stands in front of two paths towards a desirable goal. One way is short and easy but it is necessary to swim to get to the goal, the other one is long and harsh but without the need to swim. Is it fair to split the group equally, disregarding their ability to swim, or should they be split in a way that allows the largest portion of them to make it to the goal? Taking this analogy to the context of credit scoring, we may want to ask ourselves whether we want to grant loans equally or if we want to grant them in a way that maximizes the long-term welfare of the applicants. Delayed impact optimization is the attempt to achieve the latter. This is implemented by calculating the delayed impact for the unprivileged group and letting it negatively enter the standard loss term. This way, the model is incentivized to make predictions that are accurate and result in a high delayed impact.

It is sensible to also think about the theoretical implications. One convenient property is that this method is model agnostic. In general, it can be applied to any neural network, if the sensitive attribute is known. On the other hand, it is comparatively resource-intensive. Calculating the delayed impact for each individual at every time step adds computational effort to the process, making it around 9% slower than unconstrained training in experiments. In addition to this, due to the structure of the adjusted loss function discussed below, the loss can only be calculated if the binary cross-entropy is not zero since otherwise, a division by zero error would occur. This necessitates the assumption that the classifier will never be a perfect one that can assign all labels in a batch without any residual doubt, which fortunately holds for virtually any real-world prediction scenario.

3.2 Implementation of the Penalty

As previously mentioned, the delayed impact of the unprivileged group should negatively enter the loss to incentivize its decrease. This will be referred to as a penalty, as it penalizes a small delayed impact in that group. To ensure optimal performance and make use of available GPU or TPU processors, the penalty is formulated using mostly 1-dimensional tensors and scalars. Since for backpropagation to work, all functions used in the loss need to be differentiable, which comparisons like "$\leq$" are not, they need to be replaced with combinations of differentiable functions that result in the same outcome. Therefore, simply replacing all predictions smaller than or equal to τ with zero and all predictions larger than τ with one is not possible, but Equation 10 achieves the same result while using solely differentiable functions.

Equations 10 to 15 show the structure of the loss function, which uses predictions, true labels, sensitive group membership, and the parameters τ, p_+ (individual gain in case of a true negative prediction), and p_- (individual loss in case of a false negative prediction) as input and returns the loss. Equation 14 is the standard formula for binary cross-entropy.

Note, that for better readability, tensors are depicted as uppercase letters, while scalars are depicted as lowercase letters and that the first tensor, which would be $\hat{Y}$, is written out for a more intuitive visualization. Furthermore, note that the Hadamard-Product is depicted by "$\circ$", indicating element-wise multiplication, which must not be confused with matrix multiplication.

$$T = Sign\left(ReLU\left(\begin{bmatrix} \hat{y_1} \\ \hat{y_2} \\ \vdots \\ \hat{y_n} \end{bmatrix} - \tau\right)\right) - 1 * (-1) \tag{10}$$

$$P_+ = T \circ U \circ Y_{inv} * p_+ \tag{11}$$

$$P_- = T \circ U \circ Y * p_- \tag{12}$$

$$p = \overline{P_+} + \overline{P_-} \tag{13}$$

$$b = -\frac{1}{n}\sum_{i=1}^{n}(y_i * log(\hat{y}_i) + (1 - y_i) * log(1 - \hat{y}_i)) \tag{14}$$

$$l = b - \frac{p}{b} \tag{15}$$

where

T	=	Binary tensor that is one for $\hat{y} \leq \tau$
U	=	Binary tensor that is one for the unprivileged class
Y	=	True binary labels
Y_{inv}	=	Inversion of Y, i.e., zeros are turned into ones and vice versa
p_+	=	Individual gain in case of successful repayment of a granted loan
p_-	=	Individual loss in case of unsuccessful repayment of a granted loan
P_+	=	Tensor of individual gains in case of successful repayment of a granted loan
P_-	=	Tensor of individual losses in case of successful repayment of a granted loan
p	=	Total penalty
l	=	Loss
n	=	Number of elements in a tensor

It is important that in the last equation of the loss function (Equation 15) the penalty is not simply subtracted from the loss but also divided by a variable like the binary cross-entropy. Otherwise, it would enter the loss term as an added constant, which does not affect the gradients through backpropagation, due to constants being omitted in deviations. While turning the penalty from a constant into a variable value would be possible in many ways as can be seen in equations 16 to 19, dividing it by the binary cross-entropy brings an additional advantage: during earlier stages of training, the binary cross-entropy is larger, which reduces the impact that the penalty has on the loss. This allows the model to focus on making accurate predictions first, without the fairness penalty interfering too much, and thereby to quickly move in the direction of an optimum. Then, with increasing accuracy and decreasing binary cross-entropy, the effect of the penalty grows, and the model is steered in a fairer direction. A similar approach of a dynamic penalty showed promising results in the context of constrain-bound reinforcement learning (Yoo et al., 2021).

Another important question is how to choose the parameters for the individual gain and loss from successful repayment and default. The values used in the following experiments are derived from the work of Liu et al. (2018), where penalties are modeled as a change in the FICO score, one of the most widely used credit scores in the United States. They assume a score decrease of -150 in the case of a default and a score increase of +75 in the case of successful repayment. This ratio of -2:1 is transferred to the values used in the loss function and remains the same for all penalty weights. To set the arbitrary starting point for the values at a penalty

weight of one, the values represent the percentual change in the FICO score that Liu et al. (2018) assume in their paper. Since the FICO score can range from 300 to 850, this results in base values of -0.272 and 0.136.

3.3 Alternative Implementations of the Loss Function

Other structures for the final part of the loss function depicted in Equation 15 that have been tried but did not show satisfying results when tested for different hyperparameters include the following:

$$L = b + \frac{(p_p - p_u)^2}{b} \tag{16}$$

$$L = \frac{b}{p + \varepsilon} \tag{17}$$

$$L = b - p * \frac{b}{b + b * \varepsilon} \tag{18}$$

where

ε	=	Infinitesimal value
p_p	=	Penalty for the privileged group
p_u	=	Penalty for the unprivileged group

The idea behind Equation 16 is that it would not only maximize the delayed impact in the unprivileged group but instead equalize the delayed impact in both groups. This should bring the delayed impact difference between both groups closer to zero, preventing excessive favoring of the previously unprivileged group exceeding equality, which can be seen with Equation 15 for large enough penalty weights.

Similar to Equation 15, Equation 17 decreases the loss for predictions that are more desirable for the unprivileged group. The added epsilon in the denominator is necessary since, in the beginning, the penalty is always zero, which would result in an error. Since it lacks the advantage of first supporting accurate predictions in the earlier training stages before increasing the importance of fairness, it is inferior to Equation 15.

Equation 18 is supposed to approximate the simplest possible form $L = b - p$, which by itself would not be functional due to the aforementioned problem of adding a constant to the loss. Through the addition proposed in Equation 18, the outcome hardly changes since the penalty is multiplied by a value very close to one, while at the same time, the penalty is not a constant anymore but becomes a function of the binary cross-entropy. Samples of performance and fairness metrics achieved with all three alternative loss functions can be found in appendix A.

4 Methodology

This section comprises the experimental setup used to determine the performance of delayed impact optimization and to compare it to adversarial debiasing. It describes the metrics used to evaluate performance and fairness, the models that the fairness processors are applied to, and the data that they are trained and evaluated on.

4.1 Metrics

Several different performance and fairness metrics are used to evaluate each processor-data-combination. First, to have an objective measure for the discriminatory abilities of the models, the area under the receiver operating characteristic curve (AUC) is calculated. It is a good indicator of how well the model is able to correctly classify both positive and negative observations for different cutoff values.

In addition to this, the binary cross-entropy is computed. It measures how close the predicted default probabilities are to the actual targets and will often, but not always, behave inversely proportional to the AUC. It can give additional insight into the prediction process since minimizing it is the main optimization goal besides decreasing the delayed impact.

Next, the acceptance rate is returned as the percentage of potential customers who would receive a loan, given the fixed cutoff value. While the acceptance rate that the unconstrained models are supposed to produce is known, as it is the base for choosing the cutoff value, the further development of the acceptance rate for different adversary loss and penalty weights is informative since an acceptance rate becoming very high or low might be a sign that a method achieves fairer results to a large extent by accepting more or fewer applicants in general. While this would make predictions fairer – a model that accepts or declines everyone cannot be biased – it is not compliant with business goals and would give reason to question the method.

In real-world scenarios, the most valuable performance metric for profit-maximizing financial institutions is profit. Therefore, the next metric is especially important from the business perspective: the estimated profit. It is calculated based on the Expected Maximum Profit criterion (EMP), proposed by Verbraken et al. (2014), following the adaption by Kozodoi et al. (2022).

As in their work, the base scenario is the rejection of all credit applications, to allow for an intuitive interpretation. Besides that, the financial institution faces cost C_{FN} for all false negatives, i.e., applicants that are predicted to repay but default, opportunity cost C_{FP} for all false positives, i.e., applicants that are predicted to default but would have repaid and benefit B for all true negatives, i.e., applicants that are predicted to repay and do repay. True positives do not enter the estimation, because an individual that does not receive a loan and would not have repaid causes neither a loss nor a gain for the institution.

Equation 19 defines C_{FN} as a function of the loss given default (LGD), the exposure at default (EAD), and the principal A, which can vary in [0,1]:

$$C_{FN} = \frac{LGD * EAD}{A} \tag{19}$$

Following Kozodoi et al. (2022), it is treated as a random variable that equals zero with probability p_0, one with probability p_1 and is uniformly distributed in (0,1) with $F(C_{FN}) = 1 - p_0 - p_1$. Furthermore, C_{FP} is defined as the return on investment (ROI) of the loan. The estimated profit can thus be calculated as:

$$\text{Profit} = \int_0^1 \left[C_{FP} \cdot (\pi_0 (1 - F_0(\tau)) - \pi_0 F_0(\tau)) - C_{FN} \cdot \pi_1 (1 - F_1(\tau))\right] f(C_{FN}) d(C_{FN}) \tag{20}$$

where

π_1 = Prior probability of an actual default
π_0 = Prior probability of actual repayment
F_1 = Predicted cumulative density function of the scores of class one given a cutoff value τ
F_0 = Predicted cumulative density function of the scores of class zero given a cutoff value τ

In accordance with Kozodoi et al. (2022), a constant ROI of 0.2664 and the point masses $p_0 = 0.55$ and $p_1 = 0.1$ are assumed.

Table 1: Cost Matrix for All Possible Outcomes

	Predicted Label	
Actual Label	1	0
1	π_{TP} Cost: 0	π_{FN} Cost: p_-
0	π_{FP} Cost: 0	π_{TN} Benefit: p_+

Next is a novel fairness metric specific to this task. It measures the delayed impact difference, i.e., the absolute difference between the average delayed impact given the individual gain and loss parameters in the privileged group and the unprivileged group. Each mean is divided by the individual gain parameter, which keeps the results comparable for different penalty weights, since gain and loss are always affected equally. With the probabilities π_i^j for prediction-target-combination i and group j, which can be seen in Table 1, the delayed impact difference is thus defined as follows:

$$\text{Delayed Impact Difference} = \left| \frac{\pi_{FN}^{priv} * p_- + \pi_{TN}^{priv} * p_+}{p_+} - \frac{\pi_{FN}^{unpriv} * p_- + \pi_{TN}^{unpriv} * p_+}{p_+} \right| \tag{21}$$

Finally, the three standard fairness metrics that have been explained above - disparate impact, sufficiency, and separation - are computed.

4.2 Models

To test the performance of delayed impact optimization, two very similar neural networks are used for the two data sets. Since the goal is not to achieve maximum predictive performance with sophisticated models but to investigate the performance of a fairness processor, simple models suffice for this purpose and bring the advantage of shorter training times. Both models consist of two ReLU-activated hidden layers and one single-node, sigmoid-activated output layer, with a capacity of 30 and 35 hidden nodes, respectively. The batch size is set to 128 for both models and 50 epochs of training make up each iteration. These hyperparameters are the outcome of grid search optimization without k-fold cross-validation. A fixed split is chosen deliberately because cross-validation would ensure better generalization capabilities of the model, which once again is not necessary for understanding the performance of the processors. The exact values of the hyperparameter grid can be seen in Table 2.

Table 2: Hyperparameter Grid

Parameter	**Candidates**
Nodes for the first hidden layer	10,15,20
Nodes for the second hidden layer	10,15,20
Learning rate	$5e^{-5}$,$1e^{-4}$
Batch size	64,128,256

All models are trained using the Adam optimizer (Kingma & Ba, 2014). Early stopping is not used, as not being trained for the same number of epochs would decrease the comparability between different runs, while the prevention of overfitting can be neglected when evaluating the methods. All parameters that are determined optimal for delayed impact optimization are, where possible, transferred to the adversarial debiasing models to ensure comparability. This adds up to four models to work with, one for each data-processor-combination.

To decrease the chance of randomness distorting the performance, the two delayed impact optimization models are trained with the same parameters five times with different seeds for all relevant random processes. The results are then averaged across these five runs to determine the values that are shown in the tables and graphs. This procedure is put in place to counter partially large fluctuations in predictive performance that can be observed when deploying delayed impact optimization without this countermeasure and could otherwise lead to distortions. Since adversarial debiasing does not result in similar fluctuations, this is only applied to delayed impact optimization.

The ranges of both the penalty weights and the adversarial loss weights for which all metrics are calculated have been determined in previous experiments with larger and smaller ranges. This way, the effective weight ranges for each method-data-combination can be analyzed without

distortion from results that are irrelevant because they neither improve fairness nor predictive accuracy consistently.

It is also important to mention that the data sets get split into two partitions, one training data set and one test data set, without another validation set. This would not have been sufficient if the aim was to build and choose among others a model with optimal generalization abilities, where being able to evaluate its performance on entirely new data in the form of a separate validation data set is indispensable. But, since model optimization is only a subordinate task in this pipeline, doing so would not have been worth further reduction of the amount of training data available to the model.

4.3 Data

The main data set used for developing and testing the method is a simulated data set which is produced using code provided by Kozodoi et al. (2020) with minor adjustments to fit this task. It generates binary target credit data and produces continuous features using multivariate Gaussian distribution, categorical features based on latent variables from the continuous features, and binary features using binomial distributions.

To inject bias into the data for this experiment, one binary feature is chosen to be the sensitive attribute, where the feature being one implies belonging to the privileged group, whereas it being zero means belonging to the unprivileged group. To manifest this difference in the distribution of the other features, the ratio of bad outcomes, meaning the inability to repay the loan, is set to 30% in the unprivileged group and 15% in the privileged group. Accordingly, the continuous features' normal distributions are shifted to higher values for the privileged group. This results in a disparate impact of 0.8235, which is close to the aforementioned 80% rule set by the US Equal Employment Opportunity Commission (*Albemarle Paper Co. v. Moody*, 1975) and leaves ample room for optimization without being too unrealistic. Regarding privileged and unprivileged individuals, this data set is perfectly balanced, which is not realistic, as generally minorities experience prejudice and bias, but serves the purpose of making discrimination as present as possible in the data set to enable the evaluation of the method even if it turns out to perform inadequately on real-world data. Lastly, to allow the calculation of expected profits, normally distributed loan amounts are randomly assigned to all observations, with a mean of 7000$US and a standard deviation of 2000$US based on the average personal loan amount in the US (TransUnion LLC, 2021).

For testing the generalization of the method, data from the Kaggle competition 'Give Me Some Credit'[1] (GMSC) is used. This data set does not explicitly provide loan amounts, which is why they are randomly assigned from the same normal distribution as well. The disparate impact in

[1] Source: https://www.kaggle.com/c/GiveMeSomeCredit

this data set is 0.9518, which is considerably higher than for the simulated data and might lead to comparatively fair models without optimization.

The Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) is applied to both data sets to make them more balanced. The simulated data is designed to be comparatively balanced, therefore it is not strictly necessary in this case, however, since it notably improves the accuracy of all models for the GMSC data, it is used on both sets to rule out differences in outcomes that originate from differences in preprocessing. Details of both data sets can be seen in Table 3.

Table 3: Data Sets

Data Set	Obser-vations	Continuous Features	Binary Features	Categorical Features	Percentage Unprivileged	Default Percentage Unprivileged	Default Percentage Privileged
Sim.	100,000	2	1	2	50%	30%	15%
GMSC	150,000	11	1	0	7.35%	11.11%	6.61%

5 Results & Discussion

This section depicts the key results of all experiments and investigates both the reasons behind those results, as well as their implications. First, the performance and fairness metrics of all four processor-data-combinations are analyzed individually and compared. Thereafter, the profit-fairness trade-off of both fairness processors is compared for each data set by investigating the Pareto frontiers.

5.1 Performance and Fairness Analysis

The following subsections present and interpret the development of all performance and fairness metrics for changing hyperparameters, i.e., increasing penalty and adversary loss weights.

5.1.1 Delayed Impact Optimization - Simulated Data

As can be seen in Figure 4, model performance in terms of the AUC remains stable and high for penalty weights lower than 60. Beyond this point, the AUC drastically decreases and fluctuates increasingly. For penalty weights larger than 90, the model mostly does not exceed random guessing, indicated by an AUC of 0.5. It is notable that only for individual losses below -16.32 and individual gains above 8.16 - the base penalty parameters multiplied by a weight of 60 - the penalty significantly influences the predictions. This is also true for all other observed metrics. Note, that due to the pronounced volatility for higher weights, the results for smaller weights get compressed and while the AUC appears to be static for weights for small weights due to compression from the wide range of the axis, there is a minor increase present in the data.

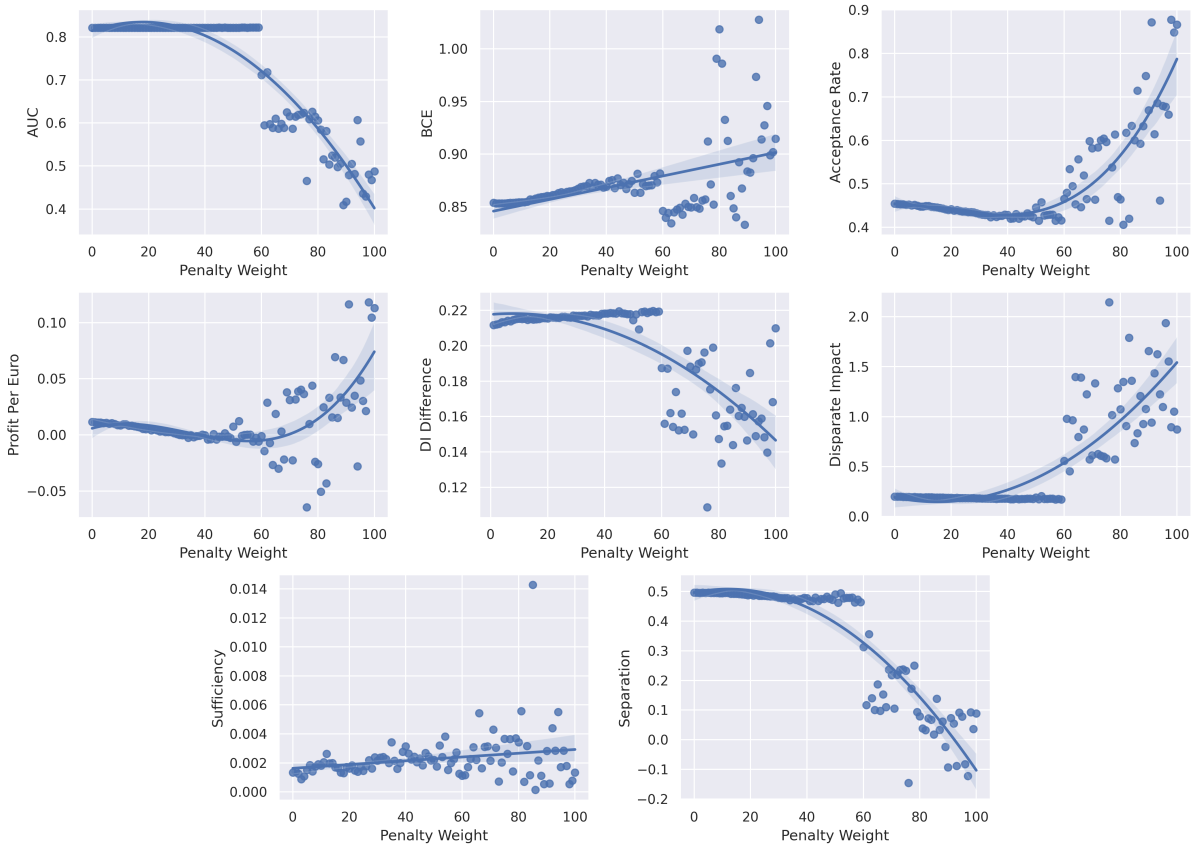

Figure 4: Performance and Fairness Metrics - Delayed Impact Optimization (Simulated Data)

The binary cross-entropy moderately increases until at the same penalty weight of 60, it diminishes below the value for an unconstrained model, depicted by a penalty weight of 0, before it increases again and becomes increasingly volatile. This growing volatility is observable in all metrics.

On the right, we see the acceptance rate starting at 0.46 without a penalty, then decreasing to just above 0.4 for penalty weights between zero and 50. Then, from a penalty weight of 50 upwards, both the acceptance rate and its variance increase, reaching acceptance rates just below 0.9 for penalty weights close to 100. Interestingly, as can be seen in Table 4 and is evident from the graphs, the AUC has a strong and statistically significant negative correlation with the acceptance rate. Note, that the set of weights for which all metrics are calculated can be thought of as a sample from the infinite 'population' of all possible weights in the same range. Thus, a correlation coefficient is defined as statistically significant in this paper if the corresponding p-value, i.e., the probability that more or differently chosen weights would have resulted in a coefficient closer to zero, is below 5%. A possible explanation for the negative correlation between AUC and acceptance rate is that delayed impact optimization with higher penalty weights incentivizes the model to accept more potential customers, i.e., predict lower default probabilities and therefore fewer defaults, which leads to more misclassification and a decreasing AUC. Looking at how the histograms of all scores change from lower to higher penalty weights in animation 1,

which can be found in the digital appendix[2] of this thesis together with similar animations for the other three experiments, this hypothesis appears to be reinforced, as the distribution mass shifts towards lower predictions with increasing weights.

The development of the estimated profit closely resembles that of the acceptance rate, a connection that we will see in all experiments and fathom below. With increasing penalty weights, the estimated profit slightly decreases until a weight of around 50 is reached, where it begins to increase but also starts to fluctuate more, ranging from less than -0.05 to more than €0.12 profit per euro lent.

Three of the fairness metrics are influenced in a desirable way. The delayed impact difference, disparate impact, and separation remain relatively unaffected for penalty weights up to 60 at around 0.22, 0.2, and 0.5 with a marginal trend in the respectively unfair direction, before they show the same pattern of moving in the desirable, i.e., fairer, direction with increasing variance for weights exceeding 60. For some weights close to 100, separation becomes negative. This downwards trend suggests that for increasing weights, the predictions will become less fair again in terms of separation. Only the sufficiency worsens from start to end, however, only moderately so. Notably, the sudden change at a weight of 60 cannot be seen here, instead, there is a more continuous change in sufficiency and variance thereof.

Looking at what all metrics, besides sufficiency, have in common, there appears to exist a threshold for some data, below which the method does not have a notable effect on the predictions, neither in terms of predictive performance, nor the fairness of those predictions.

Table 4: Spearman Correlation Matrix - Delayed Impact Optimization (Simulated Data)

	PW	AUC	BCE	PPE	DID	DI	Sep.	Suf.	AR
PW	1.000								
AUC	-0.592	1.000							
BCE	0.374	(0.094)	1.000						
PPE	(0.169)	-0.490	-0.246	1.000					
DID	-0.561	0.962	(0.131)	-0.331	1.000				
DI	0.572	-0.964	(-0.105)	0.380	-0.974	1.000			
Sep.	-0.943	0.601	-0.373	(-0.032)	0.593	-0.570	1.000		
Suf.	0.221	(0.089)	0.258	-0.423	(-0.054)	(0.054)	-0.239	1.000	
AR	0.482	-0.821	-0.272	0.763	-0.742	0.775	-0.388	-0.238	1.000

Coefficients with a p-value larger than 0.05 are depicted in parentheses. Abbreviations: PW = Penalty Weight, PPE = Profit per Euro, DID = Delayed Impact Difference, DI = Disparate Impact, Sep. = Separation, Suf. = Sufficiency, AR = Acceptance Rate

[2]Digital Appendix: https://github.com/time-stack/masters-thesis

5.1.2 Delayed Impact Optimization - Give Me Some Credit

When looking at Figure 5, it becomes apparent that similar to the results obtained with the simulated data, the AUC of the model trained and tested on the GMSC data set decreases with increasing penalty weights. However, it does not start to decrease after some threshold is reached. Instead, it falls right from the beginning with a relatively consistent decline between an AUC of 0.718 for a penalty weight of zero and an AUC of 0.618 for a penalty weight of 100. As for the simulated data, all performance and fairness metrics start with a relatively small variance that increases with increasing penalty weights.

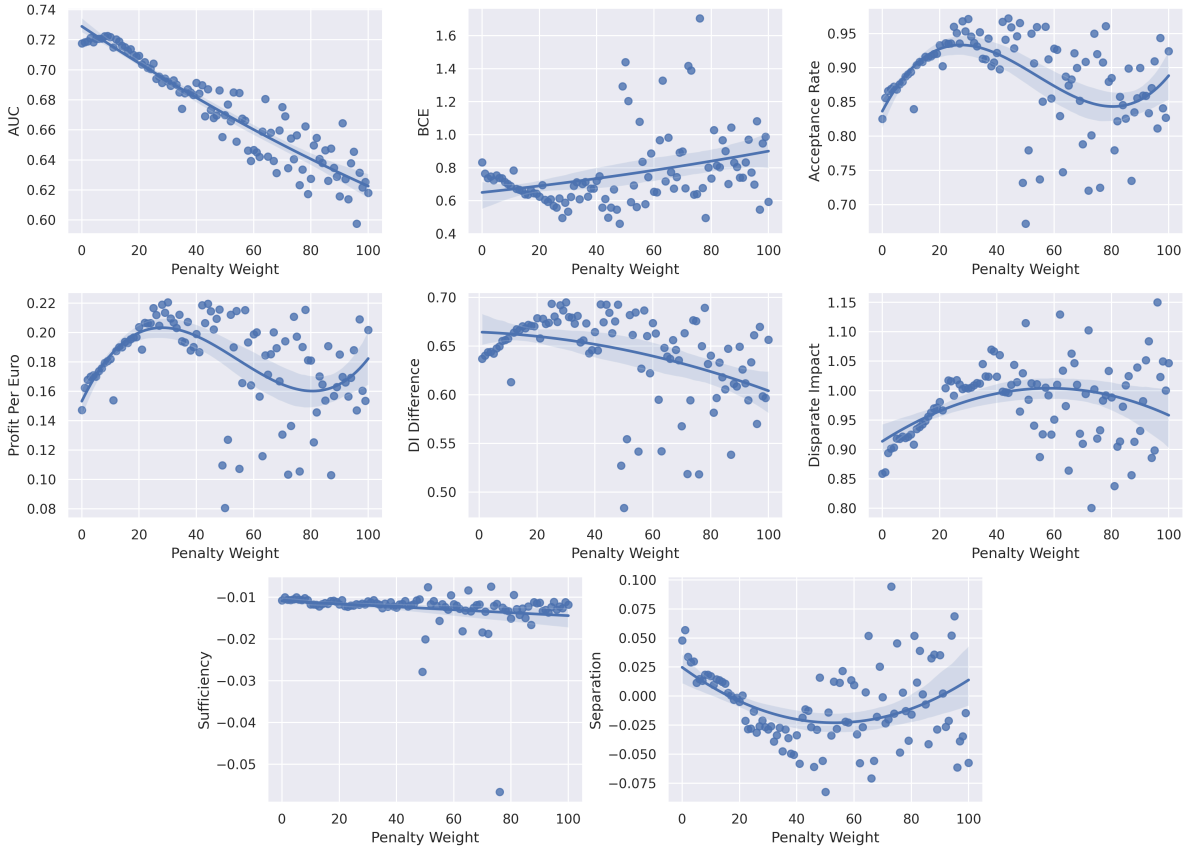

Figure 5: Performance and Fairness Metrics - Delayed Impact Optimization (Give Me Some Credit)

The binary cross-entropy decreases for penalty weights up to 30 before it starts to increase and fluctuate more, which can be explained partially by the very high and increasing acceptance rate that starts at 82.5% and is significantly higher than the 40% that it is supposed to start at. This is because the training and measuring process is repeated for the same penalty weight with several different seeds to stop random noise from influencing the predictions too much, while the process to define a cutoff value only takes one seed. Discrepancies between the supposed and actual acceptance rate are the consequence. In future research, this potential error could be avoided by finding cutoff values for all seeds that will be used and utilizing their mean as the final cutoff. At its highest point, the acceptance rate reaches 96.8%. Being this close to indiscriminately accepting everyone may also lead to distortion of the other metrics. It should

thus be noted that these findings are somewhat limited in their informative value and validity. The binary cross-entropy, for instance, decreases for weights between zero and 20, because, as can be seen in animation 2, most predicted default probabilities are relatively high in the beginning. This is compensated by a high cutoff value that was chosen to achieve an acceptance rate of around 40% but results in much higher acceptance rates for other seeds. With decreasing probabilities being predicted for higher weights, the comparatively high binary cross-entropy decreases, because the default probabilities were overestimated from the beginning. The acceptance rate, on the other hand, increases, because more predicted probabilities fall short of the high cutoff value. While the discrepancy between cutoff determination and experimental runs exists for all models due to the identical process, in all other instances, the acceptance rate is close to 40% for the unaffected models, which suggests that the difference, originating from different seeds, is especially high for this specific model-data-combination. Nevertheless, even for exceptionally high acceptance rates, the trends in the results can still be interpreted and carry valuable information.

Looking at the estimated profit and its change, it becomes apparent that it once again tightly resembles the changing acceptance rate and while correlation does not imply causation, the statistically significant and almost perfect correlation expressed by a Spearman correlation coefficient of 0.996, as seen in Table 5, and the fact that we see the same connection with different data encourages the hypothesis that a model that accepts more customers generally achieves higher profits with the previously elucidated profit estimation formula. Of course, the data set itself should not be disregarded as the source of this behavior but since the credit amounts, which are the basis for profit calculation, are randomly and independently distributed by design, the profit calculation itself and the sampling and oversampling are left as possible sources for this behavior. Reinspecting the label distributions before and after oversampling does not yield any signs of relevant changes, which leaves the profit estimation itself as a possible source of error.

To test whether it rewards higher acceptance rates with higher profits, we can imagine a hypothetical data set that is perfectly balanced, i.e., half of the customers repay and the other half default. Also, we may assume that the credit amounts are distributed independently and identically and are normalized to a mean of one. With the profit estimation used for the experiments, we can now calculate two extreme cases, one where the model accepts everyone and one where every potential customer is declined, to then compare the estimated profit. For the former case with an acceptance rate of one this gives an estimated profit of -0.01405, whereas, for the latter case with an acceptance rate of zero, the estimated profit is -0.1322. Accepting no one is almost ten times as costly as accepting everyone. Because the opportunity cost is part of the calculation, while no benefit is associated with the avoidance of defaults, the profit calculation, as described in Section 4, overestimates the overall cost of predicting defaults. In future research, this could be avoided by either not considering the opportunity cost of not receiving interest for

a loan that would have been repaid or by also taking into account the avoidance of potential losses whenever a defaulter is correctly classified. In the context of this work, this problem is addressed by focusing on the changes in estimated profit rather than its absolute value and by comparing it to the more objective measure of the AUC to discover any discrepancies.

Another unexpected behavior is that the delayed impact difference, a metric that closely resembles what is meant to be minimized through the penalty by increasing the delayed impact in the unprivileged group, also shows the same trend as the estimated profit, and is highly correlated with the acceptance rate with a correlation coefficient of 0.977 that is significant on the 5% level. A possible explanation for this is that the delayed impact optimization becomes ineffective for very high acceptance rates. However, it is hard to explain that it is the exact opposite of what one would expect, whereas for the simulated data it does decrease the difference, even if only for penalty weights above 60. This difference between the data sets further supports the supposition that these unexpected trends arise from the specific data-model-combination of delayed impact optimization and the GMSC data and not necessarily the method in general. Testing it with more data sets would be an interesting starting point for further research.

Continuing with the standard fairness metrics, the disparate impact increases with increasing penalty weights to around one for a penalty weight of 20, which is the fairest achievable result regarding this metric. The fact that this is achieved at an acceptance rate below 100%, which would always lead to perfectly fair results, shows that the achieved fairness is not solely due to the high acceptance rate and supports the general functionality of delayed impact optimization. With higher penalty weights, once again the results are increasingly volatile. Nevertheless, the trend remains with the disparate impact reaching 1.1, implying results that are unfair for the previously privileged group. Only for weights larger than 50, the trend appears to shift to decreasing disparate impact again. Even though this pattern also resembles the acceptance rate to some degree, the statistically significant correlation coefficient of disparate impact and acceptance rate is 0.251 and thus not as alarming as for profit and delayed impact difference.

Like the disparate impact, the separation, i.e., the average odds difference, is affected in a desirable way and reaches close to optimal fairness, expressed by a separation metric of zero for penalty weights close to 20, before it becomes negative and therefore less fair again, which allows the same argumentation as for the disparate impact and further supports that the method is capable of achieving fair results when applied to real-world data.

This leaves sufficiency, measured as the predictive parity difference, a criterion that is satisfied if the sensitive attribute is statistically independent of the target given the prediction. It stays at around -0.01 for most penalty weights with a slightly negative trend. It is important to note that if the targets are not independent of the sensitive attribute and the predictions are a binary classifier with a nonzero false positive rate, the criteria of separation and sufficiency cannot both hold (Barocas et al., 2019). For the data used here, both can be assumed to be true: targets

and sensitive attribute are not independent, indicated by the previously mentioned disparate impact that is not one in both of the original data sets, and the false positive rate is not zero for the unconstrained models, making it unlikely to become zero for higher penalty weights that generally decrease the accuracy and tend to encourage misclassification. Therefore, seeing only one of these criteria satisfied is not surprising. In summary, the unconstrained model already seems to produce predictions with a relatively small predictive parity difference and delayed impact optimization appears to have only a minor and inconsistent effect on it.

Comparing these findings to what we see when delayed impact optimization is applied to the simulated data, it becomes apparent that there exist certain penalty weight thresholds for both data sets, at which the effects of the method change. With the simulated data, there is close to no effect for weights below 60, whereas with the GMSC data, the trend of most metrics points in one direction for lower weights before it changes to the opposite direction for higher weights. In contrast to the more stable and homogeneous changes achieved with adversarial debiasing we will see below, this implies the need for even more in-depth optimization of the penalty weight hyperparameter to obtain satisfying results.

Table 5: Spearman Correlation Matrix - Delayed Impact Optimization (Give Me Some Credit)

	PW	AUC	BCE	PPE	DID	DI	Sep.	Suf.	AR
PW	1.000								
AUC	-0.943	1.000							
BCE	0.353	-0.366	1.000						
PPE	-0.272	0.253	-0.951	1.000					
DID	-0.421	0.423	-0.953	0.969	1.000				
DI	0.216	-0.325	(-0.171)	0.283	(0.104)	1.000			
Sep.	(-0.145)	0.261	(0.179)	-0.296	(-0.115)	-0.917	1.000		
Suf.	-0.484	0.463	(-0.193)	(0.147)	0.270	-0.428	0.430	1.000	
AR	-0.291	0.276	-0.959	0.996	0.977	0.251	-0.262	(0.176)	1.000

Coefficients with a p-value larger than 0.05 are depicted in parentheses. Abbreviations: PW = Penalty Weight, PPE = Profit per Euro, DID = Delayed Impact Difference, DI = Disparate Impact, Sep. = Separation, Suf. = Sufficiency, AR = Acceptance Rate

5.1.3 Adversarial Debiasing - Simulated Data

At first glance when looking at Figure 6, it stands out that the data points of all metrics are arranged in U-shapes and inverted U-shapes, indicating some performance minimum and fairness maximum for adversary loss weights of around 0.05. From the highest to the lowest value, the AUC decreases by less than 0.01, and the binary cross-entropy increases by less than 0.01, which is a very small drop in performance for a separation decrease from 0.20 to 0.06. With an AUC decrease of 0.9%, the separation metric can be decreased by 73.7%, compared to delayed impact optimization, where for the first relevant change, which is a 13.3% AUC decrease, separation is only lowered by 37.0%. Adversarial debiasing outperforms delayed impact optimization in terms of the achieved additional fairness in relation to the loss in predictive performance.

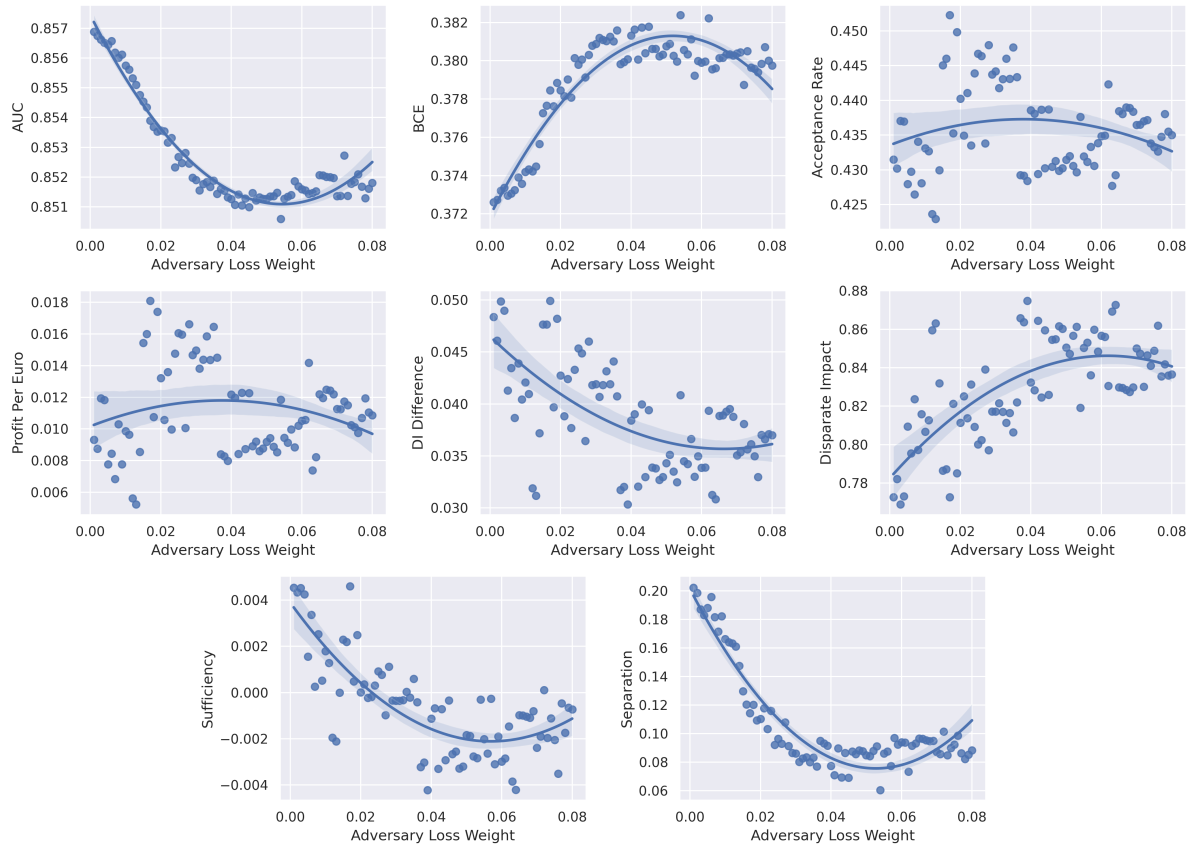

Figure 6: Performance and Fairness Metrics - Adversarial Debiasing (Simulated Data)

Meanwhile, the acceptance rate seems to fluctuate arbitrarily, but once again the connection between estimated profit and acceptance rate becomes obvious, also expressed by a statistically significant Pearson correlation coefficient of 0.999, seen in Table 6, suggesting that the estimated profit, in this case, is a direct consequence of how many potential customers are accepted. Except for delayed impact optimization being applied to the simulated data, where it is high but not as close to one, the correlation between profit and acceptance rate looks like this for all experiments, which can be traced back to the biased profit estimation function as explained in the previous section.

The delayed impact difference is considerably lower than for all weights in the delayed impact difference optimization case with this data set and continues to further decrease for higher weights. Similarly, the disparate impact starts higher than with delayed impact optimization, implying fairer predictions even for low weights, and continues to increase. Unlike in all other experiments, one can see that the sufficiency of the predictions clearly improves for increasing weights. The requirements for sufficiency to be satisfied are even almost met, i.e., the predictive parity difference is close to zero.

Lastly, the separation starts at around 0.2, compared to 0.5 for delayed impact optimization. It then follows a U-shape, reaching an optimum of around 0.06 before increasing again with growing adversary loss weights.

Focusing on the separation and especially comparing it to the separation achieved with delayed impact optimization in Figure 4, we see that delayed impact optimization can 'push' the separation further towards a fair result and apparently beyond that into unfairness again which, if we look at the disparate impact, probably puts the previously privileged group at a disadvantage. Adversarial debiasing, on the other hand, appears to have a separation minimum above zero that it does not fall below.

Table 6: Spearman Correlation Matrix - Adversarial Debiasing (Simulated Data)

	ALW	AUC	BCE	PPE	DID	DI	Sep.	Suf.	AR
ALW	1.000								
AUC	-0.656	1.000							
BCE	0.517	-0.826	1.000						
PPE	(0.017)	(0.044)	0.370	1.000					
DID	-0.557	0.554	(-0.162)	0.691	1.000				
DI	0.610	-0.608	0.235	-0.619	-0.992	1.000			
Sep.	-0.572	0.877	-0.949	-0.281	0.240	-0.307	1.000		
Suf.	-0.663	0.687	-0.374	0.472	0.927	-0.958	0.407	1.000	
AR	0.014	0.039	0.377	0.999	0.688	-0.616	-0.286	0.471	1.000

Coefficients with a p-value larger than 0.05 are depicted in parentheses. Abbreviations: ALW = Adversary Loss Weight, PPE = Profit per Euro, DID = Delayed Impact Difference, DI = Disparate Impact, Sep. = Separation, Suf. = Sufficiency, AR = Acceptance Rate

5.1.4 Adversarial Debiasing - Give Me Some Credit

Looking at the AUC in Figure 7 and how it changes for increasing adversary loss weights when adversarial debiasing is applied to the GMSC data set, we see the same downwards trend as with delayed impact optimization. However, the decrease is significantly less pronounced with a difference of less than 0.021 between the lowest and highest AUC. A direct comparison is not possible only considering the AUC and the weights since the weights, as well as their ranges, are not comparable between delayed impact optimization and adversarial debiasing. What can be compared are the Pareto frontiers that show the relationship between AUC and separation, which are available in appendix A, and their profit-separation counterparts, which will be thoroughly analyzed below. What also stands out, is that the AUC appears to be more volatile for lower weights in contrast to what is measured for delayed impact optimization, where it is more volatile for higher weights. Except for the disparate impact, this behavior can be observed for all other metrics as well. This appears to be another difference between delayed impact optimization and adversarial debiasing. With the former, the volatility of most metrics is smaller for lower penalty weights, while with the latter, their volatility is smaller for higher adversarial loss weights.

The binary cross-entropy decreases for higher weights, which is atypical and the opposite of what happens in the other three cases, but not a reason for concern because a decrease in binary cross-entropy does not imply an increase in accuracy: while the AUC measures the rank order

of all predictions, i.e., whether the predictions for observations with an actual target of one are higher than those for predictions with an actual target of zero, the binary cross-entropy measures the difference between the actual probability distribution and the predicted distribution. Therefore, they do often, but not necessarily, correspond. The acceptance rate decreases only slightly for increasing weights.

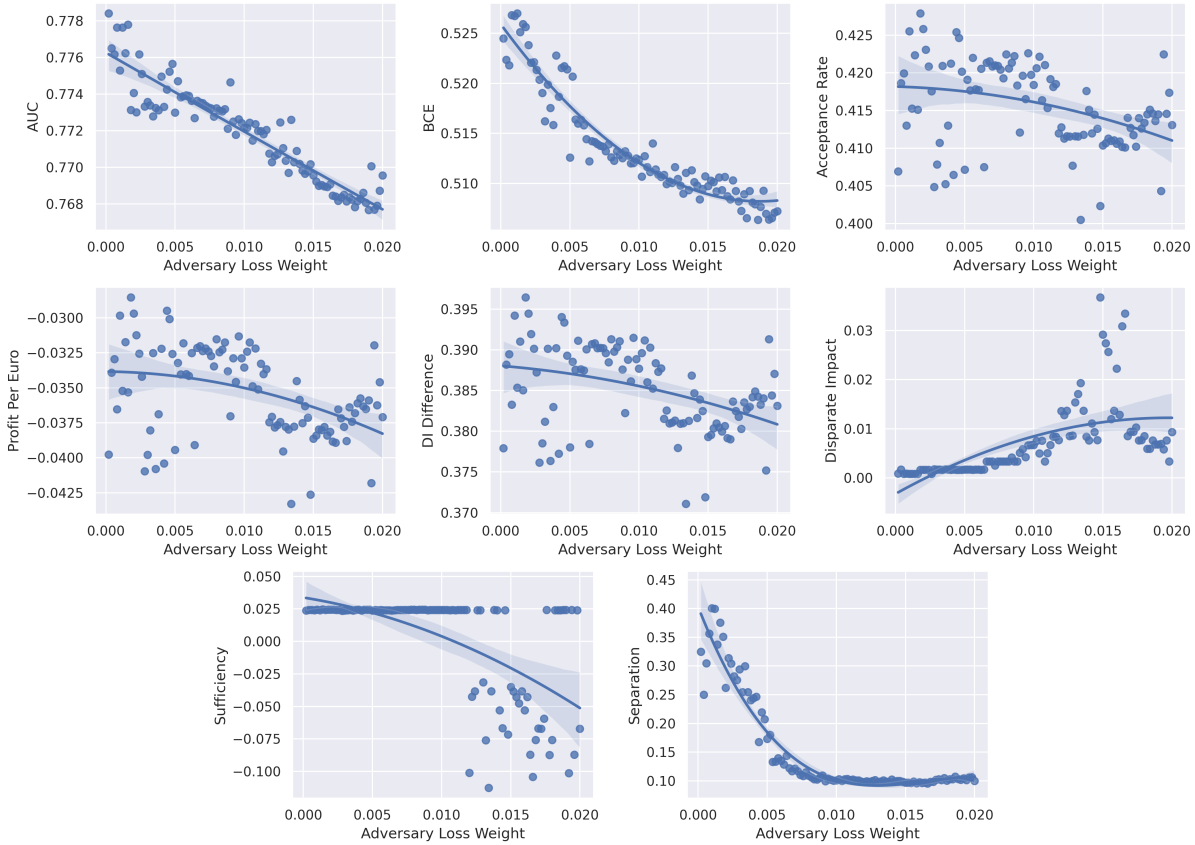

Figure 7: Performance and Fairness Metrics - Adversarial Debiasing (Give Me Some Credit)

The estimated profit lies below zero even without being constrained by the adversary. Once again, this appears to be linked to the acceptance rate that is slightly lower than with the simulated data, which produced similarly small but positive profits.

Inspecting the delayed impact difference, one can observe the same downwards trend as with delayed impact optimization but from a lower starting point, presumably due to the large difference in acceptance rates that can influence fairness by treating more people equally for acceptance rates closer to one. Similar to the results obtained when delayed impact optimization is applied to this data set, the correlation coefficients between acceptance rate and profit, as well as acceptance rate and delayed impact difference, are both significant on the 5% level and very close to one as can be seen in Table 7. This substantiates the hypothesis that accepting many customers not only increases fairness but also profits and further supports the notion that this correlation is not a defect of the delayed impact optimization method but is rather rooted in the profit estimation itself.

Moving on to the standard fairness metrics, the very low disparate impact especially stands out, which implies that the probability to receive the desirable output in the unprivileged group is very low compared to the probability in the privileged group. In fact, the percentage of negative predictions based on the testing data, i.e., no default being predicted, is 0.5352 in the privileged group, compared to 0.9996 in the unprivileged group for an adversary loss weight of 0.001 using the previously defined threshold value of 0.26. With a higher weight of 0.015, the ratio is 0.5528 to one. It exceeds the scope of this study to examine why exactly the network under the influence of adversarial debiasing predicts most customers from the unprivileged group to successfully repay their loan, even though in reality, 11.11% of all unprivileged individuals in the ground data fail to do so, which in addition is increased for the training data through synthetic minority oversampling. Still, the disparate impact increases with increasing weights, but keeping the deviating distribution in mind, the meaningfulness of this result must be questioned.

Again, the sufficiency remains relatively unaffected around 0.025. Only for weights larger than 0.011, the predictive parity difference is negative and therefore unfair in favor of the unprivileged group with values reaching down below -0.1.

Lastly, the separation steeply decreases from high values of up to 0.4, before the marginal separation decrease from each weight increase becomes smaller until it stabilizes around 0.1. As seen before, it seems to be a key difference between adversarial debiasing and delayed impact optimization that the former reaches some fairness maximum in respect to separation and the disparate impact that in the tests are not a perfectly fair solution, while the latter can push the outcome to a fair distribution and beyond. This also constitutes a main weakness when it comes to delayed impact optimization, as it does not stop at a fair result. Of course, this can be avoided by testing a larger range of weights and deciding on one that does produce the fairest outcome. However, this can be resource-intensive and might not be feasible for large data sets. It would therefore be more convenient to have a penalty function that punishes unfair results in both directions, similar to Equation 16, and not only results that are unfair for the group that is disadvantaged in an unconstrained environment. Unfortunately, this approach did not reliably improve fairness in preliminary tests. What needs to be addressed at this point, however, is that zeroing in on hyperparameters like the penalty weight is also necessary with other fairness processors. Nevertheless, ruling out the possibility of overshooting the fairness target would be desirable and might be another starting point for future research.

All metrics, except separation, change by only moderate amounts from the lowest to the highest adversary loss weight. At first, this might appear to suggest that the chosen weight range is too small. Hence, it is important to once again mention that this range is a deliberate choice, made after investigating wider ranges with larger intervals which showed that for weights that exceed 0.02, not only does the separation, which appears to be the fairness measure that profits the most from adversarial debiasing in this case, improve no more, but also the other fairness metrics start to become very volatile and shift in less desirable directions again.

Table 7: Spearman Correlation Matrix - Adversarial Debiasing (Give Me Some Credit)

	ALW	AUC	BCE	PPE	DID	DI	Sep.	Suf.	AR
ALW	1.000								
AUC	-0.944	1.000							
BCE	-0.965	0.940	1.000						
PPE	-0.400	0.437	0.449	1.000					
DID	-0.400	0.437	0.447	0.997	1.000				
DI	0.840	-0.825	-0.840	-0.572	-0.582	1.000			
Sep.	-0.796	0.769	0.764	0.394	0.411	-0.936	1.000		
Suf.	-0.548	0.547	0.586	0.851	0.840	-0.674	0.517	1.000	
AR	-0.389	0.424	0.437	0.998	0.998	-0.565	0.392	0.844	1.000

Coefficients with a p-value larger than 0.05 are depicted in parentheses. Abbreviations: ALW = Adversary Loss Weight, PPE = Profit per Euro, DID = Delayed Impact Difference, DI = Disparate Impact, Sep. = Separation, Suf. = Sufficiency, AR = Acceptance Rate

5.2 Pareto Frontiers

Due to the profit-fairness trade-off, maximizing fairness in credit scoring scenarios is a multi-objective optimization. On one hand, fairness criteria need to be met to a certain degree due to regulation, on the other hand, an institution wants to maximize its profit and therefore minimize the misclassification cost of its machine learning models. The Pareto frontier, also often called the Pareto front, consists of all Pareto efficient solutions, i.e., all solutions that are not strictly dominated by any other.

In this credit scoring scenario, these are the solutions that constitute the best combinations of high profit and fairness. Separation has been chosen as one fairness criterion to be depicted in the frontiers since it appears most suitable for the context of credit scoring in that it accounts for imbalanced misclassification cost and thereby for both the interest of the customer and the loan market, represented by the financial institution (Kozodoi et al., 2022). The delayed impact difference serves as a second fairness metric as it is the closest resemblance to the secondary training objective.

5.2.1 Delayed Impact Optimization vs. Adversarial Debiasing (Simulated Data)

Since in all experiments, negative separation was always more costly than positive separation, the Pareto frontiers show only non-dominated solutions with a separation metric larger than or equal to zero. As Figure 8 shows, delayed impact optimization can decrease the separation metric for the simulated data by around 0.05 at the cost of a profit decrease of €0.020 per euro lent. This is relatively expensive compared to adversarial debiasing, that for the same data set achieves a similar decline in separation of slightly less than 0.05 at a profit decrease of only €0.006 per euro granted, observable in Figure 9, which is less than a third of the cost.

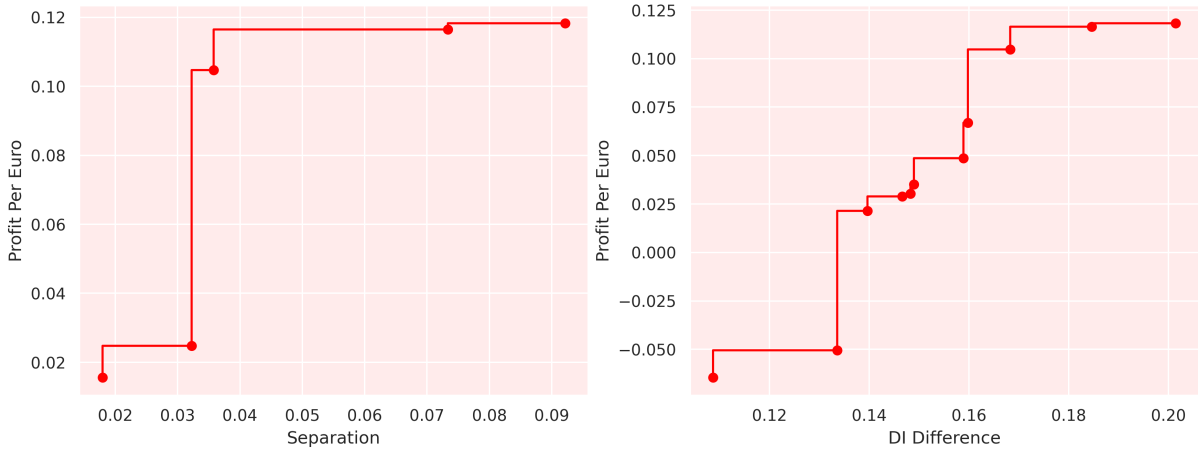

Figure 8: Pareto Frontiers: Profit vs. Fairness - Delayed Impact Optimization (Simulated Data)

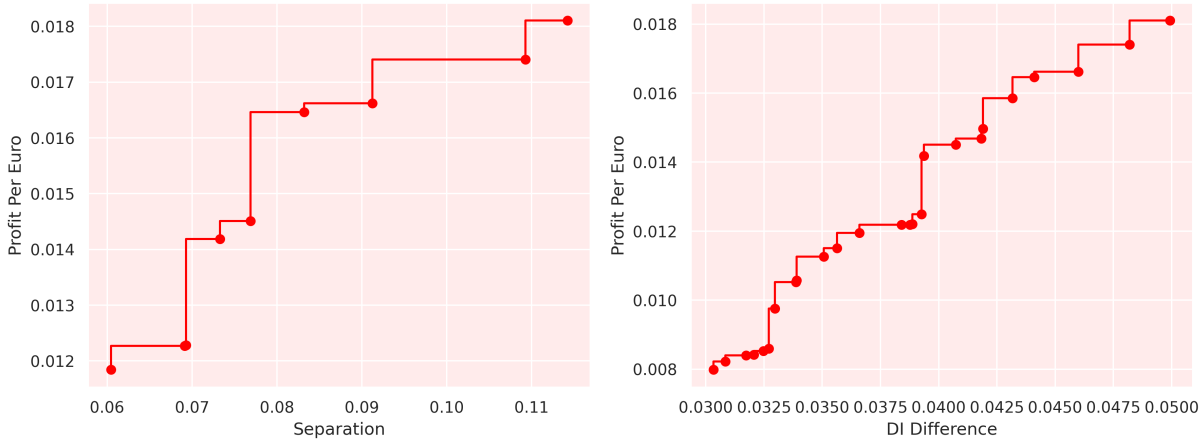

Figure 9: Pareto Frontiers: Profit vs. Fairness - Adversarial Debiasing (Simulated Data)

Getting closer to zero is significantly more expensive. Decreasing the separation below 0.020 comes at the cost of more than €0.100 per euro spent, which is almost all of the profit.

The delayed impact difference decrease, on the other hand, is accompanied by a relatively steady profit decline. A decrease from 0.200 to 0.150 comes at a cost of roughly €0.073 less profit per euro, another similar decrease from 0.150 to 0.105 costs €0.110 in marginal profit. The non-dominated combinations of delayed impact difference and profit that adversarial debiasing produces, seen on the right in Figure 9, look similar but on a smaller scale. The maximum decrease in delayed impact difference is 0.020 at a cost of €0.010 per euro lent, which again makes it more cost-effective than delayed impact optimization.

It must also be noted that the delayed impact difference is extremely small even for the smallest weights compared to all other method-data-combinations including adversarial debiasing applied to the GMSC data, although all other metrics do not indicate strikingly fair predictions. This implies that the model produces predictions that are relatively fair in terms of the delayed impact difference between both groups even without large adversary loss weights, which, bearing in mind that the marginal cost of fairness tends to increase with fairer results, as can be

derived from the Pareto frontiers' tendency to follow a rectangular shape, only makes adversarial debiasing appear even more cost-effective compared to delayed impact optimization.

5.2.2 Delayed Impact Optimization vs. Adversarial Debiasing (Give Me Some Credit)

In Figure 10, one can see the results of delayed impact optimization applied to the simulated data. The separation is very low even for small weights, which, as discussed above, is a likely consequence of the too-high acceptance rates. Still, the metric can be decreased from 0.016 to almost zero, which decreases the marginal profit by €0.020. This is more costly than adversarial debiasing, which achieves a separation decrease of 0.250 for less than a fifth of the cost, as can be seen in Figure 11. However, as can be seen in all graphs and is shown by Kozodoi et al. (2022), in most cases the marginal cost of fairness increases with more fairness, so moving from very fair predictions to perfectly fair predictions will often be prohibitively costly. Thus, a direct comparison of fairness gains from exceedingly different starting points can be misleading.

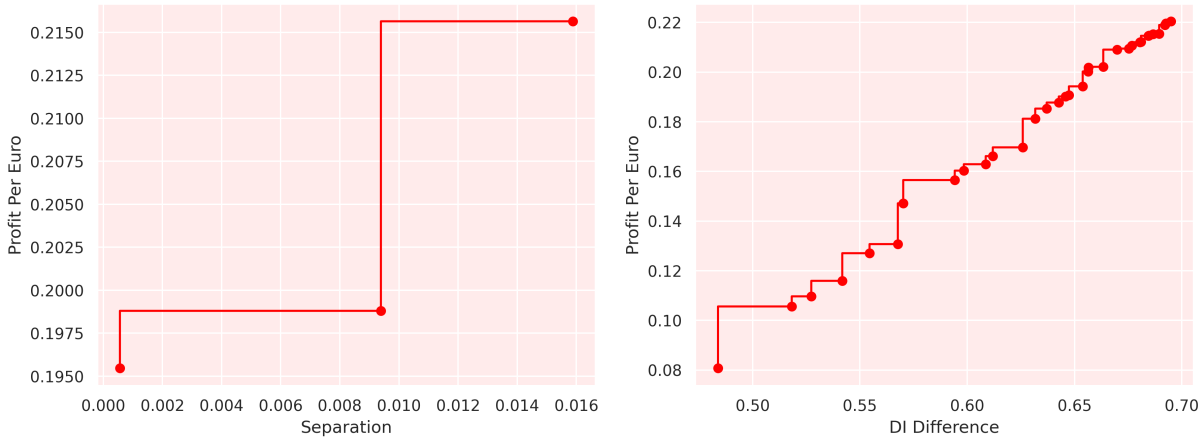

Figure 10: Pareto Frontiers: Profit vs. Fairness - Delayed Impact Optimization (Give Me Some Credit)

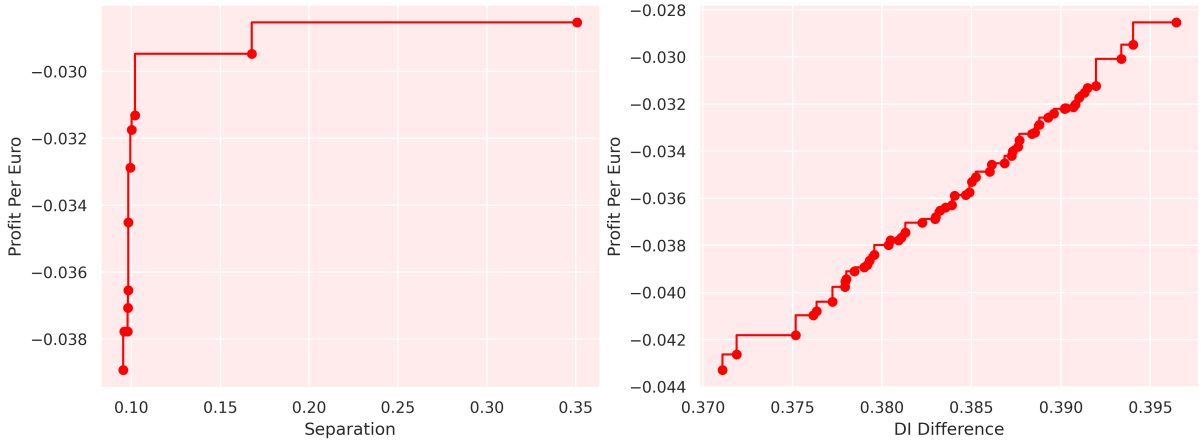

Figure 11: Pareto Frontiers: Profit vs. Fairness - Adversarial Debiasing (Give Me Some Credit)

Moving on to the delayed impact difference, one can observe an almost linear coherence between profit and delayed impact difference for both delayed impact optimization and adversarial debiasing. The former achieves a decrease in delayed impact difference of 0.225 at a cost of

€0.140 in marginal profit, the latter a decrease of 0.024 at a cost of €0.015, which is more effective, although the starting point was markedly lower.

While inspecting the profit calculation function above, several shortcomings of the profit estimation function became apparent that also manifest in the non-dominated solutions shown here. Since the main focus of this work is the business perspective and the changes in estimated profit can still be interpreted, profit was nevertheless determined to be the best way to analyze the connection between predictive performance and fairness in a credit scoring context.

To extend the available information and allow the comparison with another performance metric, similar Pareto frontiers are provided in appendix A. They depict the non-dominated solutions' fairness metrics and AUC instead of profit.

6 Conclusion

This paper presents a novel approach for a fairness processor that takes on the concept of delayed impact and sets out to maximize it for one of two sensitive groups, to improve the average long-term outcome in terms of fairness. As we have seen, the implementation of this makes some deviations from standard procedures inevitable, like the prior choice of one fixed cutoff value, which usually varies to allow a fixed acceptance rate.

The experiments that were carried out show that delayed impact optimization is capable of removing bias from predictions to the point of close to perfect fairness without simply accepting or declining everyone when it is applied to particularly biased simulated data. It did increase fairness in terms of standard fairness criteria like separation and disparate impact, substantially decreased the delayed impact difference between the two sensitive groups, and only moderately reduced fairness in terms of sufficiency. Considering that separation is likely to be a more suitable criterion than sufficiency in the context of credit scoring (Kozodoi et al., 2022), these results by themselves are promising. When applied to real-world data, delayed impact optimization demonstrated similar abilities, with the exception of a rising delayed impact difference for increasing penalty weights. When comparing its performance to that of adversarial debiasing, it became apparent that delayed impact optimization can remove bias from the predictions to a larger extent.

However, in doing so, it showed to be significantly less cost-effective, decreasing the predictive performance too much in exchange for the attained bias reduction to the point where it becomes prohibitively costly, rendering the models useless in terms of accomplishing business objectives. Furthermore, the experiments displayed that with larger penalty weights, which are necessary for serious fairness improvements, all results became increasingly volatile, varying so strongly between different weights that the consistency needed for business applications cannot be provided. Also, the initial goal of increasing the long-term fairness as measured by the delayed impact could not reliably be achieved. Summing up these findings, one must conclude

that delayed impact optimization, as proposed in this paper, is not a viable alternative to existing fairness processors. While it achieves a significant bias reduction, the loss in predictive performance that accompanies it is too high, outweighing the potential advantages. Further studies are necessary to determine whether this is a consequence of the implementation at hand since, throughout this study, several obstacles and limitations were described. Hence, it cannot be ruled out that solving them might yield more promising results.

Another interesting extension in future research on the topic, besides overhauling the residual shortcommings, would be to further extend the scope of the delayed impact measure by incorporating the framework into a multi-step feedback loop, that uses previous predictions as input for another iteration of the model training. This would enable an even more long-term-oriented depiction of discriminatory development. In essence, however, delayed impact optimization as presented here fell short of initial expectations and was demonstrated to be inferior to the established fairness processor.

References

Albemarle Paper Co. v. Moody, 422 US 405 (1975). https://supreme.justia.com/cases/federal/us/422/405/

Al-Heeti, A. (2019, January 4). *Amazon has sold more than 100 million Alexa devices*. CNET. https://www.cnet.com/home/smart-home/amazon-has-sold-more-than-100-million-alexa-devices/

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and machine learning*. https://fairmlbook.org/

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *Advances in Neural Information Processing Systems*, *29*. https://proceedings.neurips.cc/paper/2016/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html

Bureau of Consumer Financial Protection. (2019). *The Consumer Credit Card Market*. https://files.consumerfinance.gov/f/documents/cfpb_consumer-credit-card-market-report_2019.pdf

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, *356*(6334), 183–186. https://doi.org/10.1126/science.aal4230

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chouldechova, A., & Roth, A. (2018). The Frontiers of Fairness in Machine Learning. *arXiv e-prints*, Article arXiv:1810.08810.

Council Directive 2000/43/EC. (2000). http://data.europa.eu/eli/dir/2000/43/oj/eng

Darlington, R. B. (1971). Another Look at "Cultural Fairness". *Journal of Educational Measurement*, *8*(2), 71–82. https://doi.org/10.1111/j.1745-3984.1971.tb00908.x

De Scitovszky, T. (1943). A Note on Profit Maximisation and its Implications. *The Review of Economic Studies*, *11*(1), 57–60. https://doi.org/10.2307/2967520

Dimensions AI. (2022). Retrieved March 8, 2022, from https://www.dimensions.ai/

Dutta, S., Wei, D., Yueksel, H., Chen, P.-Y., Liu, S., & Varshney, K. (2020). Is There a Trade-Off Between Fairness and Accuracy? A Perspective Using Mismatched Hypothesis Testing. *Proceedings of the 37th International Conference on Machine Learning*, 2803–2813. https://proceedings.mlr.press/v119/dutta20a.html

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and Removing Disparate Impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259–268. https://doi.org/10.1145/2783258.2783311

Ferrer-i-Carbonell, A. (2005). Income and well-being: An empirical analysis of the comparison income effect. *Journal of Public Economics*, *89*(5), 997–1019. https://doi.org/10.1016/j.jpubeco.2004.06.003

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 329–338. https://doi.org/10.1145/3287560.3287589

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, *27*. https://papers.nips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

Griggs v. Duke Power Co., 401 US 424 (1975). https://supreme.justia.com/cases/federal/us/401/424/

Hardt, M., Price, E., Price, E., & Srebro, N. (2016). Equality of Opportunity in Supervised Learning. *Advances in Neural Information Processing Systems*, *29*. https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, *33*(1), 1–33. https://doi.org/10.1007/s10115-011-0463-8

Kamiran, F., Karim, A., & Zhang, X. (2012). Decision Theory for Discrimination-Aware Classification. *2012 IEEE 12th International Conference on Data Mining*, 924–929. https://doi.org/10.1109/ICDM.2012.45

Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2012). Fairness-Aware Classifier with Prejudice Remover Regularizer, 35–50. https://doi.org/10.1007/978-3-642-33486-3_3

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv e-prints*, Article arXiv:1412.6980.

Kozodoi, N., Jacob, J., & Lessmann, S. (2022). Fairness in credit scoring: Assessment, implementation and profit implications. *European Journal of Operational Research*, *297*(3), 1083–1094. https://doi.org/10.1016/j.ejor.2021.06.023

Kozodoi, N., Katsas, P., Lessmann, S., Moreira-Matias, L., & Papakonstantinou, K. (2020). Shallow Self-learning for Reject Inference in Credit Scoring. In U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, & C. Robardet (Eds.), *Machine Learning and Knowledge Discovery in Databases* (pp. 516–532). Springer International Publishing. https://doi.org/10.1007/978-3-030-46133-1_31

Larson, J., Surya, M., Kirchner, L., & Angwin, J. (2016, May 23). *How We Analyzed the COMPAS Recidivism Algorithm*. ProPublica. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm?token=XHXNgmPVImDaW15WrSaCeUz9xwCZGa7E

Liu, L. T., Dean, S., Rolf, E., Simchowitz, M., & Hardt, M. (2018). Delayed Impact of Fair Machine Learning. *International Conference on Machine Learning*, 3150–3158. https://proceedings.mlr.press/v80/liu18c.html

Lum, K., & Isaac, W. (2016). To predict and serve? *Significance*, *13*(5), 14–19. https://doi.org/10.1111/j.1740-9713.2016.00960.x

Musk, E. (2016, July 20). *Master Plan, Part Deux*. Tesla, Inc. https://www.tesla.com/de_ch/blog/master-plan-part-deux

Rajkomar, A., Hardt, M., Howell, M. D., Corrado, G., & Chin, M. H. (2018). Ensuring Fairness in Machine Learning to Advance Health Equity. *Annals of Internal Medicine*, *169*(12), 866–872. https://doi.org/10.7326/M18-1990

Roemer, J. E. (1993). A Pragmatic Theory of Responsibility for the Egalitarian Planner. *Philosophy & Public Affairs*, *22*(2), 146–166.

Roemer, J. E., & Trannoy, A. (2016). Equality of Opportunity: Theory and Measurement. *Journal of Economic Literature*, *54*(4), 1288–1332.

Smith, A. (1776). An inquiry into the nature and causes of the wealth of nations (cannan ed.), vol. 1. Methuen.

TransUnion LLC. (2021, August 18). *Healthy Consumer Credit Market Drives Return to Lending*. https://newsroom.transunion.com/healthy-consumer-credit-market-drives-return-to-lending/

Treaty establishing the European Community. (2002). http://data.europa.eu/eli/treaty/tec_2002/oj/eng

Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research*, *238*(2), 505–513. https://doi.org/10.1016/j.ejor.2014.04.001

Wakefield, J. (2016, March 24). *Microsoft chatbot is taught to swear on Twitter*. BBC. https://www.bbc.com/news/technology-35890188

Watson v. Fort Worth Bank & Trust, 487 US 977 (1988). https://supreme.justia.com/cases/federal/us/487/977/

Yoo, H., Zavala, V. M., & Lee, J. H. (2021). A Dynamic Penalty Function Approach for Constraint-Handling in Reinforcement Learning. *IFAC-PapersOnLine*, *54*(3), 487–491. https://doi.org/10.1016/j.ifacol.2021.08.289

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). Mitigating Unwanted Biases with Adversarial Learning. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 335–340. https://doi.org/10.1145/3278721.3278779

Appendices

A Alternative Loss Functions

The following graphs show the performance and fairness metrics as measured when delayed impact optimization with modified loss functions as depicted in functions 16, 17, and 18 in Section 3. These modified loss functions have been discarded because they offered an inferior combination of performance and stability compared to the implementation following Equation 15. Note, that no confidence intervals are depicted in the charts since the small number of observations combined with the large fluctuations in some metrics lead to wide confidence intervals that would compress the axes excessively.

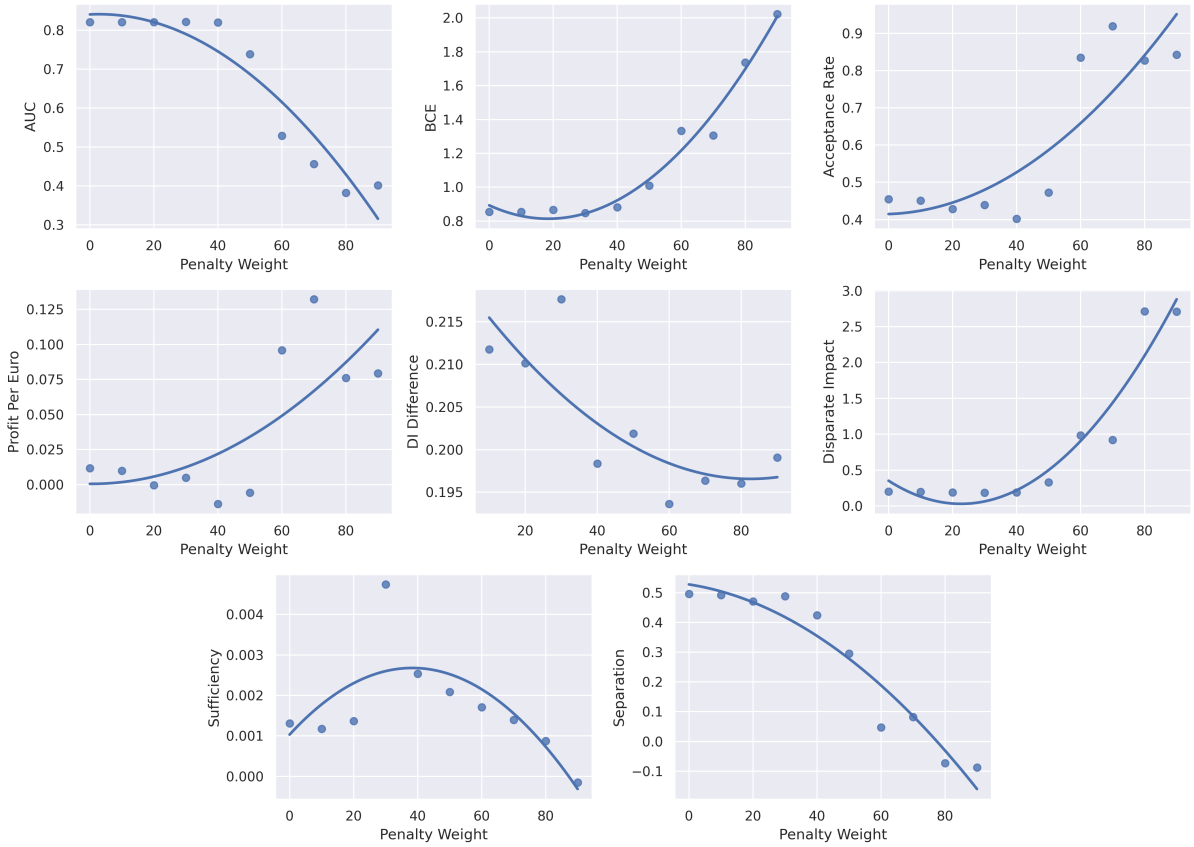

Figure 12: Performance and Fairness Metrics of the Implementation as presented in Function 16 (Simulated Data)

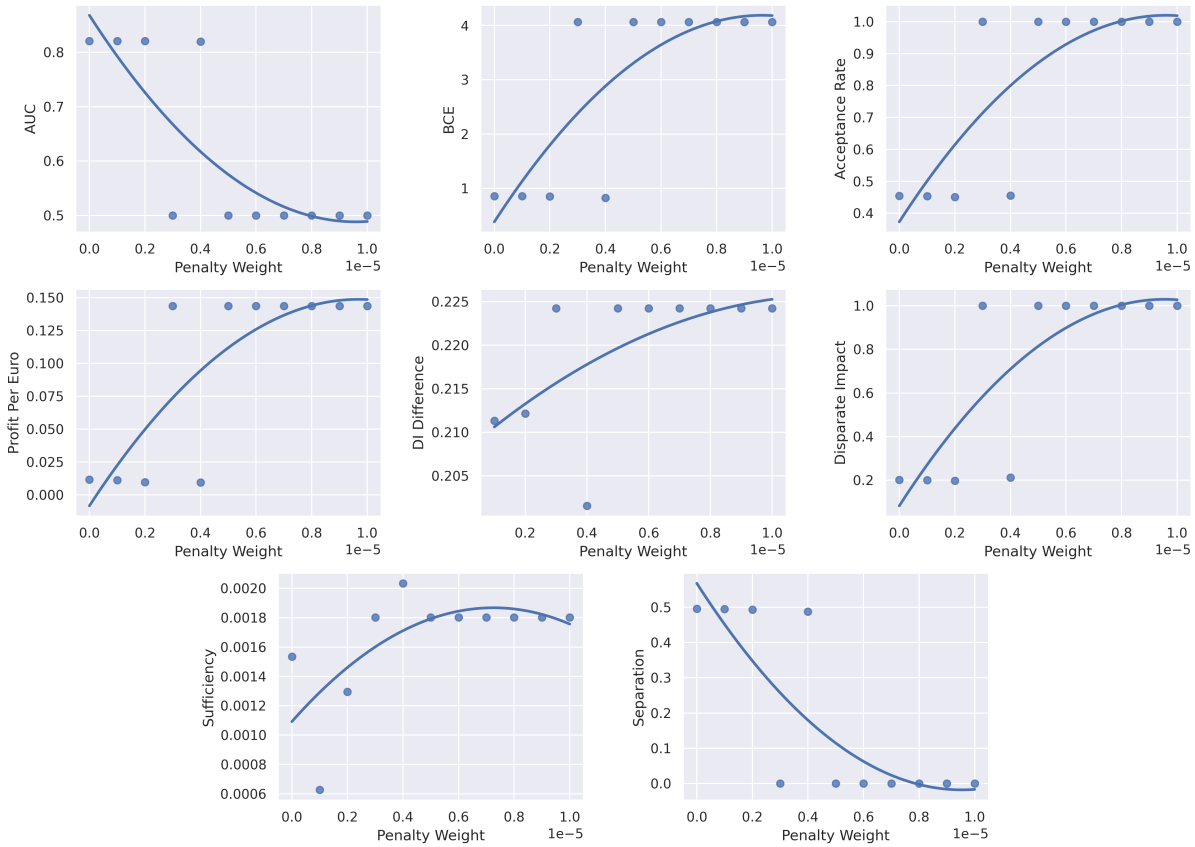

Figure 13: Performance and Fairness Metrics of the Implementation as presented in Function 17 (Simulated Data)

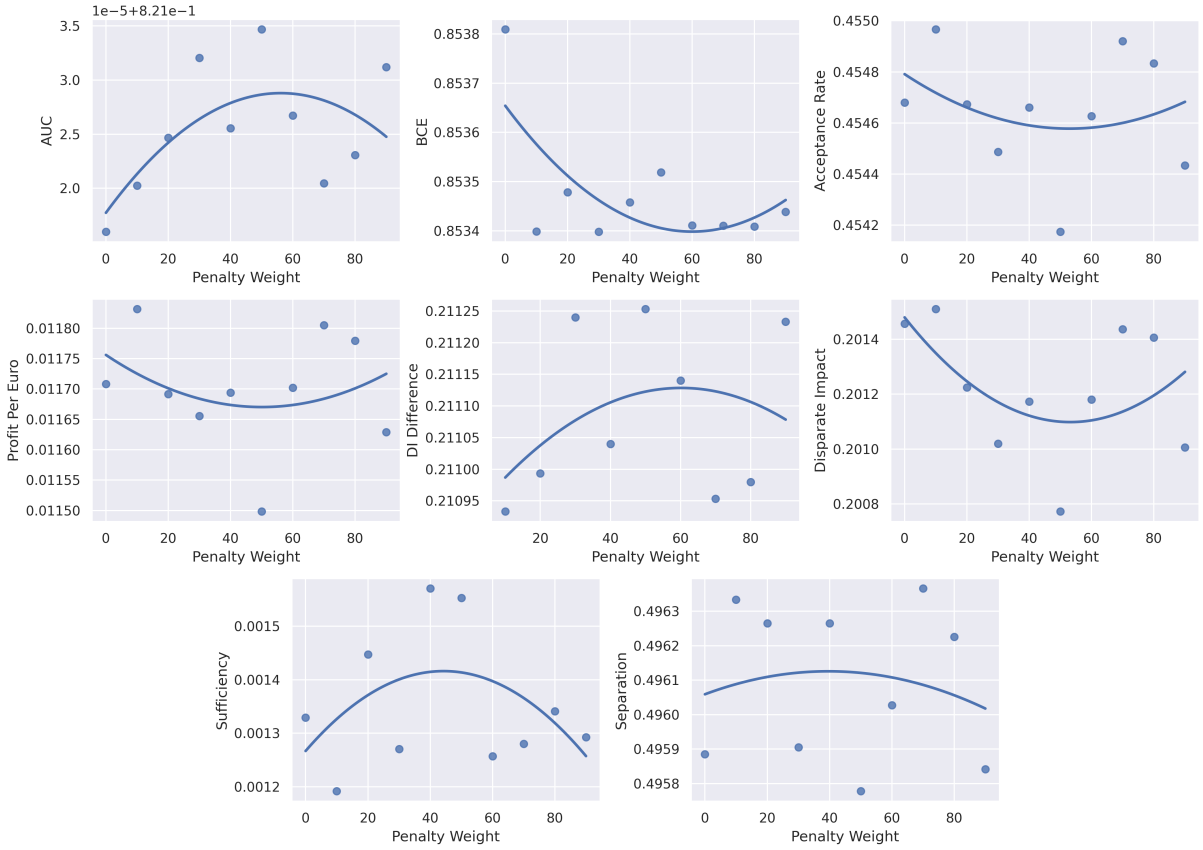

Figure 14: Performance and Fairness Metrics of the Implementation as presented in Function 18 (Simulated Data)

B Pareto Frontiers

These graphs show the AUC and the separation metric or delayed impact difference, respectively, of all non-dominated solutions for all data-processor-combinations. They once again show the large decrease in predictive performance one has to accept to obtain fairer predictions with delayed impact optimization, compared to adversarial debiasing, which might not come as close to perfect fairness but is significantly more cost-effective.

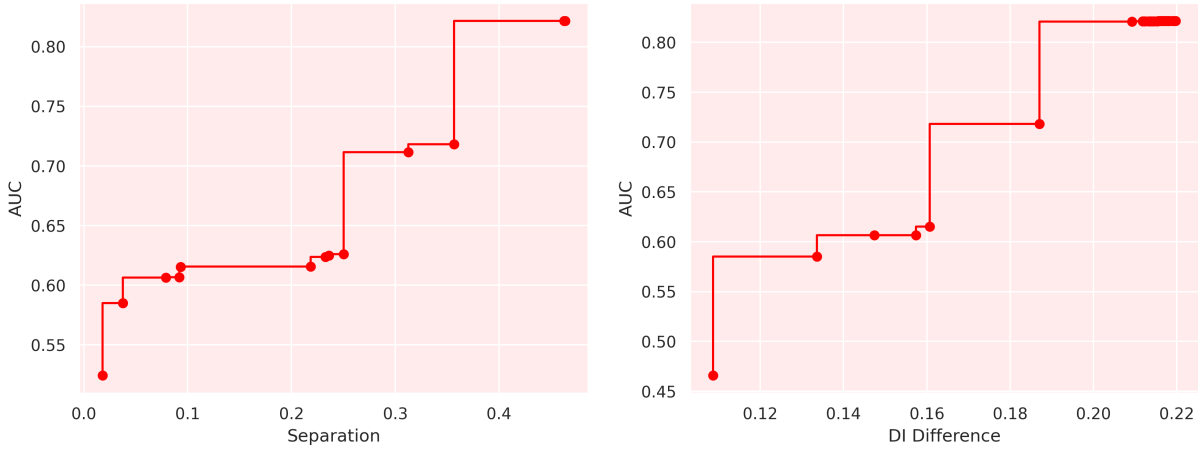

Figure 15: Pareto Frontiers: AUC vs. Fairness - Delayed Impact Optimization (Simulated Data)

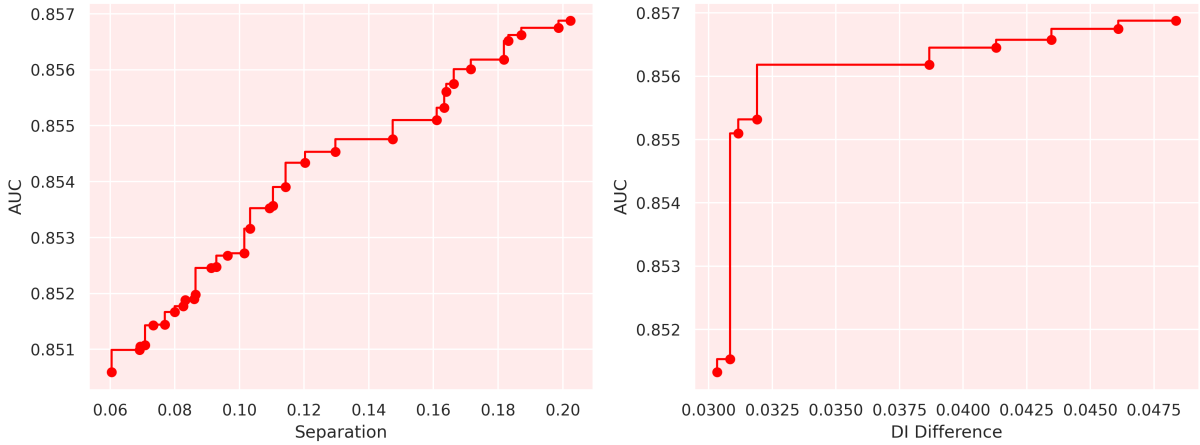

Figure 16: Pareto Frontiers: AUC vs. Fairness - Adversarial Debiasing (Simulated Data)

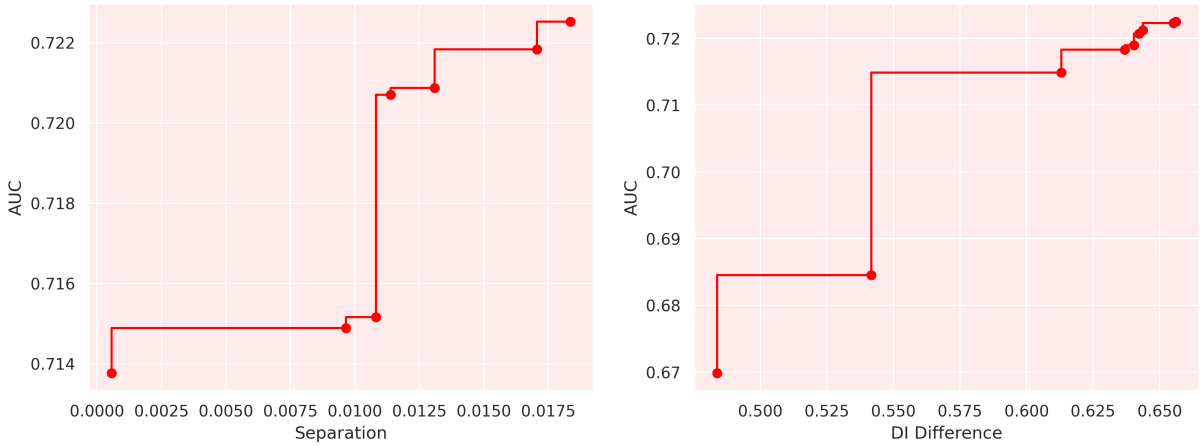

Figure 17: Pareto Frontiers: AUC vs. Fairness - Delayed Impact Optimization (Give Me Some Credit)

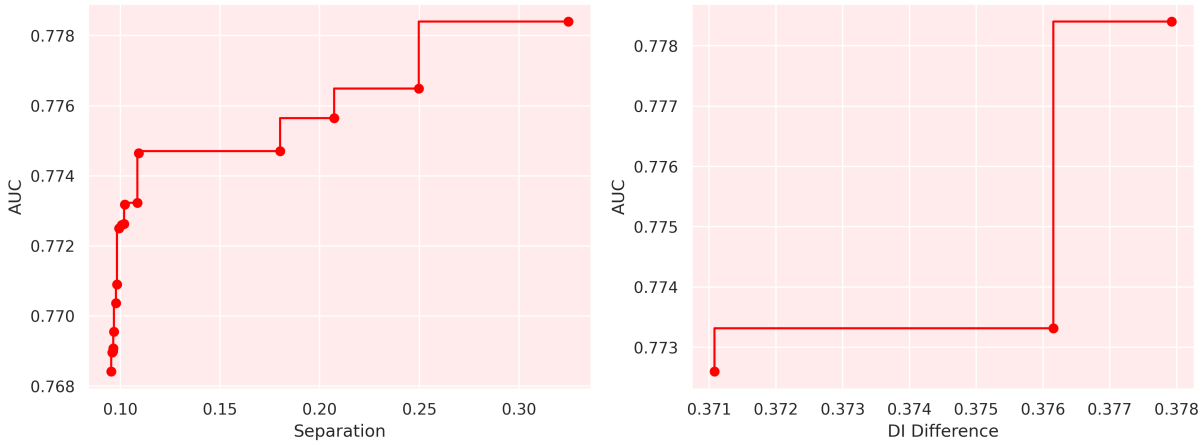

Figure 18: Pareto Frontiers: AUC vs. Fairness - Adversarial Debiasing (Give Me Some Credit)