FAIRNESS IN CREDIT SCORING: ASSESSMENT, IMPLEMENTATION AND PROFIT IMPLICATIONS

PREPRINT

Nikita Kozodoi[*]
Humboldt University of Berlin
Berlin, Germany

Johannes Jacob
Humboldt University of Berlin
Berlin, Germany

Stefan Lessmann
Humboldt University of Berlin
Berlin, Germany

ABSTRACT

The rise of algorithmic decision-making has spawned much research on fair machine learning (ML). Financial institutions use ML for building risk scorecards that support a range of credit-related decisions. Yet, the literature on fair ML in credit scoring is scarce. The paper makes two contributions. First, we provide a systematic overview of algorithmic options for incorporating fairness goals in the ML model development pipeline. In this scope, we also consolidate the space of statistical fairness criteria and examine their adequacy for credit scoring. Second, we perform an empirical study of different fairness processors in a profit-oriented credit scoring setup using seven real-world data sets. The empirical results substantiate the evaluation of fairness measures, identify more and less suitable options to implement fair credit scoring, and clarify the profit-fairness trade-off in lending decisions. Specifically, we find that multiple fairness criteria can be approximately satisfied at once and identify separation as a proper criterion for measuring the fairness of a scorecard. We also find fair in-processors to deliver a good balance between profit and fairness. More generally, we show that algorithmic discrimination can be reduced to a reasonable level at a relatively low cost.

***K*eywords** Credit scoring · Algorithmic fairness · Fair machine learning

1 Introduction

Financial institutions increasingly rely on machine learning (ML) to develop models that support decision-making processes. One of the prominent areas is retail credit scoring, where ML models guide loan approval decisions (Crook et al., 2007). Retail credit is an economically and socially important sector. In 2020, the total outstanding amount of retail credit in the US alone exceeded $4,161 billion[2]. ML-based scoring models, also known as scorecards, have played a major role in approving the corresponding credit applications. Their accuracy determines the risk exposure and financial health of lending institutions. Governing access to financing, scorecards also affect the economic well-being of retail clients.

In 2016, the Executive Office of the President of the US published a report on algorithmic systems, opportunity, and civil rights (Executive Office of the President, 2018). The report highlights the dangers of automated decision-making to the detriment of historically disadvantaged groups. It emphasizes credit scoring as a critical sector with a large societal impact, calling practitioners for using the principle of "equal opportunity by design" across different demographic groups. Similar actions were taken by the EU when they supplemented its General Data Protection Regulation with a guideline that stresses the need for regular and systemic monitoring

[*]E-mail: *nikita.kozodoi@hu-berlin.de*

[2]Source: `https://www.federalreserve.gov/releases/g19/current`

of the credit scoring sector (European Commission, 2017). The guidelines issued by the EU and the US reveal the political concern that potential violations of anti-discrimination law in credit scoring can have undesired economic effects for society by affecting debt and wealth distribution (Liu et al., 2018).

A growing body of literature on fair ML echos these concerns and proposes a range of statistical fairness measures and approaches for their optimization. It is common practice to discuss algorithmic fairness through the lens of differences between groups of individuals. The groups emerge from one or multiple categorical attributes that are considered sensitive. Examples of a sensitive attribute may include gender, religious denomination or ethnic group. The goal of fair ML is to ensure that ML model predictions (e.g., credit scores) meet statistical criteria, which aim at quantifying fairness. Narayanan (2018) distinguishes 21 such criteria. As Barocas et al. (2019) show, however, there are three fairness criteria, from which many other fairness measures can be derived: independence, separation, and sufficiency. Each of these criteria serves as a constraint in the optimization problem, which underlines the development of an ML model, and is implemented by fairness-enhancing algorithms denoted as fairness processors.

Surprisingly, the fair ML and credit scoring literature seem to share little touching points. We are aware of only three previous, related studies (Fuster et al., 2017; Hardt et al., 2016; Liu et al., 2018), and these focus on specific research questions rather than the broad scope of fair credit scoring. More specifically, no previous study offers a broad overview, systematization, and evaluation of fairness criteria and fairness processors for credit scoring. We argue that such a holistic perspective is crucial to help risk analysts stay abreast of recent developments in fair ML, judge their impact on credit scoring practices, and help to focus future research initiatives concerning fair credit scoring. This is what we strive to achieve in this paper.

The paper makes two contributions. First, we review and catalog state-of-the-art fairness processors across multiple important dimensions, such as the target fairness criterion, implementation method, and requirements for the classification problem. The catalog provides a systematic overview of fairness processors and clarifies whether and when these processors meet requirements associated with the application context of a scorecard and its implementation in loan approval processes. The catalog also addresses the critique of Mitchell et al. (2018), who demand a more uniform fairness terminology among scholars. Furthermore, we examine the extent to which the three established fairness criteria (and their implicit understanding of distributional equality) are appropriate for credit scoring. Given that different fairness criteria may conflict with one another (Chouldechova, 2017), our discussion is useful to inform the selection of a suitable fairness criterion or set of criteria in a credit scoring setting.

Second, we empirically compare a range of different fairness processors along several performance criteria using seven real-world credit scoring data sets. In contrast to the fair ML literature, which often examines model performance in terms of classification accuracy, we optimize and evaluate fairness processors in terms of their impact on the profit of the financial institution. Furthermore, we measure fairness not only with the criterion optimized by a processor but using different established fairness criteria. The experiment, therefore, extends the conceptual discussion on the suitability of the fairness criteria for credit scoring with a comprehensive empirical analysis of the agreement of the criteria and their correlation with profit. The comparison of fairness processors also aims at identifying techniques that best serve the interest of credit scoring practitioners (e.g., maximizing profit) and regulators (e.g., reducing discrimination of sensitive groups).

2 Related Work

This section reviews the related work on fair ML. First, we examine methods to integrate fairness constraints into the model development pipeline. Second, we provide theoretical background and intuition behind the established fairness criteria: independence, separation and sufficiency. We also show their strong connections to other fairness criteria suggested in previous studies.

2.1 Fairness Processors

Research on fair ML has recently emerged from the continuous integration of automated decision-making into important areas of social life and fairness concerns arising during this process (Barocas & Selbst, 2016). Much fair ML literature focuses on classification settings in which an unprivileged demographic group experiences discrimination through a classification model (Gajane & Pechenizkiy, 2017). Several attempts have been

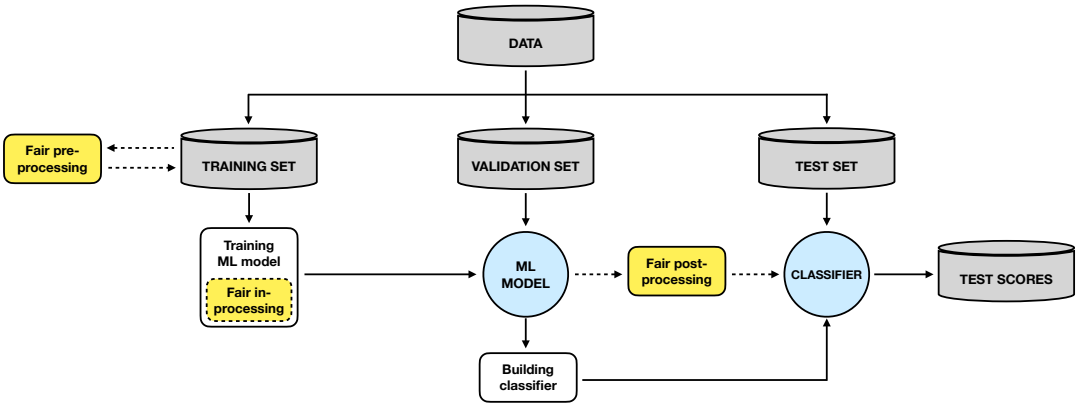

Figure 1: Fairness Integration in the ML Pipeline: In-processing, Pre-processing and Post-processing.

made to formalize the concept of fairness. Incorporating the corresponding fairness criteria in the ML pipeline facilitates measuring the degree to which class predictions discriminate against minorities (Barocas et al., 2019).

Algorithmic interventions designed to implement statistical fairness constraints are denoted as fairness processors. A processor can alter different stages in the ML pipeline. The literature distinguishes three methods of intervention: pre-processing, in-processing and post-processing (d'Alessandro et al., 2017). Their application generally depends on the conceptual and technical feasibility of a given prediction task. Figure 1 illustrates the fairness processors within an ML pipeline. We describe selected approaches from each group in Section 4.

Integrating a fairness processor into the pre-processing stage transforms the training data such that the input to a model is fair with respect to one or more sensitive features. Typically, fair pre-processing involves decorrelating the feature space with the sensitive attribute (e.g., Calmon et al., 2017). Even though modifying the training data is sometimes not possible or practical, the advantage of fair pre-processing is that if fairness is ensured before ML model training, it will also be ensured during the next model development steps (Barocas et al., 2019).

In-processing methods introduce auxiliary fairness constraints during ML model training. Then, training involves minimizing the empirical risk of the model while also optimizing a fairness criterion. In-processing renders a learned classifier (approximately) fair for the training data (Zafar et al., 2017a). Optimizing fairness during training has the potential to generate the highest utility as the tuning process also considers the fairness constraint. At the same time, in-processors are typically developed for settings with specific requirements (e.g., supporting only a single sensitive attribute), which limits their generality (Barocas et al., 2019). Another disadvantage is that implementing a fair in-processor requires full access to the training process and the input data. This is especially problematic in heavily regulated domains such as credit scoring, where changes to a risk model might require regulatory approval and are associated with high costs.

After an ML model is trained, post-processing can be applied to adjust the learned classifier or change its predictions according to the requirements of a particular fairness criterion (Hardt et al., 2016). The standard procedures include modifying the predicted scores or labels for specific observations. Unlike pre- or in-processing, post-processors need no information about the input data or the base model. This has the advantage that post-processors can be applied to any set of predictions. However, generality has a price. Post-processing is often less effective than alternative approaches and may substantially decrease classification accuracy (Barocas et al., 2019).

2.2 Fairness Criteria

This subsection introduces three established fairness criteria from a credit scoring perspective. Consider a setting in which a financial institution uses data on previous customers to predict whether a loan applicant will

default. Let $X \in \mathbb{R}^k$ denote the k features of a loan applicant and $y \in \{0, 1\}$ a random variable indicating if the applicant repays the loan ($y = 1$) or defaults ($y = 0$). The institution approves applications using a scoring model that predicts risk scores $s(X) = \mathbf{P}(y = 1|X)$. The score function can be turned into a classifier by accepting customers with scores above a cutoff τ. Let $x_a \in \{0, 1\}$ denote a protected attribute associated with certain characteristics of an applicant. For example, x_a could indicate whether she has a disability ($x_a = 1$) or not ($x_a = 0$). Clearly, the value of x_a must not impact the decision of the credit institution.

In the following, we consider a binary protected attribute to simplify the exposition. The discussed fairness criteria generalize to polynary protected attributes. Also, note that the fair ML literature often uses the terms protected attribute and sensitive attribute interchangeably. From a methodological perspective, it is less important whether the use of an attribute is socially undesirable or regulated by law. We use the term sensitive attribute throughout the paper while acknowledging that our example attribute disability is not only sensitive but protected.

2.2.1 Independence

The score $s(X)$ satisfies independence at a cutoff τ if the fraction of customers classified as good risks ($y = 1$) is the same in each sensitive group. Formally, this condition can be written as:

$$\mathbf{P}\left[s(X \mid x_a = 0) > \tau\right] = \mathbf{P}\left[s(X \mid x_a = 1) > \tau\right] \tag{1}$$

Equation (1) states that $s(X)$ is statistically independent of the sensitive attribute x_a (Barocas et al., 2019). Classifier predictions are not affected by the sensitive attribute, and the probability to be classified as a good risk is the same in both groups (Pleiss et al., 2017).

This strict constraint is usually not feasible for real-world applications like credit scoring, as the resulting loss in model performance can make a business unsustainable. Therefore, it is a common practice in anti-discrimination law to allow the score and the sensitive attribute to share at least some mutual information and introduce a relaxation of the independence criterion (Barocas & Selbst, 2016). The Equal Opportunity Credit Act has a regulation that is referred to as the "80 percent rule" (Feldman et al., 2015). The rule requires that $\mathbf{P}(s(X \mid x_a = 1) > \tau) \leq 0.8 \cdot \mathbf{P}(s(X \mid x_a = 0) > \tau)$, where $\{x_a = 0\}$ is the privileged group (Kleinberg et al., 2016).

This paper measures independence using a metric denoted as IND, which we define as:

$$\text{IND} = \left|\mathbf{P}\left[s(X \mid x_a = 0) > \tau\right] - \mathbf{P}\left[s(X \mid x_a = 1) > \tau\right]\right| \tag{2}$$

A positive difference between the two fractions implies that the group $\{x_a = 0\}$ is considered the privileged group and vice versa. The closer independence is to zero, the lower is the discrimination.

2.2.2 Separation

The separation criterion is satisfied if the classification based on the score $s(X)$ and the cutoff τ is independent on x_a conditional on the true outcome y (Barocas et al., 2019). Formally, the score $s(X)$ satisfies separation at a cutoff τ if the following constraints hold:

$$\begin{cases} \mathbf{P}\left[s(X \mid y = 0, x_a = 0) > \tau\right] = \mathbf{P}\left[s(X \mid y = 0, x_a = 1) > \tau\right] \\ \mathbf{P}\left[s(X \mid y = 1, x_a = 0) \leq \tau\right] = \mathbf{P}\left[s(X \mid y = 1, x_a = 1) \leq \tau\right] \end{cases} \tag{3}$$

The expression in the first line compares the false positive rate (FPR) across the sensitive groups, whereas the second line compares the false negative rate (FNR) per group. The separation criterion, therefore, requires that the FNR and the FPR are the same for the sensitive groups.

Separation acknowledges that x_a may be correlated with y (e.g., applicants with a disability might can a higher default rate). However, the criterion prohibits the use of x_a as a direct predictor for y. When the difference between group sizes is large, the criterion will punish models that perform well only on the majority group (Hardt et al., 2016). We measure separation with a metric denoted as SP that captures the average group difference in the FPR and FNR:

$$\text{SP} = \frac{1}{2}\left|\left(\text{FPR}_{\{x_a=1\}} - \text{FPR}_{\{x_a=0\}}\right) + \left(\text{FNR}_{\{x_a=1\}} - \text{FNR}_{\{x_a=0\}}\right)\right| \tag{4}$$

A positive difference between the group-wise error rates indicates that the $\{x_a = 0\}$ group has a lower misclassification rate and is, therefore, the privileged group. Similar to the independence criterion, perfect separation (i.e., SP = 0) is observed when the group-wise error rates are equal. Higher values of the metric indicate stronger discrimination.

2.2.3 Sufficiency

The score $s(X)$ is sufficient at a cutoff τ if the likelihood that an individual belonging to a positive class is classified as positive is the same for both sensitive groups (Barocas et al., 2019). This implies that for all values of $s(X)$ the following condition holds:

$$\mathbf{P}(y = 1 \mid s(X) > \tau, x_a = 0) = \mathbf{P}(y = 1 \mid s(X) > \tau, x_a = 1) \tag{5}$$

Equation (5) requires that the positive predictive value (PPV) is the same for the sensitive groups (Chouldechova, 2017). This paper defines the sufficiency metric SF as the absolute difference between the group-wise PPV:

$$\text{SF} = \left|\text{PPV}_{\{x_a=0\}} - \text{PPV}_{\{x_a=1\}}\right| \tag{6}$$

2.2.4 Further Criteria

The three fairness criteria introduced above comprise a number of other fairness concepts, which have been proposed in prior work. Many scholars only use different terminology or a slightly varying statistical formulation of the same fairness concepts (Barocas et al., 2019; Mitchell et al., 2018). Table 1 illustrates how independence, separation and sufficiency have been referred to in the literature and how they relate to the other formulations.

Table 1 reveals that the statistical formulation of fairness constraints originates from the field of psychological testing (Darlington, 1971) and has been rediscovered for ML applications much later. The 19 fairness concepts presented in the table can be derived from independence, separation and sufficiency, which underpins their relevance and justifies our focus on these criteria in the paper.

Table 1 embodies the concept of group-based fairness. Other fairness concepts exist and include individual and counterfactual fairness. Individual fairness implies that a fair classifier generates similar outputs for similar individuals, where similarity is estimated using a distance metric (Dwork et al., 2012). The idea behind counterfactual fairness is to ensure that a classifier output remains the same when the sensitive attribute is changed to its counterfactual value (Kusner et al., 2017).

3 Fairness and Credit Scoring

The section discusses fair ML in the context of credit scoring. We summarize previous work on fair credit scoring, systematically review and catalog existing fairness processors across different dimensions and discuss their applicability in credit scoring. We also touch on the appropriateness of fairness criteria for credit scoring. The considerations provided in this section contribute to the literature on fair credit scoring by examining the implications of different fairness integration methods for decision-makers and retail clients.

3.1 Fair Credit Scoring

To the best of our knowledge, only three papers focus on algorithmic fairness in credit scoring. Liu et al. (2018) argue that achieving fairness in credit scoring must be formalized as a long-term goal. One of the common challenges in credit scoring is sample selection bias that arises from training a classifier on previously accepted cases (Banasik et al., 2003). This usually leads to poor generalization. Scoring models tend to overestimate the creditworthiness of certain groups of applicants. The credit scoring literature discusses sample bias in the scope of reject inference and examines its implications for predictive accuracy or profit (Kozodoi et al., 2019). Pointing out that sample bias also perpetuates existing unfairness, Liu et al. (2018) call for mathematical constraints that optimize a long-term societal goal rather than applying traditional static fairness interventions. However, the formulation of these constraints is still subject to further research and not yet a practical solution

Table 1: Fairness Criteria and their Relation to Independence, Separation, and Sufficiency

Reference	Criterion	Closest relative	Relation
Darlington (1971)	Darlington criterion (4)	Independence	Equivalent
Dwork et al. (2012)	Statistical parity	Independence	Equivalent
Dwork et al. (2012)	Group fairness	Independence	Equivalent
Dwork et al. (2012)	Demographic parity	Independence	Equivalent
Corbett-Davies et al. (2017)	Conditional statistical parity	Independence	Relaxation
Darlington (1971)	Darlington criterion (3)	Separation	Relaxation
Hardt et al. (2016)	Equal opportunity	Separation	Relaxation
Hardt et al. (2016)	Equalized odds	Separation	Equivalent
Kleinberg et al. (2016)	Balance for the negative class	Separation	Relaxation
Kleinberg et al. (2016)	Balance for the positive class	Separation	Relaxation
Zafar et al. (2017a)	Avoiding disparate mistreatment	Separation	Equivalent
Chouldechova (2017)	Predictive equality	Separation	Relaxation
Woodworth et al. (2017)	Equalized correlations	Separation	Relaxation
Berk et al. (2018)	Conditional procedure accuracy	Separation	Equivalent
Cleary (1968)	Cleary model	Sufficiency	Equivalent
Darlington (1971)	Darlington criterion (1), (2)	Sufficiency	Relaxation
Chouldechova (2017)	Predictive parity	Sufficiency	Relaxation
Chouldechova (2017)	Calibration within groups	Sufficiency	Equivalent
Berk et al. (2018)	Conditional use accuracy	Sufficiency	Equivalent

for practitioners. Therefore, the main body of the fairness literature and this study is concerned with static fairness interventions.

Hardt et al. (2016) propose the equalized odds fairness criterion and develop an algorithm that adjusts classifier predictions in a way that is fair according to this criterion. They demonstrate how their method improves fairness compared to a maximum profit benchmark using a credit scoring example based on FICO scores. Focusing on the introduction of a new method, the study does not examine the trade-off between profit and fairness and provides limited empirical evidence on how equalized odds compare to other fairness criteria or how fairness is best ensured in an ML pipeline.

Last, Fuster et al. (2017) also consider a static scenario in which they create a setting in which the introduction of ML to the credit market is formalized as an intervention. Based on this perspective, they analyze the effect of ML on interest rates in demographically different groups and claim that ML, as such, can be considered an unfair intervention in the market. While this is a novel approach to examining demographic discrimination in a domain, it has no direct implications for operational loan approval decisions or management decisions concerning approval strategy.

In summary, the main distinction between the focal paper and the previous studies on fairness in credit scoring is that we undertake a comprehensive empirical analysis of alternative fairness criteria and fairness processors, which optimize these criteria. Prior work fails to account for the breadth of approaches that have been proposed in the scope of fair ML. Also, to our best knowledge, no previous study examines the interplay between fairness criteria and processors. Therefore, we aim at consolidating different advancements in fair ML, discussing their suitability for credit scoring, and providing rich empirical results that clarify the degree to which fairness constraints affect the predictive ability of credit scorecards and the corresponding profit implications, and how the trade-off between fairness and profit develops across fairness criteria and processors. We hope that our results offer actionable insights on how to set and pursue fairness objectives in credit scoring.

3.2 Cataloging Fairness Processors

The fair ML literature has developed a variety of fairness processors to implement independence, separation and sufficiency constraints. The complexity between these processors varies considerably, from simply relabeling the prediction outcomes (e.g., Kamiran et al., 2012) to complex deep learning approaches for training a discrimination-free classifier (e.g., Zhang et al., 2018). Furthermore, some processors are limited

to specific problem setups. This motivates us to develop a structured overview of fairness processors with respect to their characteristics and applicability. Specifically, we catalog existing fairness processors in Table 2 using six dimensions: (i) point of intervention into the ML pipeline; (ii) optimized fairness criterion; (iii) classification problem type supported by a processor (binary or polynary); (iv) possible number of sensitive attributes (one or multiple); and (vi) supported types of sensitive attributes (binary or polynary).

Three main conclusions emerge from Table 2. First, the majority of processors implement the independence criterion. This can be explained by the fact that the other two criteria have only recently been established in the fair ML literature (see Table 1 for comparison). Furthermore, independence allows implementation via pre-processing, which provides an additional point of intervention in the ML pipeline. In many scenarios, however, fairness through independence may not be a suitable choice. This calls for additional processors that implement the other two criteria.

Second, the choice of a suitable fairness processor is limited by the application and implementation context of a scorecard. The application context determines the type of target variable and sensitive attribute(s) to be handled by a processor. For instance, in a setup with multiple sensitive attributes optimizing separation is only possible via the adversarial debiasing or reject option classification. This is a severe limitation for credit scoring because financial institutions commonly face several protected attributes: the U.S. anti-discrimination law distinguishes nine bases that must not influence lending decisions including race, color, religion and other customer attributes (Equal Credit Opportunity Act, 1974). The implementation context can also limit possible points of intervention in the ML pipeline. Replacing a scorecard with a fair in-processor might require regulatory approval and incur additional costs. Post-processors are easier to implement since they are agnostic of the input data and the scorecard, and only require access to the predicted scores.

Third, it is a standard procedure to embed the fairness processor into an accuracy-optimizing framework. The loss in predictive accuracy is commonly used as a performance measure to judge the cost of integrating a fairness constraint. In line with this framework, Friedler et al. (2019) conducted a comparative study to examine the achieved fairness and accuracy of four fairness processors. However, recent credit scoring literature criticizes the practice of using standard performance measures for evaluating scoring models and calls for profit-driven evaluation (Verbraken et al., 2014). In such a setup, evaluation of fairness processors should be performed with a profit maximization objective instead of standard statistical performance measures such as accuracy.

To conclude, the catalog suggests that a comparative analysis of fairness processors under profit maximization is needed to clarify the "cost of fairness". We argue that the profitability aspect is underrepresented in the fair ML literature, while it is highly relevant for real-world applications. A better understanding of the (dis)agreement of profitability and different fairness criteria is also useful for policy making as it sheds some light on the thorny question which criterion lending institutions should emphasize. Which fairness processor to use for optimizing the desired criterion is yet another question with high relevance for practice. Prior literature offers limited guidance due to assessing processors typically only in terms of the single criterion that this processor implements. Contributing toward answering these pressing questions is the overall goal of the paper.

3.3 Fairness Criteria for Credit Scoring

The choice of the fairness criterion has severe consequences on the social impact of lending decisions (Liu et al., 2018). The adequacy of a fairness criterion depends on the societal fairness goal for a specific domain. Consequently, the definition of this societal goal is crucial.

An unconstrained profit-oriented scoring model will take full advantage of the available (sensitive) information and discriminate between protected groups if it enhances the predictive performance. The purpose of introducing fairness is, therefore, to constrain the profit of a financial institution for a better, discrimination-free outcome. According to the U.S. anti-discrimination law, demographic properties of a loan applicant should not influence lending decisions (Equal Credit Opportunity Act, 1974). The purpose of this law is to ensure fair access to loans irrespective of a person's demographic characteristics. Arguably, the overall societal goal behind such a law is an equal opportunity for financial well-being across demographically different groups. Achieving this goal in credit scoring is particularly difficult as clients face unequal misclassification costs. Applicants who are denied a loan they could have repaid face the cost of a missed opportunity to enhance their social and economic position. However, if applicants receive a loan they cannot repay, they are confronted

Table 2: Fairness Processors

Fairness processor	Reference	Method	Criterion	PT	PS	MS	PE	This paper
Reweighing	Calders et al. (2009)	PRE	IND					✓
Massaging	Calders et al. (2009)	PRE	IND					
Classification without discrimination	Kamiran & Calders (2009)	PRE	IND					
Discrimination discovery K-NN	Luong et al. (2011)	PRE	IND	✓				
Fair representation learning	Zemel et al. (2013)	PRE	IND	✓				
Disparate impact remover	Feldman et al. (2015)	PRE	IND		✓	✓		✓
Variational fair autoencoder	Louizos et al. (2016)	PRE	IND	✓	✓	✓		
Feature adjustment	Johndrow et al. (2019)	PRE	IND	✓	✓	✓		
Discrimination-free pre-processing	Calmon et al. (2017)	PRE	IND		✓	✓		
Prejudice remover regularizer	Kamishima et al. (2012)	IN	IND	✓				✓
Fair accuracy maximizer	Zafar et al. (2017b)	IN	IND	✓	✓	✓		
Non-discriminatory Learner	Woodworth et al. (2017)	IN	SP					
Adversarial debiasing	Zhang et al. (2018)	IN	SP	✓	✓	✓		✓
Meta-fairness algorithm	Celis et al. (2019)	IN	IND, SP, SF		✓	✓		✓
Group-wise Platt scaling	Platt (1999), Barocas et al. (2019)	POST	SF	✓	✓	✓		✓
Group-wise histogram binning	Zadrozny & Elkan (2001)	POST	SF	✓	✓	✓		
Group-wise isotonic regression	Niculescu-Mizil & Caruana (2005)	POST	SF	✓	✓	✓		
Fairness-aware classifier	Calders & Verwer (2010)	POST	IND					
Reject option classification	Kamiran et al. (2012)	POST	IND, SP		✓	✓		✓
Fairness constraint optimizer	Goh et al. (2016)	POST	IND	✓	✓	✓		
Equalized odds processor	Hardt et al. (2016)	POST	SP		✓		✓	✓
Calibrated equalized odds	Pleiss et al. (2017)	POST	SP					

Abbreviations: IND = Independence, SP = separation, SF = sufficiency; PRE = pre-processor, IN = in-processor, POST = post-processor; PT = polynary target, PS = polynary sensitive attribute, MS = multiple sensitive attributes, PE = profit-driven evaluation.

with financial debt and a long-term worsening of their financial situation as future access to loans will be more difficult. With these specific properties of the credit scoring sector in mind, the following considerations elaborate on the extent to which independence, separation and sufficiency fulfill the goal of equal opportunity for financial well-being in society.

Forcing independence on a scoring model results in the same rate of accepted customers within sensitive groups. The problem with this approach is that the ability to repay a loan can have a different distribution in each group (Barocas et al., 2019). If this is the case, but members of both groups have the same probability of receiving a loan, one group will experience more actual defaults. As mentioned before, the consequences of defaulting can be more severe for a customer than the opportunity costs associated with a rejected application. Typically, the historically unprivileged group has a higher rate of non-solvent customers. Handing out loans to such individuals might worsen their financial situation in the long term (Hardt et al., 2016). Instead of achieving fairness, this can lead to further perpetuating the existing unfair pattern. The goal of better financial equality would not be met, and the financial gap in society could become even wider.

The separation criterion addresses this dilemma and acknowledges that a sensitive attribute might correlate with the default rate. By requiring the same error rates between groups but allowing different positive classification rates, separation achieves a fair result that is not only closer to the reality of credit allocation decisions but also more desirable from the perspective of a customer. More precisely, the separation criterion accounts for different misclassification costs between groups. On the contrary, separation would be inadequate if credit scoring would have a strictly preferred outcome for a customer as it is the case in domains like college admission (Mitchell et al., 2018). Interestingly, the first formulation of the separation criterion in the context of ML by (Hardt et al., 2016) is based on the example of the credit scoring domain and the limitations of the independence criterion to meet its requirements.

Sufficiency requires the ratio of true positive classifications over all positive classifications to be the same for the sensitive groups. This concept has two disadvantages for credit scoring. First, it allows for substantial discrimination in separation. For both groups, the proportion of correctly labeled non-default customers can be the same, satisfying sufficiency. In contrast, the likelihood of a potential non-default customer to be classified as bad risk can still differ between groups, violating the separation constraint. Second, most ML algorithms are designed to achieve sufficiency without integrating a fairness constraint if the model can predict the sensitive attribute from the other features (Barocas et al., 2019). In credit scoring, the question would, therefore, be if the current procedure for assessing a customer's default risk and the associated distribution of loans is fair. The literature suggests that the answer to this question is no (Fuster et al., 2017; Liu et al., 2018; Hardt et al., 2016). Hence, sufficiency appears to be a less appropriate criterion for credit scoring. Based on these considerations, the separation criterion appears most suitable to achieve a desirable form of fairness in credit scoring. Separation accounts for the imbalanced misclassification costs of the customer, and, as these imbalanced costs also exist for the financial institution, separation is also able to consider the interests of the loan market.

The considerations provided in this section suggest that the question of which fairness constraint is most adequate for credit scoring should be a part of a wider academic and societal debate. Such a democratic process should also acknowledge the importance of studying the long-term effects of implementing different fairness constraints to judge whether the societal goal of better financial equality between demographic groups can be achieved with specific interventions (Liu et al., 2018).

4 Methodology

This section describes eight fairness processors that are part of the empirical study. The selection of processors covers all combinations of fairness interventions. Following the setup introduced in Section 2, we consider a credit scoring setting with a binary target variable $y \in \{0, 1\}$ and a binary sensitive attribute $x_a \in \{0, 1\}$ to introduce the processors. Some processors also generalize to setups with a polynary target variable and polynary sensitive attribute (see Table 2 for details).

4.1 Pre-Processors

Fairness pre-processors transform the input data to achieve fairness. Reweighing is a pre-processor that assigns weights to each observation in the training set based on the overall probabilities of the group-class combinations (Calders et al., 2009). Thus, weights for observations with $(x_a = 1, y = 1)$ are greater than weights for observations with $(x_a = 0, y = 1)$ if members of the group $\{x_a = 1\}$ have a lower probability to belong to a positive class than those of the group $\{x_a = 0\}$:

$$W(X \mid x_a = 1,\ y = 1) = \frac{\mathbf{P}_{exp}(x_a = 1 \mid y = 1)}{\mathbf{P}_{obs}\ (x_a = 1 \mid y = 1)}\ , \tag{7}$$

where $\mathbf{P}_{exp}$ is the expected probability and $\mathbf{P}_{obs}$ is the observed probability. For instance, assume that 90% of all individuals belong to the positive class and 20% percent belong to the group $\{x_a = 1\}$. Then, $\mathbf{P}_{exp}(x_a = 1 \mid y = 1) = 0.9 \cdot 0.2 = 0.18$. If, in fact, only 12% of all cases in $\{x_a = 1\}$ belong to the positive class, then $W\ (X \mid x_a = 1, y = 1) = \frac{0.18}{0.12} = 0.9$.

Based on the computed weights, a fair training set is resampled with replacement such that combinations with a higher weight reappear more often. This procedure helps to fulfill the independence criterion. A discrimination-free classifier can then be trained on the resampled data.

Another pre-processing technique is the disparate impact remover proposed by Feldman et al. (2015). The intuition behind this processor is to ensure independence by prohibiting the possibility of predicting the sensitive attribute x_a with the other features in X and the outcome y. This is achieved by transforming X into $\overline{X}$ while preserving the rank of X within sensitive groups defined by x_a. By preserving the rank of X given x_a, the classification model $f(\overline{X})$ will still learn to choose higher-ranked credit applications over lower-ranked ones based on the other features.

The transformation is performed using an interpolation based on a quantile function and the cumulative distribution of $F : \mathbf{P}(X \mid x_a = a)$. This ensures that given the transformed $\overline{X}$ at some rank, the probability of drawing an observation given $x_a = a$ is the same as for the entire data set. Hence, x_a cannot be predicted with the other attributes, and the independence criterion is fulfilled. Since ensuring perfect independence can have a strong negative impact on a classifier utility, the transformation can be modified to only partially remove disparate impact. The meta-parameter $\lambda \in [0, 1]$ allows controlling the desired level of fairness-utility trade-off during transformation.

4.2 In-Processors

In-processors achieve fairness when building a classifier. One of such methods, prejudice remover, introduces a fairness-driven regularization term to the classification model (Kamishima et al., 2012). Regularization is a standard statistical approach to penalize a model for some undesired behavior. This is typically done by adding a regularizer term to the loss function.

The fairness-driven regularization introduced by Kamishima et al. (2012) is based on the prejudice index PI, which quantifies the degree of unfairness based on the independence criterion:

$$\text{PI} = \sum_{y, x_a} \mathbf{P}(y, x_a) \ln \frac{\mathbf{P}(y, x_a)}{\mathbf{P}(x_a)\mathbf{P}(y)} \tag{8}$$

PI measures the amount of mutual information between y and x_a. High values of PI indicate that a sensitive attribute x_a is a good predictor for y. The optimization problem extends to:

$$\min_f L\left[f\left(X\right), y\right] + \eta \text{PI}\ , \tag{9}$$

where $L(\cdot)$ is the underlying loss function of the model $f(X)$, and η controls the importance of the term PI. In this study, we tune η to maximize the profitability of a scorecard. The regularization term ensures that the sensitive attribute x_a becomes less influential in the final prediction.

Adversarial debiasing is another in-processor that stacks two neural networks with contrary objectives on top of each other (Zhang et al., 2018). The first network (predictor) is trying to learn a function to predict y given

X, while also minimizing the success of the second network. The second network (adversary) takes the output layer of the first model $\hat{y}$ and the true labels y as input and tries to predict the sensitive attribute x_a. Both models have objective-specific loss functions and weights that are optimized using stochastic gradient descent.

The weights of the adversary denoted as U are updated based on the gradient that minimizes its loss function, $\nabla_U L(\hat{x}_a, x_a)$. The weights of the predictor denoted as W are propagated based on a gradient that minimizes its loss function $L(\hat{y}, y)$ but also maximizes the loss function of the adversary, such that $\nabla_W L(\hat{y}, y) - \alpha \nabla_W L(\hat{x}_a, x_a)$, where α is a meta-parameter.

Since the adversary takes the output of the predictor $\hat{y}$ as input, the predictor aims to hold back any additional information about the sensitive attribute x_a in its output $\hat{y}$ as it would improve the adversary's loss. In other words, the predictor will try to deceive the adversary and not share any additional information in $\hat{y}$. As y is known to the adversary, the algorithm acknowledges that the sensitive attribute might correlate with y, and only unnecessary information will be avoided. Hence, the adversarially debiased model will converge towards the separation criterion.

The meta fair classification algorithm is yet another in-processor designed to achieve fairness according to a number of different fairness criteria. The idea is that each criterion has a fairness metric FM, which measures the equality (or discrimination) between groups. If the metric is similar across groups, the level of fairness is high. Based on this approach, Celis et al. (2019) add a secondary condition to the optimization problem underlying classifier training $f(X)$:

$$\min_f L(f(X), y) \quad \text{s.t.} \quad \frac{\min\left[\text{FM}(f(X \mid x_a = 0)),\ \text{FM}(f(X \mid x_a = 1))\right]}{\max\left[\text{FM}(f(X \mid x_a = 0)),\ \text{FM}(f(X \mid x_a = 1))\right]} \geq \sigma \tag{10}$$

where $\sigma \in [0, 1]$ is the fairness bound. In case of sufficiency, FM is the predictive parity (PPV) with $\text{FM}(f) = \text{PPV}(f) = \frac{\mathbf{P}(f=1 \mid x_a=a, y=1)}{\mathbf{P}(f=1 \mid x_a=a)}$. If the group $\{x_a = 1\}$ has a low PPV and the group $\{x_a = 0\}$ has a high PPV, the fraction in the secondary condition is close to zero. A high σ will, therefore, bound the classifier to a high degree of fairness. During training, the value for σ can be tuned such that it maximizes profit, while minimizing the loss in fairness, i.e., the loss in sufficiency.

4.3 Post-Processors

As a post-processing method, reject option classification is based on the output of a learned classifier (Kamiran et al., 2012). In a credit scoring setup, the classifier output is a credit score that reflects the posterior probability to not default for each customer $s(X) = \mathbf{P}(\hat{y} = 1|X)$. The closer the score is to 1 or 0, the higher is the certainty with which the classifier assigns the corresponding labels, whereas a score close to 0.5 implies a high degree of uncertainty.

Reject option classification defines a critical region of high uncertainty and reassigns labels for customers that have predicted scores within this region, such that members of the unprivileged group receive a positive label ($y = 1$) and vice versa. Formally, the critical region is defined as:

$$\max\left[\mathbf{P}(\hat{y} = 1|X),\ 1 - \mathbf{P}(\hat{y} = 1|X)\right] \leq \theta\,, \tag{11}$$

where $0.5 < \theta < 1$. Given a set of predicted scores and the true outcomes, a suitable value of θ and the number of posterior reclassifications can be tuned to optimize a fairness criterion (e.g., independence) based on the allowed fairness bound $\sigma = [\sigma_l, \sigma_u]$ for the corresponding constraint.

Equalized odds processor uses a different logic to post-process classifier predictions. It finds a cutoff value τ that optimizes the predictive performance while satisfying the separation criterion, i.e., ensuring the same false negative and false positive rate per group (Hardt et al., 2016).

Consider the receiver operating characteristic (ROC) curves that depict the trade-off between true and false positive rates for two sensitive groups. In an unfair scenario, the group-wise ROC curves have different slopes, which implies that not all trade-offs are achievable in each group. In a setting where a classifier optimizes accuracy, the optimal cutoff that satisfies sufficiency lies at the intersection of group-wise ROC curves. When optimizing for profit, the missclassification costs are not the same for both error rates. Thus, the optimal cutoff could lie somewhere else. Given a loss function $L(\cdot)$, the cutoff value τ can be found by optimizing the

following objective:

$$\min \mathbf{P}\left(s(X)|x_a = a, y = 0) \leq \tau\right) \cdot L(\hat{y} = 1, y = 0) + \left[1 - \mathbf{P}\left(s(X)|x_a = a, y = 1) > \tau\right)\right] \cdot L(\hat{y} = 0, y = 1) \quad (12)$$

Platt scaling is a post-processing method that stems from the notion of calibration (Platt, 1999). Calibration addresses the problem that some classification algorithms are not able to make a statement about the certainty of their prediction, i.e., the probability with which an instance belongs to a certain class. In credit scoring, the predicted score could be an indicator of default risk, but not the actual probability of default. A score $s(X)$ is calibrated if $\mathbf{P}\left(y = 1 \mid \ s(X) = \tau\right) = \tau$.

When extending the calibration condition to the group level, it becomes apparent that it implements the sufficiency criterion (see Barocas et al. (2019) for proof):

$$\mathbf{P}\left[y = 1 \mid s(X) = \tau, \ x_a = 1\right] = \ \mathbf{P}\left[y = 1 \mid s(X) = \tau, \ x_a = 0\right] = \tau \quad (13)$$

To achieve calibration per group, Platt scaling is applied separately to each sensitive group. The method uses the output of a possibly uncalibrated score $s(X)$ as input for logistic regression fitted against the target variable y. Based on the loss function of the logistic regression, the result is a new calibrated score that represents the probability that an instance belongs to the positive class. Formally, Platt scaling minimizes the log-loss $-\mathbb{E}[ylog\ (\sigma) + (1 - y)\log(1 - \sigma)]$ by finding the optimal parameters a and b of the sigmoid function $\sigma = \frac{1}{1+\exp(aS+b)}$.

5 Experimental Setup

5.1 Data

The empirical experiment is based on seven credit scoring data sets. Data sets *german* and *taiwan* stem from the UCI Machine Learning Repository[3]. *Pakdd*, *gmsc* and *homecredit* were provided by different companies for the data mining competitions on PAKDD[4] and Kaggle[5]. *Bene* and *uk* were collected from financial institutions in the Benelux and UK (Lessmann et al., 2015).

Each data set has a unique set of features describing a loan applicant and loan characteristics. The target variable y is a binary indicator of whether the applicant has repaid the loan ($y = 1$) or not ($y = 0$). Each data set also contains a sensitive demographic attribute x_a indicating the applicant's age group. The Equal Credit Opportunity Act prohibits that demographic characteristics such as applicants' age impact credit approval decisions. We distinguish two groups of applicants: $\{x_a = 1\}$ contains applications where the applicant's age is below ψ years, and $\{x_a = 0\}$ refers to the applications from customers older than ψ. We set $\psi = 25$, following the findings of Kamiran & Calders (2009) who used one of the consumer credit scoring data sets to discover that applicants from different age groups exhibit the greatest disparate impact (i.e., difference in $\mathbf{P}\left[y = 1 \mid x_a = a\right]$) at a threshold of 25 years. Table 3 summarizes the main characteristics of the data sets.

5.2 Experimental Setup

On each data set, we implement the eight fairness processors introduced in Section 4, following the model development pipeline depicted in Figure 1[6]. First, we partition the data into training (60%) and test (40%) sets. Next, we perform five-fold cross-validation on the training set. Each of the five combinations of training folds is used to train a scoring model and implement fairness processors. An unconstrained scoring model serves as a benchmark and represents the profit maximization scenario. We also consider in-processors in the form of the prejudice remover, adversarial debiasing and meta fair algorithm. Relying on an in-processor implies that the trained in-processor serves as a scorecard. This contrasts pre- and post-processors, in which the actual

[3]Source: `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)`, `https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients`

[4]Source: `https://www.kdnuggets.com/2010/03/f-pakdd-2010-data-mining-competition.html`

[5]Source: `https://kaggle.com/c/home-credit-default-risk`, `https://kaggle.com/c/givemesomecredit`

[6]The code reproducing the experiments is available at `https://box.hu-berlin.de/d/a90f6952a89c42138e6b/`

Table 3: Credit Scoring Data Sets

Data set	Sample size	No. features	Default rate	Sensitive group rate
german	1,000	61	.30	.19
bene	3,123	82	.33	.12
taiwan	23,531	76	.23	.14
uk	30,000	51	.04	.20
pakdd	50,000	185	.26	.11
gmsc	150,000	68	.07	.02
homecredit	307,511	92	.08	.04

scorecard is still based on a conventional ML algorithm. With respect to pre-processors, we use reweighing and disparate impact remover to transform training data before training a scoring model. Considering the post-processing methods, we apply reject option classification, equalized odds processor and Platt scaling to predictions of an unconstrained scorecard for the validation folds to learn suitable post-processing. Finally, each algorithm produces predictions for the applications in the test set.

Fairness pre- and post-processors, as well as an unconstrained scorecard, use four base classifiers: logistic regression, random forest, extreme gradient boosting and artificial neural network. This allows us to check the robustness of processors across different base models. Meta-parameters of the base classifiers are tuned in a nested four-fold cross-validation on the training data. The meta-parameters of fairness processors are also tuned using grid search. The details on the meta-parameter values and the tuning procedure are provided in A.

Fairness processors and benchmarks are evaluated on the test set using multiple performance metrics. First, we measure the profitability of a scorecard using the Expected Maximum Profit (EMP) criterion (Verbraken et al., 2014). The EMP can be interpreted as the incremental profit from using a scorecard compared to a base scenario in which all credit applications are approved without screening. Table 4 provides a confusion matrix of a scoring model, where π_i are prior probabilities of good and bad risks, and $F_i(\tau)$ are predicted cumulative density functions of the scores of class i given a cutoff value τ. If an applicant is predicted to be a good risk, no additional costs or benefits are observed. In contrast, if an applicant is predicted to default, the company faces cost C in case of an incorrect prediction and gets benefit B from an accurate prediction.

The parameter B reflects the benefit of correctly identifying a bad risk. By not providing a loan to a defaulter, the company avoids losses; specifically, the expected loss in case of default:

$$B = \frac{\text{LGD} \cdot \text{EAD}}{A} \,, \tag{14}$$

where LGD refers to the loss given default, EAD is the exposure at default, and A is the principal of the loan. In practice, B can vary between 0 and 1 and several distributions may arise (Somers & Whittaker, 2007). We follow Verbraken et al. (2014) and consider B as a random variable with the following probability distribution:

- $B = 0$ with probability p_0 (a customer repays the entire loan);
- $B = 1$ with probability p_1 (a customer defaults on the entire loan);
- B follows a uniform distribution in (0, 1) with $F(B) = 1 - p_0 - p_1$.

The parameter C reflects the cost associated with misclassifying good risks. By rejecting a good customer, the company faces an opportunity cost equivalent to the return on investment ROI that could have been made when approving the loan:

$$C = \text{ROI} = \frac{I}{A} \,, \tag{15}$$

where I is the total interest payments. Given these parameters, we compute EMP as:

$$\text{EMP} = \int_0^1 \Big[B \cdot \pi_0 F_0(\tau) - C \cdot \pi_1 F_1(\tau) \Big] f(B) d(B) \tag{16}$$

Table 4: Confusion Matrix with Costs

Actual label	Predicted label: Bad risk	Predicted label: Good risk
Bad risk	$\pi_0 F_0(\tau)$ benefit: B	$\pi_0(1 - F_0(\tau))$ cost: 0
Good risk	$\pi_1 F_1(\tau)$ cost: C	$\pi_1(1 - F_1(\tau))$ cost: 0

This paper follows the empirical findings of Verbraken et al. (2014) and assumes the point masses $p_0 = 0.55$ for no loss and $p_1 = 0.1$ for full loss to compute B. With respect to the parameter C, we also follow Verbraken et al. (2014) in assuming a constant return on investment of 0.2664.

Apart from estimating the profitability of each fairness processor using the EMP, we also compute the area under the ROC curve (AUC), which is a widely used indicator of the discriminatory ability of a scoring model. In addition, we evaluate fairness by measuring independence, separation and sufficiency. We aggregate the performance of pre- and post-processors over seven credit scoring data sets, five training fold combinations and four base classifiers, obtaining 140 performance estimates per processor. Since in-processors do not require a base classifier, their performance is aggregated over 35 values obtained from seven data sets and five training fold combinations.

6 Empirical Results

This section presents the empirical results. We first examine the correlation between profit and fairness criteria. Next, we compare the performance of different fairness processors. Last, we analyze the profit-fairness trade-off by examining Pareto frontiers with non-dominated solutions.

6.1 Correlation Analysis

Table 5 depicts the mean Spearman correlation between the evaluation metrics. The correlation coefficients are computed on the performance estimates obtained from different variants of fairness processors and averaged over the seven credit scoring data sets. The results suggest that the AUC and the EMP often produce similar model rankings (correlation is 0.81). Still, there is some disagreement between the two measures, which indicates that optimizing the EMP is important to identify potentially more profitable scorecards. Therefore, we emphasize the EMP in the following.

Comparing the EMP with the fairness criteria, we observe a moderate negative correlation between independence, separation and profitability[7]. As expected, integrating fairness constraints to reduce discrimination prevents a credit scorecard from taking full advantage of the available information, which decreases the EMP. At the same time, a moderate positive correlation between sufficiency and the EMP suggests that optimizing profitability without implementing additional fairness constraints would also improve sufficiency. This result confirms the observation that most ML algorithms are designed to automatically achieve sufficiency and implies that directly optimizing sufficiency with a fairness processor is not essential.

A different conclusion emerges from examining the agreement of the other two fairness criteria. As indicated by Table 5, independence and separation have a strong positive correlation of 0.95. Optimizing either of these two criteria will, therefore, favor models that fulfill both independence and separation. In other words, reducing the mutual information between a sensitive attribute and model predictions also helps to align the parity of error rates across the sensitive groups. This is an interesting finding, given that the former constraint targeted by independence is stricter compared to the one targeted by separation. For a risk analyst, the observed result implies that it is ample to rely on a single fairness criterion. Since separation has a better ability to capture the

[7]Higher values of the AUC and the EMP indicate better performance, whereas lower values of independence, separation and sufficiency indicate higher fairness. Therefore, we invert correlation signs between the two former performance metrics and the three fairness criteria to facilitate the consistent interpretation of the results.

Table 5: Rank Correlation between Evaluation Metrics

Metric	AUC	EMP	IND	SP	SF
AUC	1				
EMP	0.8120	1			
IND	−0.3624	−0.4013	1		
SP	−0.2527	−0.3077	0.9554	1	
SF	0.4017	0.3000	−0.1550	−0.0721	1

Abbreviations: AUC = area under the ROC curve, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency.

cost asymmetry (see Section 3 for details), we conclude that optimizing and measuring the separation criterion is the most suitable way to integrate and evaluate the fairness of a credit scoring model.

6.2 Benchmarking Fairness Processors

Table 6 provides the mean ranks of fairness processors across the seven credit scoring data sets. A lower rank indicates a better performance of a processor in terms of a particular evaluation measure. Individual results for each of the data sets are provided in B.

According to Table 6, an unconstrained scoring model achieves the best performance in terms of the AUC and the EMP. Using any of the processors to integrate fairness comes at the cost of worse performance and profitability. At the same time, the unconstrained profit maximization leads to high discrimination: six out of eight fairness processors achieve better separation, five reach better independence, whereas only two processors have better sufficiency.

The reject option classification post-processor demonstrates the best fairness in terms of independence and separation. This is achieved by sacrificing a large share of profit: the mean EMP rank increases by 4.12 compared to the unconstrained model. The second-best profitability is obtained by another post-processor, Platt scaling, which has a marginally higher EMP rank than the unconstrained model. At the same time, as a sufficiency-oriented method, Platt scaling fails at improving independence and separation. These results emphasize the importance of finding a balance between profit and fairness.

Comparing processors within the implementation methods, we can identify promising fairness integration techniques. Considering post-processors, the equalized odds processor is outperformed by reject option classification in all evaluation measures. Platt scaling achieves a better EMP and sufficiency, but this comes at a high cost of the worst independence and separation across the benchmarks. Concerning pre-processors, reweighing achieves the best fairness but substantially decreases the profitability; the disparate impact remover retains a higher profit but does not improve independence and separation compared to the unconstrained scorecard.

With respect to in-processors, the meta fair algorithm is outperformed by the other techniques in all metrics except for sufficiency. The other two in-processors demonstrate a good balance between profitability and fairness. Comparing the prejudice remover and adversarial debiasing, we can conclude that the former can retain a higher EMP but improves fairness to a smaller magnitude compared to the latter algorithm. It is important to note that in-processors allow more flexibility in prioritizing fairness or profit through their meta-parameters. For instance, adversarial debiasing allows tuning the importance of fairness by changing the weight of the corresponding regularization penalty, which allows decision-makers to gear these processors toward a specific setting. This contrasts with some processors from the other methods, such as reweighing or Platt scaling, which do not have meta-parameters to control the fairness-profitability trade-off.

Overall, results suggest that the choice of a suitable processor depends on the relative importance of the conflicting objectives to a decision-maker. While some processors such as the meta-fair algorithm and the equalized odds processor are dominated by their counterparts, other techniques arrive at different solutions in the space between sacrificing profit and reducing discrimination. The best fairness is achieved with reject option classification, whereas in-processors demonstrate a higher potential for finding a balance between profit and fairness.

Table 6: Mean Ranks of Fairness Processors

Method	Fairness processor	AUC	EMP	IND	SP	SF
PRE	Reweighing	6.19 (.15)	5.49 (.32)	2.58 (.14)	2.69 (.17)	**2.77** (.26)
	Disparate impact remover	3.08 (.13)	3.48 (.15)	6.26 (.16)	6.01 (.17)	3.55 (.25)
IN	Prejudice remover	3.61 (.16)	3.59 (.17)	5.58 (.15)	5.25 (.15)	5.37 (.26)
	Adversarial debiasing	3.93 (.19)	4.81 (.16)	3.98 (.20)	3.83 (.20)	5.40 (.20)
	Meta fair algorithm	5.15 (.21)	4.94 (.19)	5.96 (.14)	5.96 (.18)	4.09 (.27)
POST	Reject option classification	7.72 (.06)	7.52 (.07)	**2.40** (.14)	**2.26** (.14)	7.07 (.16)
	Equalized odds processor	8.72 (.05)	8.31 (.07)	3.81 (.24)	4.79 (.24)	8.43 (.11)
	Platt scaling	3.89 (.15)	3.46 (.14)	8.16 (.17)	8.12 (.17)	4.39 (.33)
Unconstrained profit maximization		**2.71** (.14)	**3.40** (.15)	5.94 (.13)	5.97 (.13)	3.94 (.20)

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Ranks are averaged over seven data sets × five folds × four base models. Standard errors in parenthesis.

6.3 The Cost of Fairness

Previous results indicate that it is possible to improve fairness by decreasing the EMP. However, aggregated rank-based results do not facilitate a direct estimation of the cost of fairness. Recall that the EMP measures the incremental profit compared to a base scenario in which loan applications are accepted without screening. This typically leads to a small magnitude of EMP differences across different classifiers (Kozodoi et al., 2019b) and complicates the interpretation of the metric values. To enable a more direct interpretation, we quantify the cost of fairness by measuring profit per EUR issued by a financial institution. We follow the EMP calculation procedure and use the same assumptions on the parameters B and C, but normalize the costs such that the base scenario represents rejecting all applications. Table 7 depicts the corresponding cost matrix.

Figure 2 visualizes the trade-off between profit and fairness for all seven data sets using the concept of Pareto frontiers. The points on the frontiers refer to the test set performance of fairness processors trained with different base classifiers and on different combinations of the training folds. The frontiers only contain the non-dominated solutions, i.e., the points where it is impossible to improve on one objective (i.e., profit) without harming the other objective (i.e., fairness). Based on the previous results, we use separation to measure fairness. Frontiers depicting the trade-off between the other fairness criteria and the EMP are available in B.

Figure 2 reveals that discrimination can be substantially reduced at a relatively low cost. Recall that separation indicates the difference between the false positive and false negative rates across the sensitive groups. According to Figure 2, reducing the difference in error rates below 0.2 is possible while sacrificing less than €0.01 profit per EUR issued. Across the data sets, this translates to a mean profit reduction of 5.19% compared to the most profitable scorecard with stronger discrimination. At the same time, completely eliminating unfairness is more costly: achieving separation of 0 is only possible when sacrificing more than 29% of the profit. However, since perfect fairness is not required by regulation, we conclude that a financial institution can reduce discrimination to a reasonable extent while maintaining a relatively high profit margin.

Table 7: Cost Matrix for Profit Computation

	Predicted label	
Actual label	Bad risk	Good risk
Bad risk	0	$-B = -(\text{LGD} \cdot \text{EAD})/A$
Good risk	$-C = -I/A$	$C = I/A$

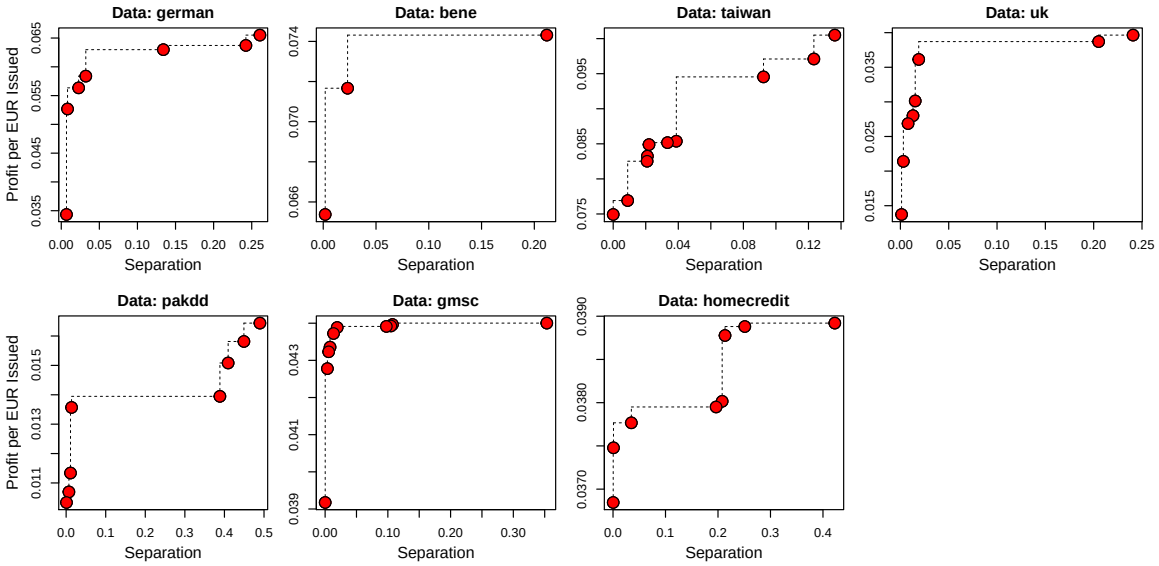

Figure 2: Profit-Fairness Trade-Off: Frontiers with Non-Dominated Solutions

7 Conclusion

The paper sets out to consolidate recent advancements in fair ML from a credit scoring perspective. We provide a comprehensive catalog of the fairness interventions suggested in previous studies and discuss the adequacy of established fairness criteria for credit scoring. We also conduct an empirical experiment in which we compare the performance of relevant fairness processors on real-world credit scoring data sets and examine the trade-off between profit and fairness.

The conceptual comparison of different fairness criteria reveals separation to be the most appropriate metric for credit scoring. Separation acknowledges the imbalanced misclassification costs, which are instrumental to the lending business. The presented catalog of fairness processors offers practitioners a starting point for deciding which processors to consider for a given problem setting. The catalog also indicates that most processors have been evaluated based on their accuracy and that some relevant credit scoring scenarios are not well covered by the available processors. For example, in a setting with multiple sensitive attributes (e.g., race and religion), only two processors, adversarial debiasing and reject option classification, facilitate optimizing the separation criterion.

The empirical study benchmarks fairness processors in a profit-oriented credit scoring setup. Several implications emerge from the results. First, examining the agreement between the fairness criteria under study reveals that separation and independence are strongly correlated. While other empirical studies support this finding (Friedler et al., 2019), it contradicts the intuition from theoretical considerations that fairness criteria are mutually exclusive (Mitchell et al., 2018). We also find that sufficiency has a property to be achievable by any well-trained classifier that can predict the sensitive attribute from the other features (Barocas et al., 2019). This calls into question the overall suitability of sufficiency for credit scoring and further emphasizes separation as a proper criterion for measuring the fairness of credit scorecards.

Second, we find that the choice of an appropriate fairness processor depends on the implementation feasibility and preferences of a decision-maker regarding the conflicting objectives of profit and fairness. Post-processing methods such as reject option classification are the easiest to implement in production but improve fairness at a high monetary cost. In-processors such as adversarial debiasing perform best in finding the profit-fairness trade-off and offer the most flexibility in calibrating the importance of the conflicting objectives. However, using in-processors requires replacing a deployed scoring model with a new algorithm, which might require regulatory approval and is, more generally, associated with considerable efforts.

Third, while achieving perfect fairness is costly, we find that reducing discrimination to a reasonable extent is possible while maintaining a relatively high profit. These results support the current anti-discrimination regulation that allows unfairness to exist up to a certain limited extent. The analysis of fairness processors from the perspective of the Pareto frontiers offers decision-makers a tool to analyze the profit-fairness trade-off specific to their context and identify techniques that reduce discrimination to a required level at the smallest monetary cost.

Our study may also have implications for customer scoring models beyond the credit industry. Fairness concerns arise from the increasing use of ML to automate decisions in many domains such as hiring (Raghavan et al., 2020), college admission (Mitchell et al., 2018) or criminal risk assessment (Berk et al., 2018). The catalog of fairness processors and the results of their empirical analysis can aid these domains in identifying suitable techniques for integrating fairness in the decision support systems. Future work on fair ML may also draw value from the empirical comparison in that it highlights effective approaches that set a benchmark for new fairness processors.

References

Banasik, J., Crook, J., & Thomas, L. (2003). Sample selection bias in credit scoring models. *Journal of the Operational Research Society*, *54*, 822–832.

Barocas, S., Hardt, M., & Narayanan, A. (2019). *Fairness and Machine Learning*. fairmlbook.org.

Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, *104*, 671–732.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., & Roth, A. (2018). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*.

Calders, T., Kamiran, F., & Pechenizkiy, M. (2009). Building classifiers with independency constraints. In *IEEE International Conference on Data Mining Workshops* (pp. 13–18).

Calders, T., & Verwer, S. (2010). Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, *21*, 277–292.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., & Varshney, K. R. (2017). Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems* (pp. 3992–4001).

Celis, L. E., Huang, L., Keswani, V., & Vishnoi, N. K. (2019). Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Conference on Fairness, Accountability, and Transparency* (pp. 319–328).

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, *5*, 153–163.

Cleary, T. A. (1968). Test bias: Prediction of grades of negro and white students in integrated colleges. *Journal of Educational Measurement*, *5*, 115–124.

Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., & Huq, A. (2017). Algorithmic decision making and the cost of fairness. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 797–806).

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, *183*, 1447–1465.

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

Darlington, R. B. (1971). Another look at "cultural fairness". *Journal of Educational Measurement*, *8*, 71–82.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness through awareness. In *Innovations in Theoretical Computer Science Conference* (pp. 214–226).

Equal Credit Opportunity Act (1974). Art. 9 & 15 U.S. code §1691. URL: `https://www.law.cornell.edu/uscode/text/15/1691c`. Accessed 1 December 2020.

European Commission (2017). Guidelines on data protection officers. URL: `http://ec.europa.eu/newsroom/document.cfm?doc_id=44100`. Accessed 1 December 2020.

Executive Office of the President (2018). Big data: A report on algorithmic systems, opportunity,and civil rights. URL: https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/2016_0504_data_discrimination.pdf. Accessed 1 December 2020.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., & Venkatasubramanian, S. (2015). Certifying and removing disparate impact. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 259–268).

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. In *Conference on Fairness, Accountability, and Transparency* (pp. 329–338).

Fuster, A., Goldsmith-Pinkham, P., Ramadorai, T., & Walther, A. (2017). *Predictably unequal? The effects of machine learning on credit markets*. Technical Report, National Bureau of Economic Research.

Gajane, P., & Pechenizkiy, M. (2017). On formalizing fairness in prediction with machine learning. *arXiv preprint arXiv:1710.03184*.

Goh, G., Cotter, A., Gupta, M., & Friedlander, M. P. (2016). Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems* (pp. 2415–2423).

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

Johndrow, J. E., Lum, K. et al. (2019). An algorithm for removing sensitive information: application to race-independent recidivism prediction. *The Annals of Applied Statistics*, *13*, 189–220.

Kamiran, F., & Calders, T. (2009). Classifying without discriminating. In *International Conference on Computer, Control and Communication* (pp. 1–6).

Kamiran, F., Karim, A., & Zhang, X. (2012). Decision theory for discrimination-aware classification. In *International Conference on Data Mining* (pp. 924–929).

Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 35–50).

Kleinberg, J., Mullainathan, S., & Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.

Somers, M., & Whittaker, J. (2007). Quantile regression for modelling distributions of profit and loss. *European Journal of Operational Research*, *183*, 1477–1487.

Kozodoi, N., Katsas, P., Lessmann, S., Moreira-Matias, L., & Papakonstantinou, K. (2019). Shallow self-learning for reject inference in credit scoring. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 516–532).

Kozodoi, N., Lessmann, S., Papakonstantinou, K., Gatsoulis, Y., & Baesens, B. (2019b). A multi-objective approach for profit-driven feature selection in credit scoring. *Decision Support Systems*, *120*, 106–117.

Kusner, M. J., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. In *Advances in Neural Information Processing Systems* (pp. 4066–4076).

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*, 124–136.

Liu, L. T., Dean, S., Rolf, E., Simchowitz, M., & Hardt, M. (2018). Delayed impact of fair machine learning. In *International Conference on Machine Learning* (pp. 3150–3158).

Louizos, C., Swersky, K., Li, Y., Welling, M., & Zemel, R. (2016). The variational fair autoencoder. In *International Conference on Learning Representations*.

Luong, B. T., Ruggieri, S., & Turini, F. (2011). K-NN as an implementation of situation testing for discrimination discovery and prevention. In *ACM SIGKDD International Conference on Knowledge discovery and Data Mining* (pp. 502–510).

Mitchell, S., Potash, E., Barocas, S., D'Amour, A., & Lum, K. (2018). Prediction-based decisions and fairness: A catalogue of choices, assumptions, and definitions. *arXiv preprint arXiv:1811.07867*.

Narayanan, A. (2018). Translation tutorial: 21 fairness definitions and their politics. In *Conference on Fairness, Accountability, and Transparency*.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *International Conference on Machine Learning* (pp. 625–632).

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, *10*, 61–74.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. Q. (2017). On fairness and calibration. In *Advances in Neural Information Processing Systems* (pp. 5680–5689).

Raghavan, M., Barocas, S., Kleinberg, J., & Levy, K. (2020). Mitigating bias in algorithmic hiring: Evaluating claims and practices. In *Conference on Fairness, Accountability, and Transparency* (pp. 469–481).

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*, 505–513.

Woodworth, B., Gunasekar, S., Ohannessian, M. I., & Srebro, N. (2017). Learning non-discriminatory predictors. In *Conference on Learning Theory* (pp. 1920–1953).

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning* (pp. 609–616).

Zafar, M. B., Valera, I., Gomez Rodriguez, M., & Gummadi, K. P. (2017a). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web* (pp. 1171–1180).

Zafar, M. B., Valera, I., Gomez Rodriguez, M., & Gummadi, K. P. (2017b). Fairness constraints: Mechanisms for fair classification. In *International Conference on Artificial Intelligence and Statistics* (pp. 962–970).

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning fair representations. In *International Conference on Machine Learning* (pp. 325–333).

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *AAAI/ACM Conference on AI, Ethics, and Society* (pp. 335–340).

A Meta-Parameters of Base Models and Fairness Processors

This appendix provides meta-parameter values of the base classifiers and the fairness processors used in the empirical experiment. Table 9 depicts the candidate values of the meta-parameters of the four base classifiers used as a scoring model by fairness pre- and post-processors as well as by the unconstrained profit maximization benchmark. The meta-parameter values are optimized with grid search using the EMP as an objective. The meta-parameter tuning is performed separately on each combination of the training folds using a nested four-fold cross-validation.

Table 8 provides candidate values of the meta-parameters of fairness processors that are tuned within the higher-level cross-validation framework. We measure the EMP of fairness processors on each validation fold to select the appropriate meta-parameter values. The notation for processor meta-parameters and their explanation is available in Section 4.

Table 8: Meta-Parameters of Fairness Processors

Method	Fairness processor	Meta-parameter	Candidate values
Pre-processing	Reweighting	–	–
	Disparate impact remover	Repair level λ	$0.5, 0.6, 0.7, 0.8, 0.9, 1$
In-processing	Prejudice remover	Fairness penalty η	$1, 5, 15, 30, 50, 70, 100, 150$
	Meta fair algorithm	Fairness penalty τ	$0.05, 0.10, 0.15, 0.20, 0.25, 0.30$
	Adversarial debiasing	Adversarial loss weight α Number of epochs Batch size	$0.1, 0.01, 0.001$ 50 128
Post-processing	Reject option classification	Fairness metric bound σ Number of thresholds Number of ROC margins	$[-0.1, 0.1]$, $[-0.2, 0.2]$, $[-0.3, 0.3]$ 100 50
	Equalized odds processor	–	–
	Platt scaling	–	–

Table 9: Meta-Parameters of Base Classifiers

Base classifier	Meta-parameter	Candidate values
Logistic regression	–	–
Random forest	Number of trees Number of sampled features	500 5, 10, 15
Gradient boosting	Number of trees Maximum tree depth Learning rate Ratio of sampled features Ratio of sampled cases Minimum child weight	100, 500, 1000 5, 10 0.1 0.5, 1 0.5, 1 0.5, 1, 3
Neural network	Size Decay Maximum umber of iterations	5, 10, 15 0.1, 0.5, 1, 1.5, 2 1000

Table 11: Performance of Fairness Processors: Bene

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.74689	.61084	.15282	.09344	.07767	.04867
	Disparate impact remover	.78747	.61376	.15478	.36216	.28325	.06935
IN	Prejudice remover	.79517	.61936	.15578	.31412	.22841	.16150
	Adversarial debiasing	.78133	.61184	.15454	.32500	.24499	.13925
	Meta fair algorithm	.78753	.61428	.15484	.32267	.24465	.09799
POST	Reject option classification	.70821	.60366	.15334	.07257	.07113	.23395
	Equalized odds processor	.66765	.60389	.15337	.03963	.08436	.24853
	Platt scaling	.78801	.61582	.15508	.44695	.36842	.06806
Unconstrained profit maximization		.78960	.61517	.15495	.35404	.27431	.08246

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

B Extended Empirical Results

This appendix provides additional results of the experiment presented in Section 6. Tables 10 – 16 compare performance of fairness processors on each of the seven credit scoring data sets. Performance of pre- and post-processors is averaged over 25 values from five cross-validation folds × five base classifiers; performance of in-processors is aggregated over five training fold combinations.

Figures 4 – 9 depict the Pareto frontiers that visualize the trade-off between the EMP and the three fairness criteria: independence, separation and sufficiency for each of the seven data sets. The points on the Pareto frontiers refer to the performance of fairness processors based on different base classifiers and trained on different combinations of training folds. The performance is evaluated on the test sets. The figures only depict the set of non-dominated solutions, i.e., points where it is impossible to improve one of the objectives (i.e., one of the fairness criteria) without decreasing the other objective (i.e., the EMP).

Table 10: Performance of Fairness Processors: German

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.76042	.61128	.16589	.22039	.17517	.15625
	Disparate impact remover	.81210	.61722	.16884	.29888	.19186	.12492
IN	Prejudice remover	.79333	.61121	.16741	.32004	.20906	.16549
	Adversarial debiasing	.79648	.61027	.16713	.25285	.18977	.17048
	Meta fair algorithm	.80743	.61577	.16843	.22622	.11165	.15552
POST	Reject option classification	.71241	.59845	.16496	.11047	.08809	.21209
	Equalized odds processor	.69989	.59650	.16470	.08358	.14754	.25140
	Platt scaling	.80120	.61385	.16851	.41951	.33694	.15316
Unconstrained profit maximization		.81235	.61431	.16890	.30777	.19788	.14454

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

Table 12: Performance of Fairness Processors: Taiwan

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.75658	.59426	.18877	.05136	.04672	.01067
	Disparate impact remover	.79537	.59987	.18850	.21667	.17429	.01201
IN	Prejudice remover	.79092	.59926	.18833	.10230	.07952	.02106
	Adversarial debiasing	.80001	.59639	.18805	.03189	.03325	.01902
	Meta fair algorithm	.79575	.60114	.18846	.15744	.12517	.00211
POST	Reject option classification	.70107	.58715	.18682	.10027	.07966	.04467
	Equalized odds processor	.63174	.57992	.18586	.19441	.18695	.04072
	Platt scaling	.74929	.59566	.18794	.37045	.31227	.01947
Unconstrained profit maximization		.74886	.59549	.18795	.15845	.12483	.03850

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

Table 13: Performance of Fairness Processors: UK

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.67858	.55439	.25772	.08071	.03960	.00557
	Disparate impact remover	.71737	.55432	.25810	.29260	.20506	.01134
IN	Prejudice remover	.70867	.55433	.25811	.30332	.24326	.01908
	Adversarial debiasing	.70922	.55431	.25810	.26137	.16218	.01807
	Meta fair algorithm	.55836	.55416	.25808	.33894	.41284	.00213
POST	Reject option classification	.65232	.55423	.25809	.06214	.02221	.01618
	Equalized odds processor	.62059	.55420	.25809	.17830	.20424	.02248
	Platt scaling	.69862	.55430	.25810	.68387	.53291	.01997
Unconstrained profit maximization		.71294	.55431	.25810	.31113	.21414	.01410

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

Table 14: Performance of Fairness Processors: PAKDD

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.57456	.58349	.17341	.05304	.04347	.01710
	Disparate impact remover	.60305	.58378	.17388	.45458	.42803	.06931
IN	Prejudice remover	.60067	.58368	.17387	.36907	.35012	.05257
	Adversarial debiasing	.58680	.58345	.17384	.29920	.27825	.08155
	Meta fair algorithm	.59794	.58364	.17386	.33930	.31341	.09164
POST	Reject option classification	.56500	.58313	.17379	.07515	.05698	.10034
	Equalized odds processor	.56097	.58307	.17378	.02446	.04124	.10451
	Platt scaling	.59930	.58374	.17388	.62671	.59783	.09824
Unconstrained profit maximization		.59808	.58372	.17387	.42151	.39698	.07050

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

Table 15: Performance of Fairness Processors: GMSC

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.84250	.55892	.24914	.05946	.04372	.00545
	Disparate impact remover	.85353	.55933	.24879	.19355	.10775	.01515
IN	Prejudice remover	.85530	.55932	.24879	.24542	.14445	.00854
	Adversarial debiasing	.85879	.55941	.24880	.11259	.05082	.01535
	Meta fair algorithm	.82614	.55930	.24879	.43184	.27663	.01824
POST	Reject option classification	.77619	.55758	.24855	.05903	.05247	.01868
	Equalized odds processor	.59027	.55680	.24845	.38439	.28117	.02399
	Platt scaling	.85315	.55937	.24879	.56911	.35642	.00000
Unconstrained profit maximization		.85449	.55937	.24879	.24606	.14294	.01206

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

Table 16: Performance of Fairness Processors: Homecredit

Method	Fairness processor	AUC	AR	EMP	IND	SP	SF
PRE	Reweighing	.72792	.55895	.24332	.12371	.09468	.00564
	Disparate impact remover	.73721	.55893	.24338	.24096	.16781	.02038
IN	Prejudice remover	.73873	.55895	.24338	.20722	.13562	.02376
	Adversarial debiasing	.73789	.55891	.24338	.31695	.24637	.01687
	Meta fair algorithm	.73515	.55884	.24337	.34821	.26609	.00584
POST	Reject option classification	.67847	.55851	.24332	.05060	.02065	.02896
	Equalized odds processor	.61899	.55830	.24329	.24806	.24824	.04265
	Platt scaling	.74060	.55899	.24339	.48844	.36430	.01401
Unconstrained profit maximization		.74105	.55899	.24339	.33044	.24348	.01304

Abbreviations: PRE = pre-processor, IN = in-processor, POST = post-processor; AUC = area under the ROC curve, AR = acceptance rate, EMP = expected maximum profit, IND = independence, SP = separation, SF = sufficiency. Performance is averaged over five cross-validation folds × four base models.

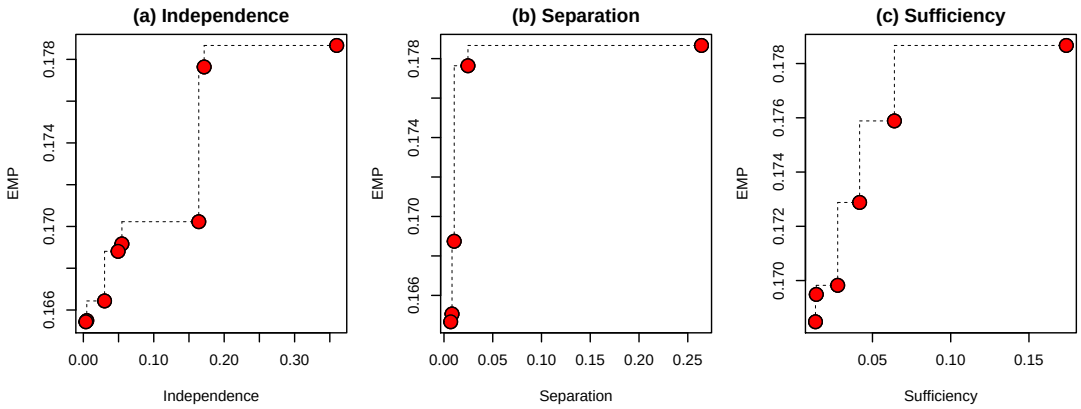

Figure 3: Frontier with Non-Dominated Solutions: German

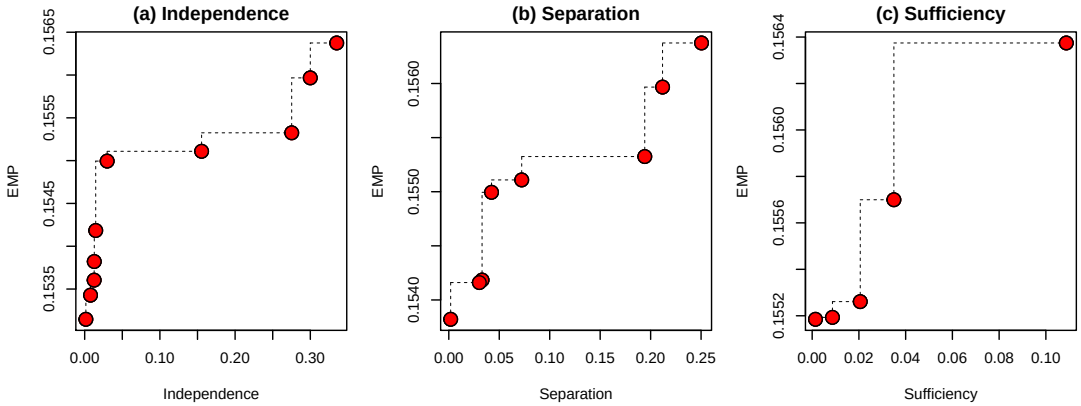

Figure 4: Frontier with Non-Dominated Solutions: Bene

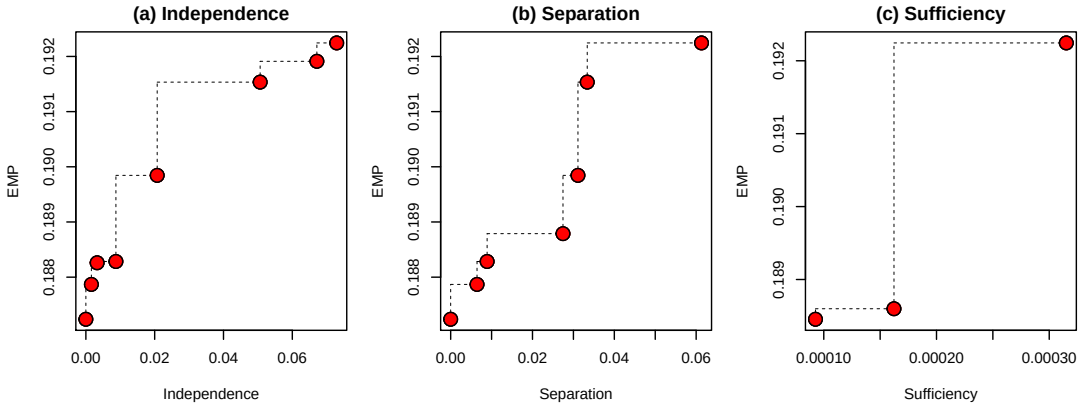

Figure 5: Frontier with Non-Dominated Solutions: Taiwan

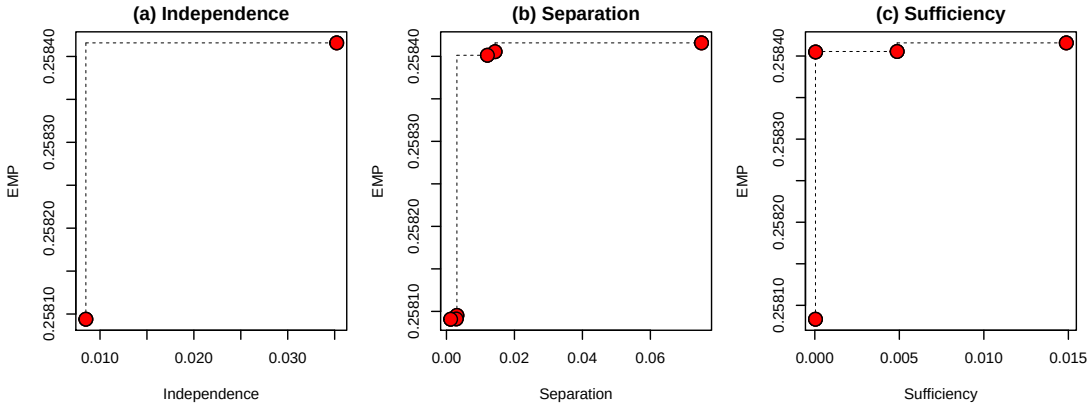

Figure 6: Frontier with Non-Dominated Solutions: UK

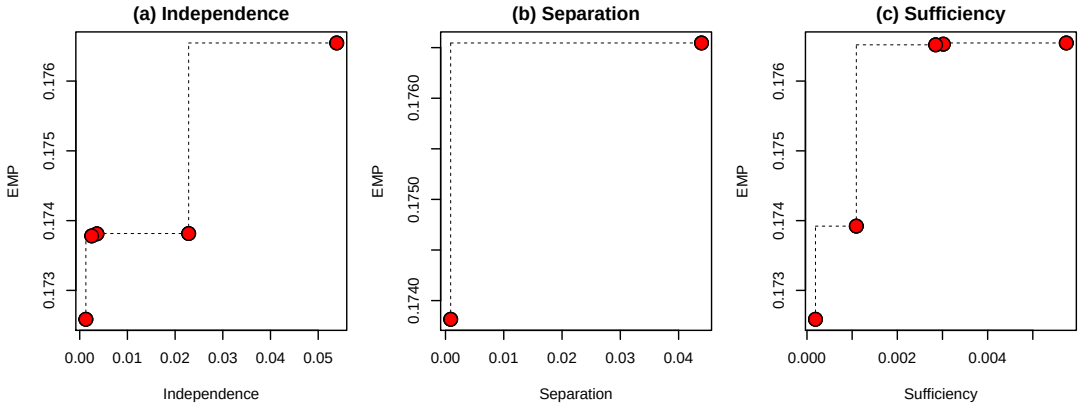

Figure 7: Frontier with Non-Dominated Solutions: PAKDD

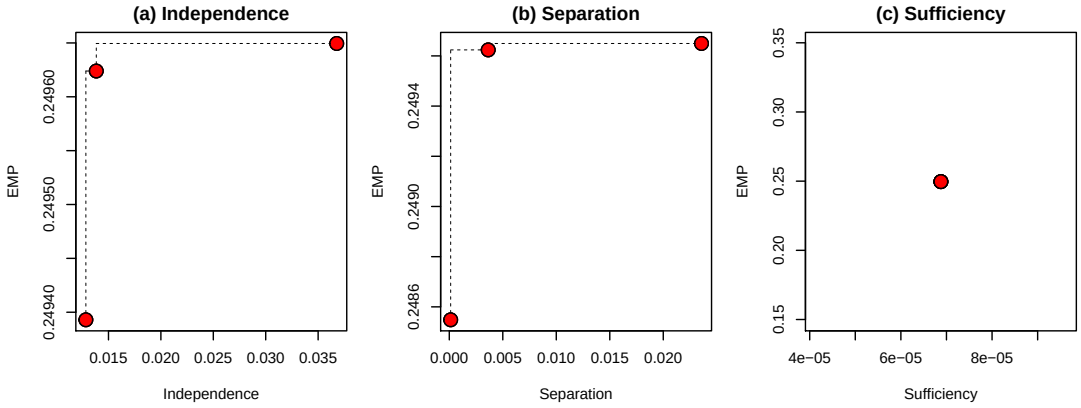

Figure 8: Frontier with Non-Dominated Solutions: GMSC

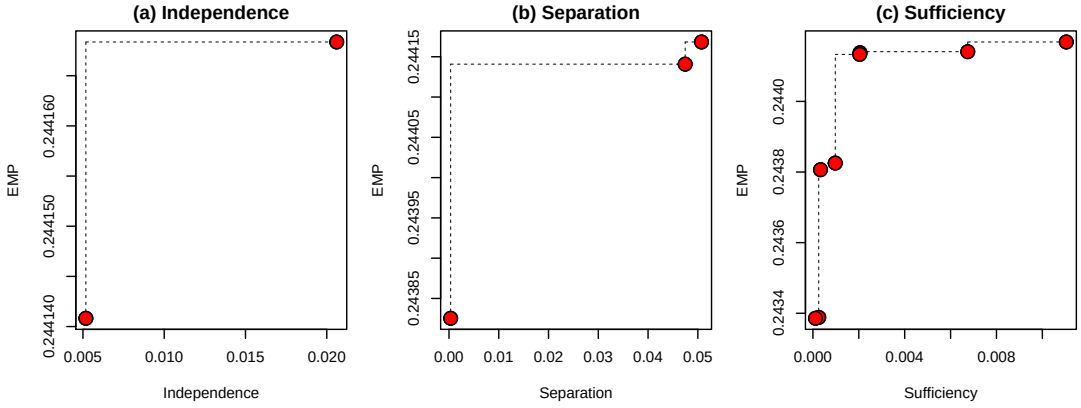

Figure 9: Frontier with Non-Dominated Solutions: Homecredit