Master Thesis

for the acquisition of the academic degree

Master of Science (*M.Sc.*)

in the subject of Statistics

Ludwig-Maximilians-University Munich

Faculty of Mathematics, Informatics and Statistics

Department: Statistics

Multicalibration in Survival Analysis: Black-Box Post-Processing for Fairness

submitted by

Carolin Becker

Supervisors: Prof. Dr. Bernd Bischl
Dr. Ludwig Bothmann
M. Sc. Florian Pfisterer

Date: 18.09.2021

Contents

1. Introduction

Motivation. The increasing usage of **automated high-stake decision-making** by states, companies, and individuals has a considerable impact on an individual's life: for example, on job applications (Schumann et al., 2020), justice (Angwin et al., 2016), credit scoring (Khandani et al., 2010), and healthcare (Grote & Berens, 2020). Nevertheless, along with many positive effects like efficiency, automated decisions can unintentionally influence the real world. By, e.g., replicating stereotypes in selected data sets or ignoring minorities in the predictions, decisions can be contradictory to an individual's or societal interests.

The research field of **algorithmic fairness** deals with the mitigation of unfair automated decision-making. Individuals or societies define "unfairness" depending on the decision-making context and their ethical values: should models treat individuals or groups equally? Should models have the same predictive performance for everyone? As a result of divergent answers to these exemplary questions, there are many definitions of fairness.

In the current research, fairness is primarily measured for groups with one binary sensitive attribute, e.g., different genders (male vs. not male) or races (dark-skinned vs. light-skinned persons). However, group fairness does not imply so-called **subgroups** fairness. Here, we compare overlapping groups with multiple features: even though an algorithm treats overall gender and ethnicity fairly, it might not mean that dark-skinned women are treated fairly (Buolamwini, 2018; Foulds et al., 2020).

One important fairness measure is **well-calibration** within the groups (Kleinberg et al., 2017). Well-calibration means that the predicted probability of a model for a subject numerically reflects the actual probability for all groups. In particular, calibration is beneficial when the result does not constitute an immediate decision but rather serves to inform decision-making during risk assessment (e.g., health risk assessment, awarding a loan; Noriega-Campero et al., 2019). In addition, well-calibration ensures that risk assessments for different (protected) populations are equivalent (Pleiss et al., 2017). For example, as small groups typically do not affect the average minimized loss over the entire training data, a trained model might be almost perfectly calibrated for the majority group but not calibrated with respect to minority groups (Chouldechova & Roth, 2020).

To address the bias in modeling with respect to subgroups, Hébert-Johnson et al. (2018) and Kim et al. (2019) presented the **multicalibration framework**. The framework post-processes an existing model by calibrating or debiasing the prediction for all subgroups. So far, this boosting algorithm has only been developed for binary classification. Survival models predict the probability that an event (e.g., death or illness) takes place at a specific time based on given features. Thus, survival models are often the basis of risk assessment (Angwin et al., 2016; Barda et al., 2021). Especially in healthcare, different risk evaluations for persons belonging to different subgroups might translate into the false allocation of medical resources (Ferryman & Pitcan, 2018). Therefore, it is crucial to have equal predictive power across all subgroups in survival analysis.

This thesis extends the multicalibration framework to a survival setting by adapting the optimized loss function.

Research Question. We answer the following research question (**RQ**):

Is a multicalibrated survival model fairer than the same survival model without post-processing?

The desired metric is the mean of the censored version of the Integrated Brier Score (IBS) with respect to the subgroups (Subgroup-IBS, S-IBS) $\ell^C_{SIBS} = \frac{1}{n_\mathcal{S}} \cdot \sum_{j \in \mathcal{C}} \ell^C_{IBS,j}$, where $\mathcal{C} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{n_\mathcal{S}}\}$ is the set of all evaluated subgroups and $n_\mathcal{S}$ is the number of subgroups which are evaluated. $\ell^C_{IBS,\mathcal{S}}$ is the IBS (9) evaluated on subgroup $\boldsymbol{x} \in \mathcal{S}$. Subquestions (SQ) in this context are:

- **SQ1:** *How does multicalibration affect the calibration overall?* The desired metric is the IBS with respect to the whole population.
- **SQ2:** *How does multicalibration affect discrimination?* The desired metric for discrimination in survival analysis is Harrell's C-index.
- **SQ3:** *How does the effect of multicalibration change if the initial survival model is trained on a data set skewed towards a majority population?*

Key Contributions. In this thesis, we present a novel method to mitigate unfairness in a survival setting. Thus, the key contributions are an interpretation of multicalibration as gradient boosting and the extension of multicalibration to survival analysis:

- First, we interpret the multicalibration framework as an adjusted version of gradient boosting that optimizes the Brier Score.

- Second, we present a possible adaption of the multicalibration framework with a modified loss function for survival analysis. To the best of our knowledge, there is no boosting approach that optimizes the censored Integrated Brier Score. Besides the theoretical framework, we provide an implementation in `R` and show empirically how multicalibration can influence the performance of a survival model.

Thesis Structure. The thesis is structured as follows. First, in **Chapter 1**, we introduce the theoretical problem of interest, motivate the proposed algorithm, and outline the main theoretical and practical contributions. **Chapter 2** provides relevant background knowledge and notation for fairness in machine learning, performance evaluation in survival analysis, gradient boosting, and multicalibration for binary classification. In **Chapter 3**, we present the adaption of the multicalibration framework to a survival setting. This modification includes a new notion of fairness, the corresponding boosting algorithm, and the implementation in R. In **Chapter 4**, we address the research question based on the implemented algorithm by conducting several experiments. After that, we show our experimental setup and the results. Finally, **Chapter 5** concludes this thesis with a discussion of the results and motivates future research.

2. Theoretical Background

First, we provide theoretical background and basic notation on fairness in machine learning, survival analysis, and gradient boosting. Second, we introduce the multicalibration framework in the context of boosting.

2.1. Fairness in Machine Learning

Fairness is an ubiquitous and relative term with many subjective and conflicting definitions of what it is, for whom, and how it should be achieved. Still, it is necessary to discuss the unintended impact of machine learning model predictions on society, as fairness can be defined and strived for within different contexts.

2.1.1. Defining Bias

Before we define fairness, we present the very central concept of biases in machine learning. The distinction between bias and fairness is ambiguous in the literature, and many authors use both terms synonymously. Therefore, we present the types and the sources of biases in the following.

Types of Biases. Mitchell et al. (2021) divide biases concerning data into statistical and societal biases. We transfer this concept to the whole model (and its prediction), as bias is introduced in the whole machine learning life cycle (Suresh & Guttag, 2019). As depicted in Figure 1, models try to represent the "world as it is." **Statistical biases** describe a systematic mismatch between the modeled and the real world. Most machine learning aims to minimize these biases. However, even if there is no statistical bias, algorithmic predictions might include **societal bias**: the decision does not meet the objectives of the decision-maker or a policy (e.g., equal positive rates between different ethnicities). These context-dependent objectives correspond to a "world as it should be." (Mitchell et al., 2021; Suresh & Guttag, 2019). As many different kinds of biases within statistical and societal bias exist (e.g., sampling bias, historical bias), we refer to Mehrabi et al. (2019) and Suresh and Guttag (2019) for an overview.

Sources of Biases. Biases can originate from the different steps of a machine learning modeling process. In Figure 1, a simplified version of the machine learning loop illustrates the four stages where biases are introduced according to Barocas et al. (2020): measurement, learning, action, and feedback. First, most biases are already included when gathering and **measuring** the data from the "world as it is." A typical bias in the data is the representation bias: the population in the data does not reflect the population subjected to the model (Suresh & Guttag, 2019). Second, also within the

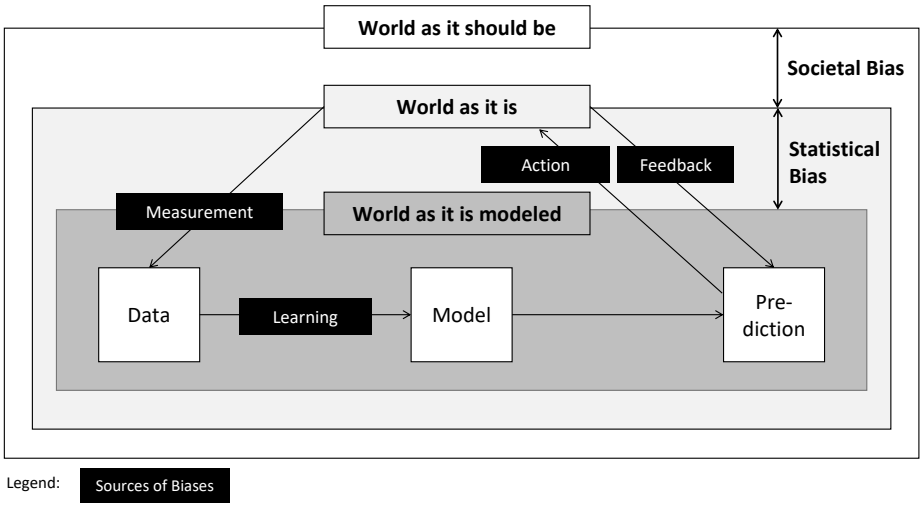

Figure 1: Types and sources of biases in machine learning partly based on Mitchell et al. (2021) and Barocas et al. (2020). Two types of biases exist: statistical and societal bias. In machine learning, we often focus on reducing the statistical bias, the systematic mismatch between the modeled and actual world. On the other hand, societal bias is the systematic mismatch between the modeled and an optimal world concerning ethical considerations. We can locate the sources of these biases in every step of the modeling process: measurement, learning, action, and feedback.

learning process, biases are included. For example, minority groups might be modeled less accurately or not at all, as the average error is minimized with respect to the whole population (Chouldechova & Roth, 2020). Third, **actions** based on automated decisions can also produce biases. For example, Bolukbasi et al. (2016) showed that word embeddings replicate stereotypes like "man is to computer programmer as a woman is to housewife." In automated systems, the application of learned connections may aggravate existing biases. For example, male candidates may be ranked higher than similarly competent female applicants in job searches that the system detects as male-associated. Lastly, **feedback** loops are a concern. Suppose we already have data biased towards a particular region in the data or model to detect criminal activities. Officers are more likely to patrol in these locations and confirm these patterns. As a result, predictions are skewed using data from targeted regions, and criminal behavior in regions is more likely to be predicted in these regions (Lum & Isaac, 2016).

2.1.2. Defining Fairness

The purpose of fairness metrics is to identify societal bias, or situations when the prediction differs from "the world as it should be" (Mitchell et al., 2021) or even amplifies the

difference between "the world as it is" and the "world as it should be" (feedback loops, action). Across different disciplines, fairness (or how "the world as it should be") can be approached from different research areas. However, within these disciplines, there is no common understanding of the definition of fairness.

Fairness in Machine Learning. Based on the ideas in these research areas, fairness has emerged as a concern in decisions based on algorithms. The **fairness** of a machine learning model is determined by a set of legal or ethical criteria that vary by country and culture (Fletcher et al., 2021) and aims to mitigate societal bias (Mitchell et al., 2021). Mehrabi et al. (2019) defined fairness in machine learning as *"the absence of any prejudice or favoritism towards an individual or a group based on their inherent or acquired characteristics. Thus, an unfair algorithm is one whose decisions are skewed toward a particular group of people."*

Roots of Fairness. As fairness definitions in machine learning are based on ideas in other research areas, we present the main ideas about fairness in other disciplines: in **law**, fairness means protecting individuals or groups from discrimination and maltreatment based on protected traits or social group categories. Thus, the main focus lies on some protected attributes defined in the law (e.g., gender or race). In **philosophy**, by contrast, the main focus lies on what morally correct decisions are. A field of philosophy, political philosophy, deals with fairness as justice and equity, i.e., how goods should be distributed. **Social science** focuses more on how members of particular groups (or identities) generally benefit from certain conditions (Mulligan et al., 2019). Finally, a mathematical perspective of fairness is taken by **quantitative disciplines** (e.g., mathematics, statistics, economics). Here fairness is assessed by a mathematically defined criterion like error rates or equitable allocation (Mulligan et al., 2019).

Fairness for what. Based on philosophical and legal ideas, we can distinguish between disparate treatment (treating individuals or groups similarly) and disparate outcome (having fair outcomes or results for individuals or groups; Gajane & Pechenizkiy, 2017). An example of disparate treatment is "fairness through unawareness" (Dwork et al., 2012). Due to it, a model is fair if we are not aware of protected attributes during modeling. However, this technique has proven ineffective in many instances, as protected variables may be correlated with not-protected factors (Gajane & Pechenizkiy, 2017). For example, living in a neighborhood within a city can be correlated with a particular ethnicity or religion. Therefore, we focus on the **disparate outcome**, including notions of group, individual, and subgroup fairness (Gajane & Pechenizkiy, 2017). Besides these central ideas, causal fairness is a considerable concern that urges the causal effect of

specific sensitive attributes to be fair (Bonchi et al., 2017; Chiappa, 2019; Kilbertus et al., 2017; Kusner et al., 2017; Kusner et al., 2019; Zhang & Bareinboim, 2018).

Fairness towards whom. Most research on fairness is conducted using statistical definitions (**group fairness**) calculated across a limited number of protected demographic categories (e.g., racial and gender). The advantage of this notion of fairness is that it is simple to obtain and easily verifiable. However, group fairness only ensures that the average members of these groups are protected, not individuals or so-called subgroups (Chouldechova & Roth, 2020). Furthermore, in contrast to group fairness, **individual fairness** requires that constraints be imposed on specific pairs of individuals (e.g., Dwork et al., 2012; Joseph et al., 2016). The disadvantage of individual fairness is that it is unclear whether individual notions of can be realized due to several obstacles (e.g., defining a similarity metric between individuals, Chouldechova & Roth, 2020; Dwork et al., 2012). **Subgroup fairness** resolves discrimination against overlapping subgroups and is often seen as ideal, as it combines group and individual fairness (Foulds et al., 2020). Subgroups are defined on a subset of the sensitive attributes. Nevertheless, it is associated with severe statistical and computational challenges, including data scarcity at the intersections of minority groups and an exponentially large number of subgroups (Yang et al., 2020).

Currently, literature in algorithmic fairness mainly deals with group fairness for a disparate outcome. There are at least over 18 different fairness measures; for an overview, we refer to current fairness literature (Barocas et al., 2020; Berk et al., 2018; Caton & Haas, 2020; Mehrabi et al., 2019; Verma & Rubin, 2018). We note that it is impossible to combine all fairness notions. Most notions are contradictory (e.g., be calibrated and have equal false positive and negative rates for all groups, Chouldechova, 2017; Kleinberg et al., 2017). Additionally, there is no single notion of fairness we can apply in every situation. It is up to the researcher or practitioner to assess their circumstances and prioritize their criteria (Makhlouf et al., 2020).

2.1.3. Achieving Fairness

As illustrated in Figure 1, there are two possibilities to reduce unfairness: having an actual world that is closer to the "world as it should be" and a model with low statistical bias (i.e., social solutions in the real world) or having machine learning models with low societal bias. In the following, we focus on the latter, technical solutions to the problem. Technically, we can reduce unfairness by adjusting the data (pre-processing), the model (in-processing), and the prediction (post-processing).

Pre-Processing. Usually, a major source of unfairness is the measurement of the data. Therefore, pre-processing methods alter protected variable sample distributions or perform data modifications to remove discrimination from training data (Caton & Haas, 2020; Kamiran & Calders, 2012).

In-Processing. In-processing methods emphasize that modeling techniques often get skewed by dominating features or other distributional effects. Therefore, these methods address unfairness by integrating fairness measures into model optimization functions to optimize performance and fairness. (Caton & Haas, 2020; Zafar et al., 2017)

Post-Processing. Post-processing typically involves modifying model predictions to improve accuracy. We only need access to the prediction and sensitive attribute data in post-processing, not the underlying machine learning model. Thus, post-processing is ideal if the whole machine learning process is unavailable and only black-box access to the model is needed. (Caton & Haas, 2020; Hardt et al., 2016; Hébert-Johnson et al., 2018)

2.2. Survival Analysis

In fairness, survival analysis has rarely been a concern (Keya et al., 2021), as most research focuses on binary classification tasks. In this section, we present the notation and the survival problem we are dealing with in the thesis. Additionally, we present how we evaluate survival models.

2.2.1. Notation and Survival Problem

In survival analysis, we define the random variable Y that describes the time until an event occurs, e.g., the time until recidivism or the death of a patient. The random variable Y is therefore non-negative, as it describes a duration. Survival analysis encompasses various tasks, including interval-censored data, time-varying effects, competing states, and time-varying features (Bender et al., 2020). We focus on a standard setting with right-censoring, one event of interest, and time-constant features.

Survival Function. The probability that an event has not occurred by a time point t is given by the survival function $S(t)$. Using the cumulative distribution function $F(t) = \mathbb{P}(Y \leq t)$, the survival function can be defined as

$$S(t) := \mathbb{P}(Y > t) = 1 - F(t). \tag{1}$$

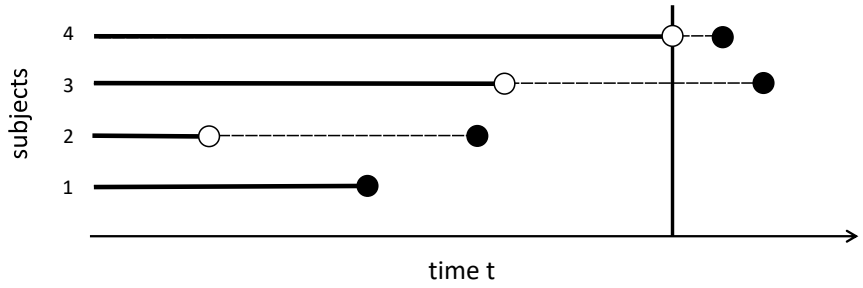

Figure 2: Censoring in right-censored data. On the horizontal axis is the time, and on the vertical axis are the subjects. The bold vertical line is the end of the study. The white circles mark the censoring time, and the black circles the true death time. Subject 1 is uncensored and dies within the observed time frame. Subjects 2-4 are censored: Subjects 2 and 3 drop out during the study time, and subject 4 drops out after the end of the study.

As the distribution function is monotonically increasing, the survival function is monotonically decreasing. The conditional survival function S is defined as the probability that the event has not occurred up to time point t given the features $\boldsymbol{x}$. We define the conditional distribution function over the survival times:

$$S(t \mid \boldsymbol{X} = \boldsymbol{x}) = \mathbb{P}(Y > t \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \mathbb{P}(Y \leq t \mid \boldsymbol{X} = \boldsymbol{x}). \tag{2}$$

In a setting without censoring, we observe the random variables of a p-dimensional feature vector and the survival time $(\boldsymbol{X}, Y) \subset \mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} \subseteq \mathbb{R}_0^+$ and $\mathcal{X} \subseteq \mathbb{R}^p$.

Censoring. Naturally, the time of an event may be unknown for various reasons, including the end of a study project or the drop-out of a subject from the study. As a consequence, the subject's data is incomplete (i.e., censored). Among different types of censoring (e.g., left-censoring or interval-censoring; Kalbfleisch & Prentice, 2002), most models assume right-censoring (Cox, 1972; Kaplan & Meier, 1958; Wang et al., 2019). Therefore, we follow this assumption. As illustrated in Figure 2, a subject is right-censored if a subject drops out of the study before the end of the study (subjects 2 and 3 in Figure 2) or if the event is not observed within the monitored time frame (subject 4 in Figure 2). Formally, a subject is right-censored if its actual event time Y is greater than the random variable $C \subset \mathcal{Y}$, which denotes the censoring time (i.e., the time point where the subject is not observed anymore).

Right-Censored Data. One challenge in survival is that we cannot observe the full data-generating process, as we can never observe both the censoring C and the event Y, only either-or. Instead we can examine the observed event time $\tilde{Y} = min(Y, C)$

and a censoring indicator $\Delta = \mathbb{I}(Y \leq C)$ with the indicator function $\mathbb{I}$. Formally, we observe the random variables $(\boldsymbol{X}, \tilde{Y}, \Delta) \subset \mathcal{X} \times \mathcal{Y} \times \{0, 1\}$. We assume that the actual survival time Y and the censoring time C are conditionally independent given $\boldsymbol{X}$ (i.e., conditionally event-independent).

Survival Task. As a result of the differences between the data generating process (Y and C) and the observed variables ($\tilde{Y}$ and Δ), there are different survival problems modeled, as the representation of Y can differ between models. According to Haider et al. (2020), survival predictions can have the following characteristics:

(1) predicting **probabilities** $p \in [0, 1]$ or real-valued ranking **scores** $s \in [-\infty, \infty]$ which are not meaningful themselves,

(2) having a **functional** response over time or a **scalar** response where survival models have a scalar prediction for a particular time point or time-independently, and

(3) making **individual** predictions (for every feature vector $\boldsymbol{x}$) or for **populations** (e.g., Kaplan-Meier models; Kaplan & Meier, 1958).

For further details on the types of survival models, we refer to Haider et al. (2020). For effective decision-making, a probability distribution over the remaining time-to-death for every subject is desirable (**individual survival distributions**; Avati et al., 2019; Haider et al., 2020):

$$h : \mathcal{X} \to \mathcal{L}^2(\mathcal{T}, [0, 1]), \tag{3}$$

where $\mathcal{T} = [t_1, t_2]$ with $t_1, t_2 \in \mathbb{R}$. Individual survival distributions map a feature vector $\boldsymbol{x} \in \mathcal{X}$ to a probability distribution over $\mathcal{T}$.

2.2.2. Evaluation Measures

In general, we measure how much the predicted value $h(\boldsymbol{x})$ reflects the observed true label y evaluating a survival model. As illustrated in Figure 3, performance measurement in survival includes two concepts: discrimination and calibration (Steyerberg et al., 2004). **Discrimination** indicates if a model can distinguish the event of two classes properly. For example, in the context of survival analysis, a model with good discrimination assigns lower survival probabilities to subjects where time to the event is low and higher survival probabilities to a subject where the time to the event is high (D'Agostino & Nam, 2003). **Calibration** measures how numerically close the predicted probability is to the true probability (D'Agostino & Nam, 2003). We can achieve both independently of each other. However, a good performance in discrimination as well as in calibration is desirable.

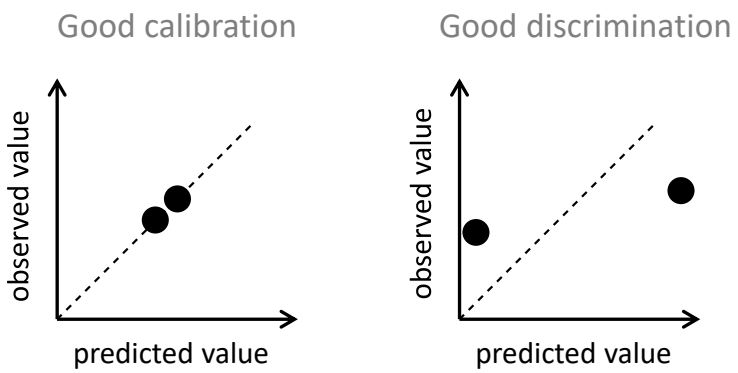

Figure 3: Calibration and discrimination. On the x-axis is the predicted value and on the y-axis is the observed value. The black dots illustrate two predictions. The left figure depicts a model with good calibration (i.e., the predicted value equals the observed value). The figure on the right side shows the predictions of a model with good discrimination (i.e., the outcome classes are separated well and have a good ranking).

Loss Function. Analogous to Brockhaus et al. (2017), we can define a loss function

$$\rho : (\mathcal{Y} \times \mathcal{X}) \times \mathcal{H} \to \mathcal{L}^1(\mathcal{T}, \mu) \tag{4}$$

mapping the data $(y, \boldsymbol{x})$ and the model h to a function in the space of integratable functions $\mathcal{L}^1(\mathcal{T}, \mu)$. To formulate it differently, ρ maps the data and the model to a function that measure the difference between $Y(t)$ and $h(\boldsymbol{x})(t)$ for each $t \in \mathcal{T}$. For better readability, we omit t in the following.
To obtain a real-valued loss, we define a loss function $\ell : (\mathcal{Y} \times \mathcal{X}) \times \mathcal{H} \to \mathbb{R}$ by integrating the loss function ρ

$$\ell((y, \boldsymbol{x}), h) = \int \rho((y, \boldsymbol{x}), h) d\mu, \tag{5}$$

where μ is the Lebesgue measure for a functional response and the Dirac measure for a scalar response.

Inverse Probability of Censoring Weights. If the data includes censored observations, we cannot evaluate the loss $\ell((y, \boldsymbol{x}), h)$, as we can only observe $(\boldsymbol{X}, \tilde{Y}, \Delta)$ and thus can evaluate the loss just indirectly. One possibility to solve this discrepancy between modeled and observed data is the application of inverse probability of censoring weights (IPCW; van der Laan & Robins, 2003):

$$\ell((\tilde{y}, \boldsymbol{x}), h \mid G) = \ell((\tilde{y}, \boldsymbol{x}), h) \cdot \frac{\delta}{G(\tilde{y} \mid \boldsymbol{x})}, \tag{6}$$

with

$$\frac{\delta}{G(\tilde{y} \mid \boldsymbol{x})} = w\left(t\right) = \begin{cases} 1/G(t \mid \boldsymbol{x}), & \text{if } t < \tilde{y} \\ 1/G(\tilde{y} \mid \boldsymbol{x}), & \text{if } t \geq \tilde{y} \text{ and } \delta = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $G(c \mid \boldsymbol{x}) = \mathbb{P}(C > c \mid \boldsymbol{x})$ is a conditional censoring function. By this weighting, subjects with a high censoring probability have less influence on an expected loss over a sample and vice versa. Subjects after their censoring time (i.e., $t \geq \tilde{y}$ and $\delta = 0$) have an unknown status, are excluded from the calculation.

Evaluating Calibration. In the survival context, so far, there is no widely accepted measure to evaluate the calibration of a survival distribution (Avati et al., 2019). Nevertheless, in the literature, the (Integrated) Brier Score is often used to evaluate the calibration in survival analysis (Kvamme et al., 2019; Lee et al., 2020; Murphy, 1973). We define the **Brier Score** (Brier, 1950; Mogensen et al., 2012) in survival analysis as

$$\rho_{BS}((y \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x})) = \frac{1}{2}(h(t \mid \boldsymbol{x}) - \mathbb{I}[y < t])^2 \in [0, 1]. \tag{8}$$

The aim is to minimize the Brier Score. The **censored Integrated Brier Score** (IBS, Integrated Graf Score; Gerds & Schumacher, 2006; Graf et al., 1999) can evaluate the Brier Score on all time points and for censored observations (IPCW):

$$\ell_{BS}^{C}((\tilde{y} \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x})) = \int w\left(t\right) \cdot \rho_{BS}((\tilde{y} \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x}))d\mu(t) \in [0, 1], \tag{9}$$

where $w\left(t\right)$ denotes the censoring weights according to IPCW weighting (7). Recently, Haider et al. (2020) proposed D-calibration to evaluate the calibration of a survival distribution. Notwithstanding, we do not focus on D-calibration, as it assumes that the times after censoring are uniformly distributed. As a result, the assessment can be too optimistic in a setting with many censored observations (Avati et al., 2019).

Evaluating Discrimination. The most common (discrimination) measure in survival analysis is the Harrell's concordance index (**C-index** or C-statistics; Gerds et al., 2013; Harrell et al., 1982). It measures the model's ability to rank the survival times based on individual survival probability. Hence, it is defined as the proportion of pairs that are properly ordered (concordant) to pairs that are comparable. A pair (i, j) is concordant if $h\left(\boldsymbol{x}_i\right) < h\left(\boldsymbol{x}_j\right)$ and it is comparable, if $\tilde{y}_i < \tilde{y}_j$ and $\delta_i = 1$. Thus, we can write the

C-index as:

$$C := \frac{\sum_{i \neq j} \mathbb{I}\left(\tilde{y}_i < \tilde{y}_j, h\left(\boldsymbol{x}_i\right) < h\left(\boldsymbol{x}_j\right), \tilde{y}_i < \tau\right) \delta_i}{\sum_{i \neq j} \mathbb{I}\left(\tilde{y}_i < \tilde{y}_j, \tilde{y}_i < \tau\right) \delta_i} \in [0, 1], \tag{10}$$

where τ is the cut-off time. The aim is to maximize the C-index. For further details, we are referring to Harrell et al. (2005) and Penciana and D'Agostino (2004), and Uno et al. (2011).

2.3. Gradient Boosting

Gradient boosting is a machine learning method that combines multiple weak learners b into a single strong learner f. Given the connection between multicalibration boosting and our application to survival analysis, we provide a brief overview of gradient boosting in classification and survival analysis. Additionally, we present the related boosting formulation AnyBoost, which serves as the basis for the definition of multicalibration and the formulation of multicalibration boosting stopping criteria.

Idea. In the gradient boosting (Friedman, 2001), we learn a linear combination of a class of base learners $\mathcal{B}$:

$$f(\boldsymbol{x}) := \left(\sum_{m=1}^{M} \eta^{[m]} b^{[m]}(\boldsymbol{x}) \right), \tag{11}$$

where $b^{[m]}(\boldsymbol{x}) \in \mathcal{B}$ is the **base learner** and $\eta^{[m]}$ the corresponding learning rate. The most popular choice for weak learners are classification and regression trees (CART, Breiman et al., 1984). We can formulate the optimization problem in boosting with a loss function ρ as (Brockhaus et al., 2017; Hothorn et al., 2014):

$$\begin{aligned} f^* &= \arg\min_f \mathbb{E}_{Y,\boldsymbol{X}} \rho((Y, \boldsymbol{X}), f) \\ &= \arg\min_f \int \rho((y, \boldsymbol{x}), f) d\mathbb{P}_{Y,\boldsymbol{X}}(y, \boldsymbol{x}). \end{aligned} \tag{12}$$

We randomly sample N points i.i.d. from a joint distribution of the target Y and the features $\boldsymbol{X}$ (i.e., $(Y_i, \boldsymbol{X}_i) \sim \mathbb{P}_{Y,\boldsymbol{X}}, i = 1, \ldots, N$). In boosting, we minimize the empirical risk where we define $\hat{\mathbb{P}}_{Y,\boldsymbol{X}}(y, \boldsymbol{x})$ as a empirical distribution which puts weight mass $w_i = \frac{1}{N}$ on an observation i (Hothorn et al., 2014):

$$\begin{aligned} f^* &= \arg\min_{f \in \text{lin}(\mathcal{B})} \int \rho((y, \boldsymbol{x}), f) d\hat{\mathbb{P}}_{Y,\boldsymbol{X}}(y, \boldsymbol{x}) \\ &= \arg\min_{f \in \text{lin}(\mathcal{B})} \left\{ \frac{1}{N} \sum_{i=1}^{N} \rho\left(y_i, f\left(\boldsymbol{x}_i\right)\right) \right\}. \end{aligned} \tag{13}$$

Algorithm. To minimize the empirical risk (13), gradient boosting takes an approximated steepest descent step: starting with a loss-optimal constant model $\hat{f}^{[0]}(\boldsymbol{x})$, the gradient boosting algorithm calculates in every step m the so-called **pseudo-residuals** $\tilde{r}^{[m]}(\boldsymbol{x})$, which are the negative gradient of the loss ρ with respect to the model f evaluated at the current model $\hat{f}^{[m]}(\boldsymbol{x})$. Then the weak learner is fitted to the pseudo-residuals by minimizing the quadratic loss. The weak learner approximates the steepest steps (the pseudo-residuals). This weak learner is added via line search or by a very small constant $\eta^{[m]}$. We repeat these steps for M repetitions. Usually, we should stop the gradient boosting algorithm early to achieve good predictive performance (Bühlmann & Hothorn, 2007; Friedman, 2001; Zhang & Yu, 2005).

Algorithm 1 Gradient Boosting Algorithm.

1: Initialize $\hat{f}^{[0]}(\boldsymbol{x}) = \arg\min_{b \in \mathcal{B}} \sum_{i=1}^{N} \rho(y^{(i)}, b(\boldsymbol{x}^{(i)}))$
2: **for** $m = 1 \to M$ **do**
3: Calculate pseudo-residuals: $\tilde{r}^{[m]}(\boldsymbol{x}) = -\left[\frac{\partial \rho((y, \boldsymbol{x}), f)}{\partial f}\right]_{f=\hat{f}^{[m-1]}}$
4: Fit a regression base learner $b^{[m]}(\boldsymbol{x})$ to the pseudo-residuals $\tilde{r}^{[m]}$:
5: $b^{[m]}(\boldsymbol{x}) = \arg\min_{b \in \mathcal{B}} (\tilde{r}^{[m]} - b^{[m]}(\boldsymbol{x}))^2$
6: Update $\hat{f}^{[m]}(\boldsymbol{x}) = \hat{f}^{[m-1]}(\boldsymbol{x}) + \eta^{[m]} \cdot b^{[m]}(\boldsymbol{x})$
7: **end for**
8: Output $\hat{f}(\boldsymbol{x}) = \hat{f}^{[M]}(\boldsymbol{x})$

Gradient Boosting in Binary Classification. In gradient boosting for classification, we keep the probabilities $\pi(\boldsymbol{x})$ within a range of $[0, 1]$ by passing the unnormalized scores into a sigmoid function. Taking the $y - \pi(\boldsymbol{x})$ as pseudo-residuals equals optimizing the Log-Loss (Good, 1952):

$$\rho_{log}((y, \boldsymbol{x}), f) = -yf(\boldsymbol{x}) + \ln(1 + \exp(f(\boldsymbol{x}))) \tag{14}$$

$$\begin{aligned} \tilde{r}(\boldsymbol{x}) &= -\left[\frac{\partial}{\partial f} \rho_{log}((y, \boldsymbol{x}), f)\right]_{f=\hat{f}^{[m-1]}} \\ &= \left[y - \underbrace{\frac{1}{1 + \exp(-f(\boldsymbol{x}))}}_{\pi(\boldsymbol{x})}\right]_{f=\hat{f}^{[m-1]}} \end{aligned} \tag{15}$$

The complete calculation can be found in Appendix A.

Gradient Boosting in Survival Analysis. First, Ridgeway (1999) considered modeling survival data with gradient boosting. In the literature, boosting in survival analysis is mainly adapted for optimizing specific survival models like a Cox proportional hazards model (Binder & Schumacher, 2008) or accelerated failure time model (Schmid & Hothorn, 2008; Wang & Wang, 2010), discrimination in survival models (Mayr & Schmid, 2014) or based on component-wise boosting (Hofner et al., 2014). The Integrated Brier Score has not been considered so far. To the best of our knowledge, boosting in fairness has not been considered for survival analysis nor subgroup fairness, but rather for binary classification (Iosifidis & Ntoutsi, 2019; Vargo et al., 2021) or in deep learning approaches (Avati et al., 2019; Kamran & Wiens, 2021).

AnyBoost. Based on the boosting formulation in Mason et al. (2000), lines 2 and 3 in Algorithm 2 can be seen as **functional gradient descent**. Instead of minimizing the least-squares error of the base learner $b(\boldsymbol{x})$ and the pseudo-residuals $\tilde{r}$, Mason et al. (2000) fit the base learner by maximizing the negative inner product of the base learner and the gradient (negative pseudo residual) $\langle U(\boldsymbol{x}), b(\boldsymbol{x})\rangle$ with $U(\boldsymbol{x}) = -\tilde{r}(\boldsymbol{x})$. It can be shown that both formulations are equivalent:

$$\arg\max_{b \in \mathcal{B}} -\langle U(\boldsymbol{x}), b(\boldsymbol{x})\rangle = \arg\min_{b \in \mathcal{B}} \sum_{i=1}^{N} \left(b(\boldsymbol{x}_i) - \tilde{r}(\boldsymbol{x_i})\right)^2.$$

For further details, we refer to Appendix A. In AnyBoost, we also have additional stop criteria within the boosting algorithm based on the inner product. If the inner product of the base learner and the gradient $\langle -U(\boldsymbol{x}), b(\boldsymbol{x})\rangle \leq 0$, the algorithms stops. If the stop criterion holds, both vectors (the gradient and the base learner) are orthogonal in the functional space, i.e., the base leaner does not explain the gradient. The smaller the inner product is, the less the base learner has a correct direction in the functional space.

Algorithm 2 AnyBoost.

1: Initialize $\hat{f}^{[0]}(\boldsymbol{x})$
2: **for** $m = 1 \to M$ **do**
3: Calculate gradients: $U^{[m]}(\boldsymbol{x}) = -\left[\frac{\partial \rho((y, \boldsymbol{x}), f)}{\partial f}\right]_{f=\hat{f}^{[m-1]}}$
4: Fit a regression base learner $b^{[m]}(\boldsymbol{x})$ to the gradients $U^{[m]}$.
5: $b^{[m]}(\boldsymbol{x}) = \arg\max_{b \in \mathcal{B}} -\langle U^{[m]}(\boldsymbol{x}), b^{[m]}(\boldsymbol{x})\rangle$
6: **if** $-\langle U^{[m]}(\boldsymbol{x}), b^{[m]}(\boldsymbol{x})\rangle \leq 0$ **then**
7: **return** $\hat{f}^{[m]}$
8: **end if**
9: Update $\hat{f}^{[m]}(\boldsymbol{x}) = \hat{f}^{[m-1]}(\boldsymbol{x}) + \eta^{[m]} \cdot b^{[m]}(\boldsymbol{x})$
10: **end for**
11: Output $\hat{f}(\boldsymbol{x}) = \hat{f}^{[M]}(\boldsymbol{x})$

2.4. Multicalibration in Binary Classification

Hébert-Johnson et al. (2018) presented the multicalibration framework, which was followed by multiaccuracy (Kim et al., 2019). Multicalibration includes a fairness definition (Section 2.4.2) that urges a model to be α-calibrated (multicalibration) or α-unbiased (multiaccuracy) for subgroups and provides a post-processing fairness method (Section 2.4.3), i.e., improving a trained model, which aims to achieve this subgroup fairness definition based on boosting.

2.4.1. Setting and Assumptions

In multicalibration, we evaluate and post-process a binary classification task in order for it to be calibrated for subgroups $\mathcal{S}$ (subgroup fairness). Suppose we have a population of N individuals in $\mathcal{X}$ and want to predict an outcome $y \in \{0,1\}^N$. p_i^* is the probability that the outcome y_i of an individual $\boldsymbol{x}_i$ with $i \in 1, ..., N$ is 1. A predictor $f : \mathcal{X} \to [0,1]$ estimates the mapping from an in individual $\boldsymbol{x} \in \mathcal{X}$ to the true parameter. The aim is to evaluate the fitness of f not only with respect to $\mathcal{X}$, but also with respect to subgroups $\mathcal{S} \subseteq \mathcal{X}$, where $\mathcal{C}$ is a collection of all subgroups. We use this notation in the following: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \mathbb{E}_{i \sim \mathcal{D}} [\boldsymbol{x}_i \cdot \boldsymbol{y}_i]$.

2.4.2. Defining Multicalibration in Binary Classification

Multicalibration. Multicalibration (Hébert-Johnson et al., 2018) determines if a binary classification model is calibrated for all subgroups $\mathcal{S}$ of a collection of subgroups $\mathcal{C}$. First, we define that a model f is α**-accurate-in-expectation** (α-AE) for a subgroup $\mathcal{S}$, if

$$| \mathop{\mathbb{E}}_{i \sim \mathcal{S}} [f(\boldsymbol{x}_i) - p_i^*] | \leq \alpha, \tag{16}$$

where we relax the expectation with a small α-bound ($\alpha > 0$) in a scenario with overlapping subgroups. α-calibration requires α-AE not in expectation over all predicted, but for all values $v \in [0,1]$. A stronger requirement is α**-calibration** for a subgroup $\mathcal{S}$. It implies that the average of the actual probabilities of the subjects getting prediction v is α-close to v for all but an α-fraction of a set $\mathcal{S}$. Multicalibration requires a binary classifier to be α-calibrated for every subgroup $\mathcal{S} \in \mathcal{C}$. Therefore, Hébert-Johnson et al. (2018) defines that a predictor f is $(\mathcal{C}, \alpha)$-**multicalibrated** if for any $v \in [0,1]$ and $\alpha \in [0,1]$

$$\left| \mathop{\mathbb{E}}_{i \sim \mathcal{S}_v \cap \mathcal{S}'} [f(\boldsymbol{x}_i) - p_i^*] \right| \leq \alpha \quad \forall\, \mathcal{S} \in \mathcal{C}, \tag{17}$$

if there exists some $\mathcal{S}' \subseteq \mathcal{S}$ with $\mathbb{P}_{i \sim \mathcal{D}} [i \in \mathcal{S}'] \geq (1 - \alpha) \cdot \mathbb{P}_{i \sim \mathcal{D}} [i \in \mathcal{S}]$ and where $\mathcal{S}_v = \{i : f(\boldsymbol{x}_i) = v\}$. However, the multicalibration definition is not computationally feasible,

as it requires the predictor f to be (1) α-calibrated for every $v \in [0,1]$, (2) defined on the true probabilities p^* which we have no direct access to, (3) measured with respect to all possible subgroups, which is information-theoretically not feasible for a small $\mathcal{D}_{val}$.

Empirical Multicalibration. Hébert-Johnson et al. (2018) propose the following modifications to measure multicalibration in a practical setting: (1) introduce discretization of v (bucketing), (2) redefining the residuals, and (3) optimizing subgroup fairness for efficiently-identifiable subgroups.

(1) **Bucketing**: Hébert-Johnson et al. (2018) introduce λ-discretization (buckets), where we divide $v \in [0,1]$ in λ equally spaced buckets. They define buckets $\Lambda[0,1]$ that are denoted by

$$\Lambda[a,b] = \left\{ a + (b-a)\frac{\lambda}{2}, a + (b-a)\frac{3\lambda}{2}, \ldots, b - \frac{\lambda}{2}(b-a) \right\} \tag{18}$$

and

$$\lambda(v) = \left[v - \frac{\lambda}{2}(b-a), v + \frac{\lambda}{2}(b-a) \right) \tag{19}$$

as the λ-interval centered around v (only the last interval is $[b - \lambda(b-a), b]$). $(\mathcal{C}, \alpha, \lambda)$-multicalibrated predictor f with $\alpha \in [0,1]$ and $\lambda > 0$ can therefore be defined on the corresponding discrete buckets with $\mathcal{S}_v(f) = \{i : f(\boldsymbol{x}_i) \in \lambda(v)\} \cap \mathcal{S}$ for all $\mathcal{S} \in \mathcal{C}$ and $v \in \Lambda[0,1]$.

(2) **Residuals**: In practice, we only have access to the true outcome y and not the true probability p^*. Suppose we assume that the samples in $\mathcal{D}_{val}$ are large enough. In this case, the empirical expectation of y in a bucket corresponds to the expectation of the true probability p^* of a bucket. For each $\boldsymbol{x} \in \mathcal{X}$, let the residual be $U(\boldsymbol{x}) = f(\boldsymbol{x}) - y$. Later, we show that this residual corresponds to the gradient $U(\boldsymbol{x})$ in boosting.

(3) **Efficiently-identifiable subpopulations** $\mathcal{C}$: Instead of optimizing multicalibration with respect to all possible definable subgroups, we want to measure multicalibration with respect to all "efficiently-identifiable subpopulations." In this case, $\mathcal{C}$ can be any class of regression algorithm fitted on the residuals $U(\boldsymbol{x})$ (e.g., decision tree regression or ridge regression). We now measure multicalibration with respect to all subgroups, which can be identified by this regression learner ("efficiently-identifiable"). As a consequence, potentially also subgroups can be defined on other features than in a fixed setting. If $\mathcal{C}$ is a class of weak learners like in gradient boosting, multiaccuracy is equivalent to the stopping criteria in AnyBoost (Mason et al., 2000).

In this case, we can rewrite the definition of multicalibration to

$$|\langle c_v(\boldsymbol{x}), U(\boldsymbol{x})\rangle| \leq \alpha \quad \forall\, c \in \mathcal{C}, v \in \Lambda[0,1], \tag{20}$$

where $c_v(\boldsymbol{x}) \in \mathcal{C}$ is a learner fitted on the residuals where $i \in \lambda(v)$. Consequently, a predictor f is $(\mathcal{C}, \alpha.\lambda)$-multicalibrated, if the mean residual $U(\boldsymbol{x}_i)$ in every "efficiently-identifiable" subgroup $c(\boldsymbol{x}) \in \mathcal{C}$ and every bucket $v \in \Lambda[0,1]$ is smaller than α.

A low scalar product $|\langle c(\boldsymbol{x}), U(\boldsymbol{x})\rangle|$ between the learned function $c(\boldsymbol{x})$ and the residuals $U(\boldsymbol{x})$ can be interpreted as a low correlation between the subgroup and the bias. Geometrically, this means that the two vectors $c(\boldsymbol{x})$ and $U(\boldsymbol{x})$ are almost orthogonal, and the subgroups or the learned class cannot explain the residuals.

Multiaccuracy. Kim et al. (2019) defined with $(\mathcal{C}, \alpha)$-multiaccuracy a relaxed version of multicalibration where the empirical version of multicalibration with only **one bucket** (i.e., $\lambda = 1$) is optimized (20). Hence, multiaccuracy corresponds to achieving α-AE (16) in every subgroup. In contrast to other subpopulation post-processing methods (e.g., Kearns et al., 2018), multiaccuracy guarantees that the improvement in subgroups does not lower the performance in the already well-predicted larger subgroups much (i.e., "do-no-harm guarantee"). Empirically, Kim et al. (2019) showed that multiaccuracy could also improve the overall accuracy of a classification model.

Related Group Fairness Metrics. Subgroup fairness metrics can be interpreted as group fairness metrics applied to subgroups with an α-bound. For example, multicalibration is a subgroup extension of the group-fairness criteria test fairness (well-calibrated; Chouldechova, 2017; Kleinberg et al., 2017). Well-calibration requires a learner to be calibrated in all groups and, consequently, a probability has the same meaning for all groups. Multiaccuracy can be interpreted as a subgroup version of predictive parity (Chouldechova, 2017; Simoiu et al., 2017), which measures if the positive predictive value (i.e., the probability that the prediction of 1 is true) is the same across groups.

2.4.3. Achieving Multicalibration in Binary Classification

To achieve multicalibration, Hébert-Johnson et al. (2018) and Kim et al. (2019) propose a **post-processing boosting algorithm** that has black-box access to an existing model $\hat{f}^{[0]}$. To post-process the model, we have access to only a relativity small validation data set $\mathcal{D}_{val}$. We assume that the validation data is not biased and has a sufficient representation of all subgroups. Thus, the multicalibration algorithm performs a variant of gradient boosting on an existing model to achieve unbiased results. However, a novelty

about this post-processing approach is that it minimizes the Brier Score (21), introduces buckets defined on the predictions, and performs a multiplicative update.

Algorithm 3 Multicalibration for Binary Classification.

1: Take the trained model: $\hat{f}^{[0]}(\boldsymbol{x})$.
2: Create buckets $v \in \Lambda[0,1]$
3: **for** $m = 0 \rightarrow M-1$ **do**
4: Calculate gradients: $U^{[m]}(\boldsymbol{x}) = -\left[\frac{\partial \rho((y,\boldsymbol{x}),f)}{\partial f}\right]_{f=\hat{f}^{[m-1]}}$
5: Fit a regression learner $c_v^{[m]}(\boldsymbol{x})$ to the gradients $U^{[m]}$ on every bucket $v \in \Lambda[0,1]$.
6: Take the bucket with the largest correlation: $v^* = \arg\max_{\lambda(v)} \left|\left\langle c_v^{[m]}(\boldsymbol{x}), U^{[m]}(\boldsymbol{x})\right\rangle\right|$
7: Check if $\hat{f}^{[m]}$ is already $(\mathcal{C},\alpha)$-multicalibrated:
8: **if** $\left|\left\langle c_{v^*}^{[m]}(\boldsymbol{x}), U^{[m]}(\boldsymbol{x})\right\rangle\right| \leq \alpha$ **then**
9: **return** $\hat{f}^{[m]}$
10: **end if**
11: Multiplicatively update $\hat{f}^{[m+1]}(\boldsymbol{x}) = exp(-\eta^{[m]} \cdot c_{v^*}^{[m]}(\boldsymbol{x})) \cdot \hat{f}^{[m]} \quad \forall x \in \lambda(v^*)$
12: Project $\hat{f}^{[m+1]}(\boldsymbol{x})$ onto $[0,1]$
13: **end for**
14: Output $\hat{f}(\boldsymbol{x}) = \hat{f}^{[M]}(\boldsymbol{x})$

Algorithm. In Algorithm 3, the pseudocode of multicalibration is depicted. Instead of starting with a loss-optimal model, the post-processing method starts with the trained model $\hat{f}^{[0]}$. Before this model can be multicalibrated, the strategy for bucketing must be set. This strategy includes how many and which buckets are to be used. After initialization, the initial trained model $\hat{f}^{[0]}$ is "nudged" in M iterations towards a multicalibrated model. In every iteration, the gradient (negative pseudo-residual) is calculated. The gradient is the loss function derived with respect to the the current model $\hat{f}^{[m]}$ for every data point $(\boldsymbol{x}, y)$. Afterwards, a regression learner is fitted on the residuals on every bucket. The correlation between the gradient and the learner is calculated, and the bucket with the highest correlation v^* is chosen. The algorithm stops early, if $\hat{f}^{[m]}$ is already $(\mathcal{C},\alpha)$-multicalibrated (corresponds to the stopping criterion in AnyBoost). In any other case, the model is updated for all values in bucket v^*. The update in multicalibration can be additive and multiplicative. They are clipped if the predictions are out of the desired range of $[0,1]$.

Bucketing. In multicalibration, we often choose ten buckets or a $\frac{1}{10}$-discretization (Barda et al., 2021). Multiaccuracy (Kim et al., 2019) is defined on one bucket (1-discretization). Nonetheless, Kim et al. (2019) propose to choose three buckets in the algorithm: one bucket over the whole population $\lambda(0.5) = [0,1]$ and two buckets with $\frac{1}{2}$–discretization (i.e., $\lambda(0.25) = [0, 0.5)$ and $\lambda(0.75) = [0.5, 1]$).

Gradient. The loss function, which is minimized in multicalibration, is the **Brier Score** (Brier, 1950) and not the Log-Loss, as proposed in gradient boosting:

$$\rho_{Brier}((y, \boldsymbol{x}), f) = \frac{1}{2}(y - f(\boldsymbol{x}))^2. \tag{21}$$

Usually, we choose the Log-Loss as a loss function for optimizing binary classification tasks, as we naturally keep the probabilities within a range of $[0, 1]$ with the sigmoid function. However, by using the Brier Score, multicalibration does not perform boosting on the probabilities $p \in [0, 1]$ but on the scores $s \in [-\infty, \infty]$. As a consequence, predicted probabilities can be out of the range $[0, 1]$. Therefore, Hébert-Johnson et al. (2018) propose to clip the predicted probabilities to [0,1]. We calculate the gradient in multicalibration (residual) in each iteration as follows:

$$\begin{aligned} U^{[m]}(\boldsymbol{x}) &= -\left[\frac{\partial \rho((y, \boldsymbol{x}), f)}{\partial f}\right]_{f=\hat{f}^{[m-1]}} \\ &= \left[\frac{\partial \frac{1}{2}(y - f(\boldsymbol{x}))^2}{\partial f}\right]_{f=\hat{f}^{[m]}} \\ &= \left[f(\boldsymbol{x}) - y\right]_{f=\hat{f}^{[m]}}. \end{aligned} \tag{22}$$

In this case, the gradient is equal to the residual.

Implementation. The multicalibration framework is implemented in the `R` package `mcboost` (Pfisterer et al., 2021).

3. Adapting the Multicalibration Framework to Survival Analysis

The multicalibration framework has already been adapted to survival tasks: Barda et al. (2020) perform distribution transfer from a baseline model for respiratory infection risk to an accurate COVID-19 mortality prediction model by boosting the model with the death rates in the subpopulations. Barda et al. (2021) analyze two medical risk assessment models for calibration in subpopulations. These models are based on survival models but only consider a risk for a fixed time point (i.e., the 10-year risk for osteoporotic fractures). It shows that post-processing for multicalibration can considerably enhance calibration metrics. However, to the best of our knowledge, the multicalibration framework has not been adapted to a survival loss function or a survival distribution.

3.1. Defining Multicalibration in Survival Analysis

Survival Problem. As mentioned before (see page 10), there are several types of survival problems (Haider et al., 2020). However, in the following, we only consider individual survival distributions that predict survival probabilities over time (3):

(1) We use **probabilities**, as risk scores cannot be evaluated concerning their calibration unless they are transformed to survival probabilities.

(2) Using **distributions over time** is a generalization of a single prediction per subject, we have to set $\mathcal{T}$ to $[t, t]$ to obtain a prediction for a single time point t. Barda et al. (2021), Barda et al. (2020) have only used a single time point in their applications of multicalibration.

(3) Multicalibrating groups instead of **individuals** contradict the idea of overlapping subgroups if the model already defines disjoint groups. Nevertheless, groups can be treated as individuals.

Therefore, we conduct the adaption of the multicalibration framework to individual survival distributions.

Adaption of the Loss Function. To adapt the multicalibration definition (20) from a binary classification to a survival task, we have to change the optimized loss function. For binary classification, the optimized loss is the Brier Score (22). However, if we deal with right-censored data, we should incorporate in the loss that we cannot observe the complete data-generating process (IPCW). The **censored Brier Score** (8) resolves this as a natural survival equivalent and common calibration measure. Following the literature, we estimate the censoring distribution $\hat{G}$ with a Kaplan-Meier model (Graf

et al., 1999; Kaplan & Meier, 1958) where we take $\delta_{cens} = 1 - \delta$. By this change in the data, we model the probability of being censored up to a time point C and not the survival probability Y. Additionally, the **censored Integrated Brier Score** (9) can measure the calibration for distribution over time.

Adaption of the Gradient. The aim of boosting is to optimize the empirical risk for function h (13). For individual survival distributions, we can specify the corresponding empirical risk function defined by the data, which is

$$\begin{aligned}\hat{\mathbb{E}}_{Y,\boldsymbol{X}}\ell((Y,\boldsymbol{X}),h) &= \iint \rho((y \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x}))d\mu(t)d\hat{\mathbb{P}}_{Y,\boldsymbol{X}}(y,\boldsymbol{x}) \\ &= \int \underbrace{\int \rho^C((\tilde{y} \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x}) \mid G)d\mu(t)}_{\ell^C} d\hat{\mathbb{P}}_{\tilde{Y},\boldsymbol{X},\Delta}(\tilde{y},\boldsymbol{x},\delta).\end{aligned} \tag{23}$$

We take a sample of N independent and identically distributed observations from the joint distribution of the observed event time $\tilde{Y}$, the features $\boldsymbol{X}$ and the censoring indicator Δ (i.e., $(\tilde{y}_i, \boldsymbol{x}_i, \delta_i) \sim \mathbb{P}_{\tilde{Y},\boldsymbol{X},\Delta}$, for $i \in \{1,\ldots,N\}$). To approximate the empirical risk, we define $\hat{\mathbb{P}}_{\tilde{Y},\boldsymbol{X},\Delta}$ which puts the mass $\frac{1}{N}$ on every observation. To approximate the measure μ, we use a discrete uniform distribution $\hat{\mu}$. By this approximation, we put a mass of $\frac{1}{L}$ on an equidistant grid $t_1 < t_2 ... < t_L$ with a sufficiently large quantity of grid points L on the target space. This approach is analogous to Hothorn et al. (2014). Consequently, the approximated expected loss can be defined as

$$\hat{\mathbb{E}}_{Y,\boldsymbol{X}}\ell((Y,\boldsymbol{X}),h) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\rho^C\left((\tilde{y}_i \leq t_l, \boldsymbol{x}_i), h\left(t_l \mid \boldsymbol{x}_i\right) \mid G\right). \tag{24}$$

Now, we can evaluate the loss function ρ at every single observation $(\tilde{y}_i \leq t_l, \boldsymbol{x}_i)\, \forall i \in \{1,\ldots,N\}, l \in \{1,\ldots,L\}$.

$$\begin{aligned}U^{[m]}\left(\boldsymbol{x}_i\right)\left(t_l\right) &= \left[\frac{\partial \rho^C_{BS}((y_i,\boldsymbol{x}_i),h)}{\partial h}\right]_{h=\hat{h}^{[m-1]}} \\ &= \left[\frac{\partial \frac{1}{2} w(t_l)\cdot\left(\mathbb{I}\left[y_i \leq t_l\right] - h(t_l \mid \boldsymbol{x}_i)\right)^2}{\partial h}\right]_{h=\hat{h}^{[m-1]}} \\ &= \left[w(t_l)\cdot\left(h\left(t_l \mid \boldsymbol{x}_i\right) - \mathbb{I}\left[y_i \leq t_l\right]\right)\right]_{h=\hat{h}^{[m-1]}} \\ &\forall i \in \{1,\ldots,N\}, l \in \{1,\ldots,L\}\end{aligned} \tag{25}$$

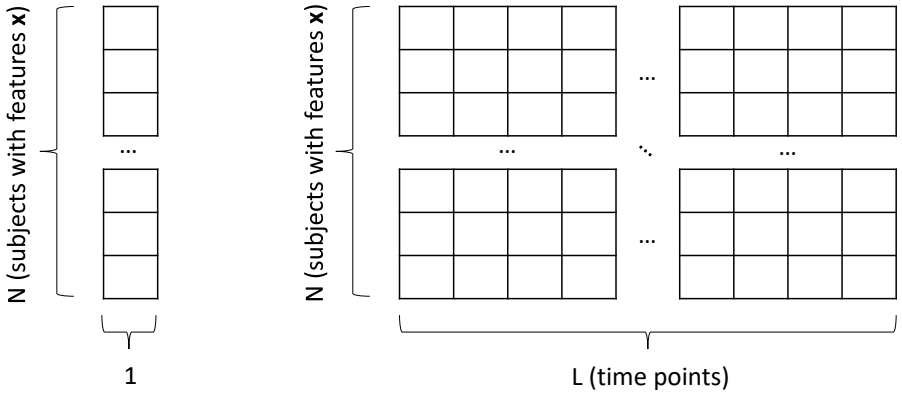

Figure 4: Comparison of evaluated observations in binary classification $h(\boldsymbol{x_i}) \forall i = 1, \ldots, N$ and individual survival distributions $h(t_l \mid \boldsymbol{x_i}) \forall i = 1, \ldots, N, l = 1, \ldots, L$. On the left, we depict the evaluated observations in the binary classification setting: We have a scalar probability prediction for every subject. On the right, we illustrate the observed grid in an individual survival distribution: We have a discrete probability distribution on L time points for every subject. Thus, we have to evaluate a two-dimensional grid (matrix) of observations in a survival setting.

with

$$w(t_l) = \begin{cases} 1/\hat{G}(t_l), & \text{if } t_l < \tilde{y} \\ 1/\hat{G}(\tilde{y}), & \text{if } t_l \geq \tilde{y} \text{ and } \delta = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Theoretical Survival Multicalibration Definition. Consequently, we can rewrite the multicalibration definition (17) for a survival probability distribution over time $[t_1, t_L]$ to

$$\left| \mathop{\mathbb{E}}_{i \sim S_v \cap S'} \underbrace{[w(t_l) \cdot (h(t_l \mid \boldsymbol{x}_i) - p_i^*)]}_{U(\boldsymbol{x}_i)(t_l)} \right| \leq \alpha \quad \forall \mathcal{S} \in \mathcal{C},\ l \in \{1, ..., L\}, \tag{26}$$

where we want a survival model h to be $(\mathcal{C}, \alpha)$-multicalibrated for any predicted value $v \in [0, 1]$, for every single time point t_l, and an $\alpha \in [0, 1]$. In accordance with the original formulation of multicalibration, this formulation is not information-theoretically possible, as it is the extension of the original definition for every time-point t_l.

Buckets in Time. As illustrated in Figure 4, we evaluate multicalibration on a grid of values in survival analysis. Analogous to the buckets for the subjects based on their

predicted value, we introduce time buckets. Instead of multicalibrating a model for every time point t_l, we perform it on time intervals. Thus, we can define the time buckets as denoted in (18) as $\Lambda_T[t_1, t_L]$ with a λ_T-discretization (19).

Empirical Survival Multicalibration. By including time buckets, the extension of the empirical multicalibration definition (20) is

$$\left| \left\langle c_{v,z}\left(\boldsymbol{x}\right), \lambda_T \cdot \frac{1}{L} \cdot \sum_{l \in \lambda_T(z)} w(t_l) \cdot \left(h\left(t_l \mid \boldsymbol{x}\right) - \mathbb{I}\left[y \leq t_l\right]\right) \right\rangle \right| \leq \alpha \quad \forall\, c \in \mathcal{C} \tag{27}$$

and for every $v \in \Lambda[0,1]$, $z \in \Lambda_T[t_1, t_L]$ and $c_{v,z}\left(\boldsymbol{x}\right)$ is a learner trained on the residuals for every value in the time frame $\Lambda_T[t_1, t_L]$ and prediction value in $\Lambda[0,1]$. For values outside the buckets $c_{v,z}\left(\boldsymbol{x}\right)$ is defined as 0. A survival model satisfying this condition is $(\mathcal{C}, \alpha, \lambda, \lambda_T)$-multicalibrated. Within every combination of time buckets λ_t and buckets over the predicted value λ we require the survival model h to be unbiased with respect to all efficiently-identifiable subgroups.

IBS-Multicalibration. If we set $\lambda_T = 1$, multicalibration for a single time bucket is defined as

$$\left| \left\langle c_v\left(\boldsymbol{x}\right), \underbrace{\frac{1}{L} \cdot \sum_{l=0}^{L} w(t_l) \cdot \left(h\left(t_l \mid \boldsymbol{x}\right) - \mathbb{I}\left[y \leq t_l\right]\right)}_{U(\boldsymbol{x})} \right\rangle \right| \leq \alpha \quad \forall\, c \in \mathcal{C}, v \in \Lambda[0,1], \tag{28}$$

where $c_v\left(\boldsymbol{x}\right) \in \mathcal{C}$ is a learner fitted on the residuals where $i \in \lambda(v)$ and 0 otherwise. This corresponds to deriving the loss function ℓ with respect to h evaluated at the current survival model $\hat{h}^{[m-1]}$ for the **censored Integrated Brier Score** (9):

$$\begin{aligned} U^{[m]}\left(\boldsymbol{x}\right) &= \left[\frac{\partial \ell_{BS}^{C}((y, \boldsymbol{x}), h)}{\partial h}\right]_{h=\hat{h}^{[m-1]}} \\ &= \left[\frac{\partial \left\{\frac{1}{2} \cdot \frac{1}{L} \cdot \sum_{l=0}^{L} w(t_l) \cdot \left(\mathbb{I}\left[y \leq t_l\right] - h(t_l \mid \boldsymbol{x})\right)^2\right\}}{\partial h}\right]_{h=\hat{h}^{[m-1]}} \\ &= \left[\frac{1}{L} \cdot \sum_{l=0}^{L} w(t_l) \cdot \left(h\left(t_l \mid \boldsymbol{x}\right) - \mathbb{I}\left[y \leq t_l\right]\right)\right]_{h=\hat{h}^{[m-1]}}. \end{aligned} \tag{29}$$

Depending on the number of buckets in time and predict values, we can now decide how exactly we calibrate survival models. The multicalibration definition ranges from calibration for every predicted value at every time point (26) in theory to the mean over time

and all predicted values (a combination of IBS-multicalibration (29) and multiaccuracy)

3.2. Achieving Multicalibration in Survival Analysis

The main changes in the multicalibration algorithm (Algorithm 4) result from the adapted multicalibration definition, directly affecting the stopping criterion, the gradient, and the bucketing strategy. However, the algorithmic structure does not change, as we still perform gradient boosting optimizing the Brier Score. Depending on the number of time buckets, we adapt the distribution differently. For the IBS-Multicalibration definition, we obtain one residual (the mean residual over time) per subject $U^{[m]}(\boldsymbol{x})$. In this case, the whole distribution is multiplied by one factor. If we have more than one bucket, the distribution changes within a time bucket $\lambda_T(z)$. This can result in a model h that does not meet the requirement that survival curves are monotonically decreasing. Then, we have to extend the idea of clipping probabilities to the survival curves.

Obtaining Survival Curves. By introducing bucketing in time, the survival model h can lose its inherent property: individual survival distributions are monotonically decreasing and have values in $[0, 1]$. The latter is solved under the original algorithm by clipping the value. To keep the survival prediction **monotonically decreasing**, we transfer the clipping process to a survival curve defined on a time frame from $[t_1, t_L]$ with L time points:

$$h^*(\boldsymbol{x})(t_l) = \begin{cases} h^*(\boldsymbol{x})(t_l), & \text{if } h^*(\boldsymbol{x})(t_{l-1}) \geq h^*(\boldsymbol{x})(t_l) \\ h^*(\boldsymbol{x})(t_{l-1}), & \text{otherwise.} \end{cases} \quad \forall\, l \in \{2, ..., L\}. \tag{30}$$

Alternative Time Buckets. As survival curves are monotonically decreasing, the original probability buckets in multicalibration Λ can be redefined. In practice, the survival probabilities within a smaller time frame are often similar across subjects, and therefore the idea is to create probability buckets depending on the time bucket. Otherwise, it could happen, for example, that within the first time steps all survival probabilities are above 0.9. As a result, some combinations of time and probability buckets are always empty. Probability buckets in general Λ can be reformulated to probabilities within the time bucket $\Lambda_P(z) = \Lambda[p_{z,min}, p_{z,max}]$, where $p_{z,min}$ is the minimal, and $p_{z,max}$ the maximal predicted probability within the time bucket $\lambda_T(z)$.

In this section, we provided a theoretical adaption of the multicalibration framework by adapting the multicalibration definition, the bucketing strategy, and bucketing to individual survival distributions. This lays the foundation for practical implementation.

Algorithm 4 Multicalibration in Survival Analysis.

1: Take the survival model: $\hat{h}^{[0]}(t \mid \boldsymbol{x})$.
2: Create buckets $v \in \Lambda[0,1], z \in \Lambda_T[t_1, t_L]$.
3: **for** $m = 0 \rightarrow M-1$ **do**
4: Calculate gradients: $U^{[m]}(\boldsymbol{x})(t) = -\left[\frac{\partial \rho((y,\boldsymbol{x}),h)}{\partial h}\right]_{h=\hat{h}^{[m-1]}}$
5: Fit a regression learner $c_{v,z}^{[m]}(\boldsymbol{x})$ to the gradients $U^{[m]}$ on every bucket $[\lambda(v), \lambda_t(z)]$.
6: Take the bucket with the largest correlation:
7: $v^*, z^* = \arg\max_{v,z} \left| \left\langle c_{v,z}^{[m]}(\boldsymbol{x}), \lambda_T \cdot \frac{1}{L} \cdot \sum_{l \in \lambda_T(z)} w(t_l) \cdot (h(t_l \mid \boldsymbol{x}) - \mathcal{I}[y \leq t_l]) \right\rangle \right|$
8: Check if $\hat{h}^{[m]}$ is already multicalibrated:
9: **if** $\left| \left\langle c_{v^*,z^*}^{[m]}(\boldsymbol{x}), \lambda_T \cdot \frac{1}{L} \cdot \sum_{l \in \lambda_T(z)} w(t_l) \cdot (h(t_l \mid \boldsymbol{x}) - \mathcal{I}[y \leq t_l]) \right\rangle \right| \leq \alpha$ **then**
10: **return** $\hat{h}^{[m]}$
11: **end if**
12: Multiplicatively update $\hat{h}^{[m+1]}(t \mid \boldsymbol{x}) = exp(-\eta^{[m]} \cdot c_{v^*,z^*}^{[m]}(\boldsymbol{x})) \cdot \hat{h}^{[m]}(t \mid \boldsymbol{x}) \quad \forall x \in \lambda(v^*), t \in \lambda_t(z^*)$
13: Project $\hat{h}^{[m+1]}(t \mid \boldsymbol{x})$ onto $[0,1]$ and obtain survival curve.
14: **end for**
15: Output $\hat{h}(t \mid \boldsymbol{x}) = \hat{h}^{[M]}(t \mid \boldsymbol{x})$

3.3. Implementation in R (McBoostSurv)

The R (R Development Core Team, 2020) implementation of McBoostSurv provides a survival extension to the existing R package mcboost (Pfisterer et al., 2021).

Methods. Figure 5 illustrates the three main methods of the R6 classes (Chang, 2021) McBoost and McBoostSurv:

1. initialize (new): The R6 object McBoostSurv[1] is initialized with different hyperparameters. They include the initial survival model that should be multicalibrated, the auditor algorithm (i.e., the base learner of the boosting algorithm), and other hyperparameters like the learning rate eta or the stopping criterion alpha.

2. multicalibrate: A initialized McBoostSurv object can be multicalibrated for every validation data set.

3. predict_probs: For new data, a multicalibrated McBoostSurv object can be utilized to predict survival probabilities.

[1] The implementation and adaptions can be found in the following Pull Request on GitHub: https://github.com/mlr-org/mcboost/pull/33

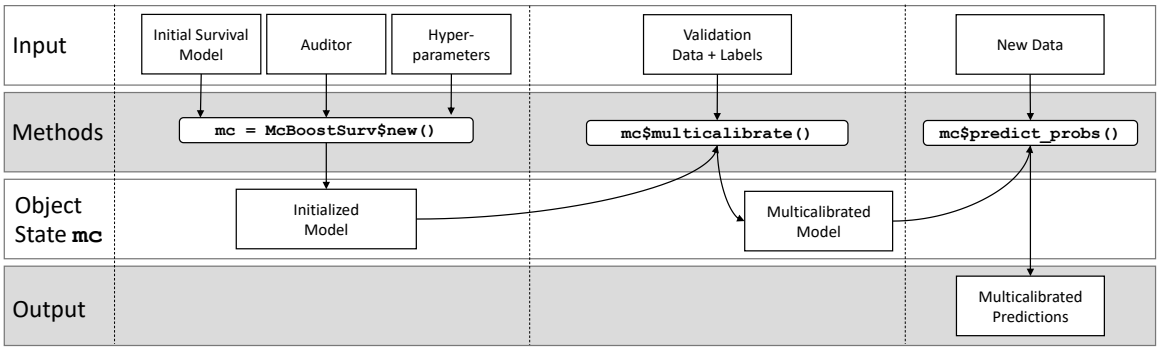

Figure 5: Methods in `McBoostSurv` based on Pfisterer et al. (2021). With `McBoostSurv$new()`, a new model can be initialized. Then, with the method `$multicalibrate()`, the object can be multicalibrated for a validation dataset. Finally, with `$predict_probs()`, the object can be used to predict multicalibrated survival probabilities.

Hyperparameters. In Table 1, we present the hyperparameters which are currently implemented in `McBoostSurv`. We introduced `time_points`, `time_buckets`, `bucket_aggregation`, `time_eval`, and `loss` in our survival extension.

Additionally, we implemented `PipeOpMCBoostSurv` as a `mlr3pipeline` (Binder et al., 2021) in order to integrate multicalibration in survival analysis into the `mlr3` universe (Lang et al., 2019) and thus connect our novel class to all existing functionalities within a typical machine learning workflow.

max_iter	Maximum number of boosting iterations (M).
alpha	Bound for accuracy of multicalibration (α).
eta	Learning rate for boosting algorithm (η).
num_buckets	Number of buckets in which the subjects are splitted based on their prediction value (λ).
bucket_strategy	Type of splitting between the buckets.
rebucket	Whether the buckets $\Lambda[0, 1]$ should be determined in every iteration.
eval_fulldata	If the auditor should be evaluated on the whole validation data or the data in the bucket.
partition	Whether there are buckets.
auditor_fitter	Base learner ($c(\boldsymbol{x})$)
subpops	Subgroups on which the learner is trained instead of a base learner.
init_predictor	Initial survival model ($\hat{h}^{[0]}(t \mid \boldsymbol{x})$)
default_model_class	Which standard model should be used if there is no initial survival model (e.g., Kaplan-Meier model)
multiplicative	Whether we multiply or add the base learner.
iter_sampling	Sampling strategy for the validation data.
time_points	Time points evaluated in the boosting algorithm ($[t_1, t_L]$).
time_buckets	Number of time buckets (λ_T^{-1})
bucket_aggregation	If time bucketing should be decided based on aggregation of the predicted values per subject (e.g., mean)
time_eval	Time quantile, which should be multicalibrated (similar to measuring 75%-IBS.
loss	Loss which is optimized during training (ρ).

Table 1: Implemented hyperparameters and their meaning in McBoostSurv.

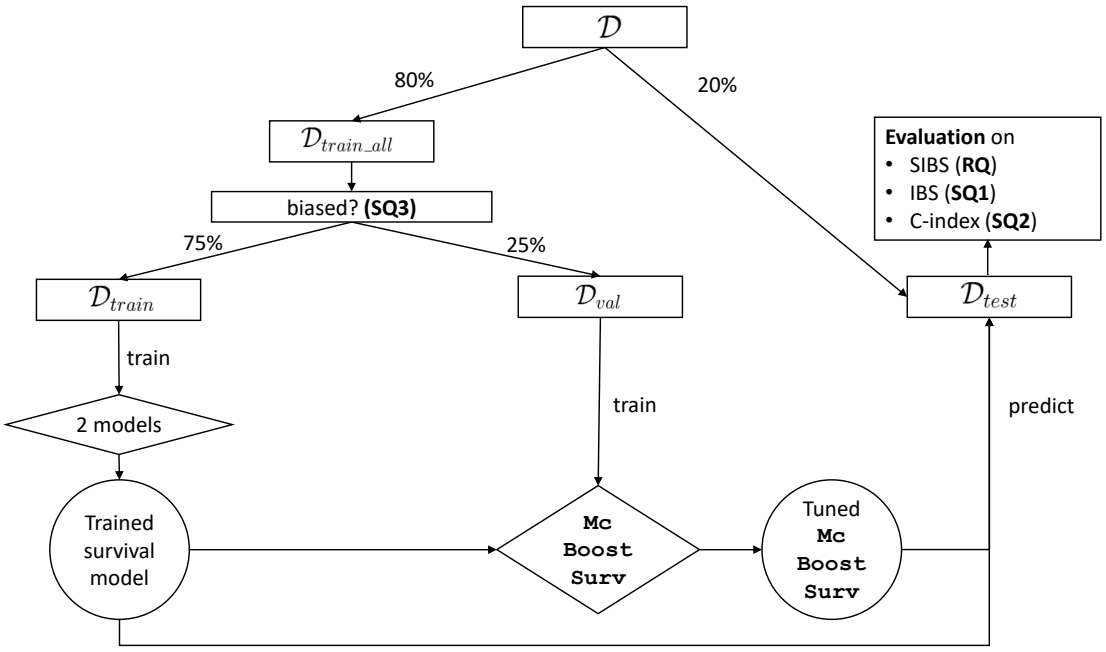

Figure 6: Benchmark setup for performance evaluation of `McBoostSurv` compared to initial survival model: repeated random sub-sampling performed on four data sets and split in test $\mathcal{D}_{test}$ and all data used for training $\mathcal{D}_{train\_all}$. $\mathcal{D}_{train\_all}$ is sampled biased or unbiased towards the majority subgroup between the training data $\mathcal{D}_{train}$ and validation data $\mathcal{D}_{val}$. We train two survival models with these configurations on $\mathcal{D}_{train}$. We post-process the trained survival model with validation data $\mathcal{D}_{val}$ on the best hyperparameter configuration. The predictions of both models on the test data $\mathcal{D}_{test}$ are evaluated with Subgroup-IBS, IBS, C-Index in every repetition of the repeated sub-sampling.

4. Experiments

We conducted several experiments to evaluate the suggested algorithm and `R` implementation `McBoostSurv` concerning the research question and the respective subquestions defined in the introduction (RQ: subgroup calibration, SQ1: calibration, SQ2: discrimination, SQ3: biased training data). The experiments compare two survival models and the respective multicalibrated models for four data sets and two different sub-sampling procedures for the desired metrics.

4.1. Experimental Design

We set up a benchmark as depicted in Figure 6. We evaluate the performance of two baseline survival models on four data sets and two sub-sampling techniques with repeated random sub-sampling. Initially, we randomly split the data $\mathcal{D}$ in $\mathcal{D}_{test}$ (20%) and all data

used for training $\mathcal{D}_{train\_all}$ (80%). The test data $\mathcal{D}_{test}$ is used to evaluate the performance of the baseline survival models and the respective multicalibrated models on the metrics defined in the research question (**RQ**) and subquestions 1 and 2 (**SQ1** and **SQ2**). The baseline survival models are trained on the training data $\mathcal{D}_{train}$, and the multicalibrated model is trained on the respective baseline survival model and the validation data $\mathcal{D}_{val}$. In addition, we perform hyperparameter tuning on the multicalibrated model concerning the Subgroup-IBS.

Baseline. As a baseline, we take two commonly used survival models, as their sole performance is not the focus of our analysis. A 10-fold cross-validated Cox proportional hazards model with elastic net penalty (`cv_glm`; Simon et al., 2011) and a survival random forest (`rf`; Breiman, 2001) implemented in the `R` package `ranger` (Wright & Ziegler, 2017). For the `cv_glm`, we estimate a survival distribution with accelerated failure time models (Cox & Oakes, 1984). For modeling the initial survival models, we use the `R` packages `mlr3learners` (Lang et al., 2021) and `mlr3proba` (Sonabend et al., 2021).

name	**N**	**p**	**%**	**sensitive attributes**			**%**
			censored	`raceBlack`	`sexF`	`age_65`	**minority group**
support	9,104	32	31.9	x	x		7.6
compas	10,310	6	73.2	x	x		9.4
kidtran	863	3	83.8	x	x		6.8
flchain	6,521	10	70.0		x	x	17.6

Table 2: Description of data sets used in experiments. N is the number of observations in the data set, and p denotes the number of features used for modeling. The percentage of censored data (*% censored*) shows the number of observations without completed status. *Sensitive attributes* indicate which overlapping subgroups we examine, and the proportion of the smallest subgroup is given (*% minority group*).

Data Sets. We conduct the experiments using publicly accessible survival analysis data sets from the real world. Additionally, we choose popular survival data sets (see Table 2), which are large enough for the described set-up ($N > 800$), are from different contexts (recidivism or healthcare), have a distinct number of features (p), and different proportion of censoring. Additionally, they include at least two sensitive attributes (i.e., gender, race, and age; Xiang & Raji, 2019) and varying size of the minority subgroup (based on two sensitive features). For the data sets support, compas, and kidtran dataset, we evaluate subgroups based on the defined binary attributes `raceBlack` (i.e., if the person is a person of color) and `sexF` (i.e., if the person is female). For the flchain

`alpha`	{**0.001**, 0.01, 0.05}
`auditor_fitter`	{"TreeAuditorFitter, **"RidgeAuditorFitter"**}
`eta`	{0.01, **0.1**}
`multiplicative`	{**TRUE**, FALSE}
`num_buckets`	{**1**, 2}
`time_buckets`	{1, **2**}

Table 3: Search space for hyperparameter tuning on `McBoostSurv`. We marked in bold our proposal for the default hyperparameter space.

data set, we use the binary-encoded variable `age_65` (i.e., if the person's age is > 65) as the second sensitive attribute instead of the race, as there is no information about the ethnic origin of the persons. To avoid fragmentation into very small subpopulations, we limit the number of sensitive attributes per subgroup to two. We include complete descriptions of the data sets and their pre-processing in Appendix B.

Biased Training Data. To address subquestion **SQ3**, we use two different sub-sampling methods to split the $\mathcal{D}_{train\_all}$ between the training $\mathcal{D}_{train}$ and validation data $\mathcal{D}_{val}$. In **unbiased** sampling, the data is sampled stratified by the sensitive attributes and status. In contrast, if the sampling is **biased towards the majority subgroup**, it is only stratified by the censoring indicator Δ. Additionally, the majority groups are sampled with a double probability in the train data $\mathcal{D}_{train}$ than in the validation data $\mathcal{D}_{val}$. The proportions of different subgroups in training $\mathcal{D}_{train}$ and validation data $\mathcal{D}_{val}$ can be found in Appendix B.

Hyperparameter Tuning on McBoostSurv. To use the optimal parameters in the evaluation, we tune the `McBoostSurv` learner on a discretized parameters space (see Table 3) that reflect a broad enough setting. We perform on the validation data $\mathcal{D}_{val}$ 3-fold cross validation with an exhaustive search on the whole parameter space. Then, we choose the best parameter set based on the Subgroup-IBS (31), as this is the evaluation measure in outer performance evaluation. With this parameter set, we train the `McBoostSurv` learner on the whole validation data set $\mathcal{D}_{val}$.

Evaluation. To answer the research question (**RQ**), we evaluate the mean over the censored version of the Integrated Brier Score with respect to all subpopulations (Subgroup-IBS, S-IBS):

$$\ell^C_{S-IBS}((\tilde{y} \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x})) = \frac{1}{n_{\mathcal{S}}} \cdot \sum_{j \in \mathcal{C}} \ell^C_{IBS,j}((\tilde{y} \leq t, \boldsymbol{x}), h(t \mid \boldsymbol{x})), \tag{31}$$

where $\mathcal{C} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{n_{\mathcal{S}}}\}$ and $n_{\mathcal{S}}$ is the number of subgroups which are evaluated

(i.e. four in our case). $\ell^C_{IBS,\mathcal{S}}$ is the IBS (9) evaluated on subgroup $\boldsymbol{x} \in \mathcal{S}$. To address **SQ1** and **SQ2**, we additionally evaluate the IBS (9) and C-Index (10) on the whole population.

Implementation. All experiments are implemented with the `batchtools R` package (Lang et al., 2017) and the `mlr3 R` machine learning framework (Lang et al., 2019).

4.2. Results

The mean results of the experiments are in Table 4 and the corresponding standard deviations can be found in Appendix B. In the following, we present the results of our experiments:

Subgroup Fairness (RQ). To answer the research question, we evaluated the mean IBS over the subgroups (31). The result shows (Table 4a) that in the mean, the subgroup fairness on the test data is considerably improved for the `cv_glm` and slightly decreased for `rf`. This observation is true for the unbiased training data. In general, the measures for the survival random forest are in the initial and multicalibrated models better than for the Cox proportional hazards model with an elastic net penalty. In a biased setting, for the kidtran data set, the performance of both models is slightly improved by multicalibrating them.

Calibration (SQ1). We addressed subquestion 1 by measuring the Integrated Brier Score on the whole population (Table 4b). The improvements are similar to the improvements in the Subgroup-IBS: an improvement of the IBS for all `cv_glm` and `rf` and slight depreciation for the survival random forest.

Discrimination (SQ2). The standard measure for discrimination in survival analysis is Harrell's C-index (Table 4c). The results clearly show a decrease in the measure in all models and data sets on the test data. Also, there is no improvement in the biased and unbiased setting except for the survival random forest trained on the support data set (line 8). This configuration improved in the mean in the biased and in the unbiased setting.

Biased Training Data (SQ3). The experiments regarding the last subquestion show that the effect is not very strong in our proposed setting. In general, the results did not change much in mean.

Additionally, we had the following additional result:

Default Hyperparameters for `McBoostSurv`. During hyperparameter tuning, we performed an extensive search on the defined discrete hyperparameter space. In Table 3, we marked our results: We clearly saw that the an ridge regression as auditor (`auditor_fitter`= "RidgeAuditorFitter"), not discretizing the probabilities (`num_buckets` = 1), a larger step size (`eta` = 0.1), a small alpha (`alpha` = 0.01 and `alpha` = 0.01) and a multiplicative update (`multiplicative` = TRUE) were the best parameters. We propose to tune the hyperparameter `time_buckets`, as this parameter has only a weak tendency to two buckets.

a. ↓ Subgroup Integrated Brier Score (RQ)						
			unbiased $\mathcal{D}_{train}$		biased $\mathcal{D}_{train}$	
#	data set	model	baseline	McBoostSurv	baseline	McBoostSurv
1	compas	cv_glm	0.2098	**0.1656**	0.2160	**0.1644**
2		rf	**0.1437**	0.1510	**0.1446**	0.1511
3	flchain	cv_glm	0.1933	**0.1412**	0.1923	**0.1521**
4		rf	**0.1007**	0.1068	**0.1060**	0.1082
5	kidtran	cv_glm	0.1614	**0.1540**	0.1587	**0.1546**
6		rf	**0.1469**	0.1630	0.1572	**0.1561**
7	support	cv_glm	0.2836	**0.2010**	0.2781	**0.2020**
8		rf	**0.1682**	0.1714	**0.1677**	0.1697

b. ↓ Integrated Brier Score (Calibration, SQ1)						
			unbiased $\mathcal{D}_{train}$		biased $\mathcal{D}_{train}$	
#	data set	model	baseline	McBoostSurv	baseline	McBoostSurv
1	compas	cv_glm	0.2223	**0.1763**	0.2285	**0.1748**
2		rf	**0.1538**	0.1603	**0.1541**	0.1606
3	flchain	cv_glm	0.1792	**0.1324**	0.1782	**0.1421**
4		rf	**0.0963**	0.1019	**0.1015**	0.1039
5	kidtran	cv_glm	0.1482	**0.1442**	**0.1447**	0.1453
6		rf	**0.1312**	0.1461	**0.1370**	0.1455
7	support	cv_glm	0.2808	**0.2006**	0.2752	**0.2020**
8		rf	**0.1672**	0.1706	**0.1671**	0.1700

c. ↑ C-Index (Discrimination, SQ2)						
			unbiased $\mathcal{D}_{train}$		biased $\mathcal{D}_{train}$	
#	data set	model	baseline	McBoostSurv	baseline	McBoostSurv
1	compas	cv_glm	**0.6703**	0.6436	**0.6687**	0.6505
2		rf	**0.6891**	0.6709	**0.6910**	0.6672
3	flchain	cv_glm	**0.7865**	0.7828	**0.7862**	0.7050
4		rf	**0.8037**	0.7762	**0.7834**	0.7745
5	kidtran	cv_glm	**0.6498**	0.6009	**0.6179**	0.6177
6		rf	**0.6587**	0.6255	**0.6501**	0.6272
7	support	cv_glm	**0.7556**	0.7134	**0.7553**	0.7124
8		rf	0.7256	**0.7317**	0.7255	**0.7345**

Table 4: Average results of the experiments for five repetitions. Each Table contains one evaluation measure: **Subgroup Integrated Brier Score** (Subgroup-IBS, **RQ**), **Integrated Brier Score** (IBS, **SQ1**), and the **C-index** (**SQ2**). In each table, each line is one of the two baseline survival models (`cv_glm`, `rf`) trained on the four data sets. For each combination, we compare the baseline survival model and the multicalibrated model (`McBoostSurv`), and if the training data $\mathcal{D}_{train}$ is unbiased or biased (sampled skewed towards the majority population, **SQ3**). The better value in each comparison is bold. The Integrated Brier Score should be minimized (↓), and the C-Index should be maximized (↑).

5. Discussion

This thesis proposed an extension of multicalibration, a framework for improving fairness regarding efficiently identifiable subgroups, to survival analysis. Here, we highlight the main findings from our experiments, and in the second subsection, we show possible limitations and opportunities for further research.

5.1. Main Findings

In our experiments, we computed the mean Integrated Brier Score for all subgroups, the IBS, and the C-index for the whole population. Therefore, we can answer the research questions formulated in the introduction.

(1) **Research question** (RQ): *Is a multicalibrated survival model fairer than the same survival model without post-processing?*
Our results are primarily in line with the results of the experiments conducted by Kim et al. (2019). Namely, a model with a poor calibration (`cv_glm`) in the subgroups (S-IBS) can be in mean improved by multicalibration. However, a model with a better IBS, the survival random forest (`rf`), could not be improved for the mean of the IBS evaluated for the subgroups. We suspect that the slight performance decrease happens due to overfitting, as we deal with a small validation data set $\mathcal{D}_{val}$. Nonetheless, the result also implies that retraining with a model with better calibration on a large data set that reflects the desired population can be sufficient. Thus, in a setting where we have only black-box access to an unfair survival model and a small validation set, multicalibration can enhance the performance with respect to the subgroups.

(2) **Sub-question 1** (SQ1): *How does multicalibration affect the calibration overall?*
Our results imply that in the cases where multicalibration can increase the S-IBS, it also increases the overall IBS. However, for the survival random forest, the IBS slightly decreases. Our results are also in line with Kim et al. (2019), who showed that calibration in a binary classification setting could improve the overall performance.

(3) **Sub-question 2** (SQ2): *How does multicalibration affect discrimination?*
We expected the overall model performance to decrease if a model is multicalibrated. This expectation did not hold with respect to IBS but for the discrimination of the model. Nevertheless, we can observe that the C-index in mean decreases for almost all combinations of data sets and baseline survival models by 0.02 in average.

(4) **Sub-question 3** (SQ3): *How does the effect of multicalibration change if the initial survival model is trained on a data set skewed towards a majority population?* Lastly, we expected to improve the subgroup measures from the baseline to the multicalibrated model if the training data $\mathcal{D}_{train}$ is more skewed towards the majority population than the validation data set $\mathcal{D}_{val}$. However, we could not observe a substantial change in the effect of the post-processing. The effect is merely apparent on the smallest data set kidtran: The subgroup fairness improved, and the overall IBS was higher (worse) for both baseline models. On the other hand, we cannot observe a substantial effect on the discrimination of the baseline and multicalibrated model. Possibly, effects might become more pronounced in situations of stronger sampling imbalance.

5.2. Limitations and Further Research

Our research takes a pioneering step towards post-processing black-box survival models for subgroup fairness via multicalibration. It is therefore subject to some limitations and should be interpreted accordingly. Throughout our work, we also identified several promising directions for further research.

Assumptions on the Data. Firstly, the generalization of these results is limited to the assumptions within our thesis and the multicalibration framework. These include the supposition that we only deal with right-censored survival data with time-constant features. Possibly, this can be extended by reformulating the survival problem and the multicalibration framework to a Poisson regression problem (Bender et al., 2020). Additionally, the mutlicalibration framework requires the validation data $\mathcal{D}_{val}$ to be unbiased. This strong assumption might not be fulfilled in many real-world settings (Chen et al., 2018).

Notion of Fairness. Secondly, this approach does not directly generalize to other notions of fairness and is limited to the definition of multicalibration and multiaccuracy. Different contexts or even stakeholders might require other definitions and measures of fairness (Binns, 2017). Kearns et al. (2018) propose another subgroup fairness notion (e.g., equalizing false-positive rates in the subgroups), extended to survival analysis. Another possibility is to evaluate the survival models beyond calibration - a model could output the marginal distribution and still be perfectly calibrated. A possible extension can be evaluating the sharpness of the model (e.g., Survival-CRPS; Avati et al., 2019).

Definition of Subgroups. Thirdly, the definition of subgroups in the algorithm is fixed to all computationally identifiable subgroups and in our analysis to two sensitive at-

tributes. Therefore, we believe there is much potential for further research on which and how many attributes subgroups are defined. The classical perception of "sensitive attributes" can be too tight (e.g., taste or opinion might also be an attribute that should be included in the analysis). Consequently, it might also be helpful to include other additional attributes in future data collection.

Boosting Algorithm. Lastly, the boosting algorithm in the multicalibration framework can be adapted. Currently, we only fit the auditor on the residuals per individual and time bucket. As a result, the same value shifts the curves in each time bucket, including several time points. Further research could use auditors modeling the residuals combined with the time (e.g., survival trees instead of decision trees; Bai et al., 2021). So far, we only consider boosting the first-order partial derivative of the gradient. Possibly, boosting higher orders may improve the results (Chen & Guestrin, 2016; Jung et al., 2020). Furthermore, in the multicalibration framework, we keep the probabilities within a range of $[0, 1]$ by clipping them. Alternatively, we could use the sigmoid function similar to the original gradient boosting approach (Friedman, 2001).

5.3. Conclusions

This thesis developed a post-processing algorithm for calibrating subgroups in a survival setting based on the multicalibration framework. So far, the fairness research has focused on group fairness for classification models. Hébert-Johnson et al. (2018) and Kim et al. (2019) presented multicalibration to overcome the idea of fixed groups and calibrate models for overlapping computationally identifiable subgroups. Our contributions are to extend this framework to distributional right-censored survival models:

(1) we presented the interpretation of multicalibration boosting as gradient boosting,

(2) developed a survival extension to multicalibration in theory,

(3) provided an implementation in `R`, and

(4) empirically evaluated the effects of multicalibration.

The main result of the experiments is that we can improve the calibration of the whole population and the subgroups if the model does not perform well. In our experiments, the discrimination decreased in the mean. That suggests that we might achieve considerable impact in situations where we only have black-box access to a model and a small validation data set. Multicalibration of a survival model can thus decrease the mismatch between the modeled world and the "world as it should be" and consequently reduce discrimination and bias against subgroups.

References

Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). *Machine Bias: There's software used across the country to predict future criminals. and it's biased against blacks.* (tech. rep.).

Avati, A., Duan, T., Zhou, S., Jung, K., Shah, N. H., & Ng, A. Y. (2019). Countdown regression: Sharp and calibrated survival predictions. *35th Conference on Uncertainty in Artificial Intelligence, UAI 2019*.

Bai, M., Zheng, Y., & Shen, Y. (2021). Gradient boosting survival tree with applications in credit scoring. *Journal of the Operational Research Society*.

Barda, N., Yona, G., Rothblum, G. N., Greenland, P., Leibowitz, M., Balicer, R., Bachmat, E., & Dagan, N. (2021). Addressing bias in prediction models by improving subpopulation calibration. *Journal of the American Medical Informatics Association*, *28*(3), 549–558.

Barda, N., Riesel, D., Akriv, A., Levy, J., Finkel, U., Yona, G., Greenfeld, D., Sheiba, S., Somer, J., Bachmat, E., Rothblum, G. N., Shalit, U., Netzer, D., Balicer, R., & Dagan, N. (2020). Developing a COVID-19 mortality risk prediction model when individual-level data are not available. *Nature Communications*, *11*(1), 1–9.

Barocas, S., Hardt, M., & Narayanan, A. (2020). *Fairness and Machine Learning*.

Bender, A., Rügamer, D., Scheipl, F., & Bischl, B. (2020). A general machine learning framework for survival analysis. *arXiv preprint arXiv:2006.15442*.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., & Roth, A. (2018). Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods and Research*, (1), 42.

Bhatnagar, S., Turgeon, M., Islam, J., Saarela, O., & Hanley, J. (2020). casebase: Fitting Flexible Smooth-in-Time Hazards and Risk Functions via Logistic and Multinomial Regression.

Binder, H., & Schumacher, M. (2008). Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. *BMC Bioinformatics*, *9*.

Binder, M., Pfisterer, F., Lang, M., Schneider, L., Kotthoff, L., & Bischl, B. (2021). mlr3pipelines-Flexible Machine Learning Pipelines in R. *Journal of Machine Learning Research*, *22*(184), 1–7.

Binns, R. (2017). Fairness in Machine Learning: Lessons from Political Philosophy. *Proceedings of Machine Learning Research*, *81*, 1–11.

Bolukbasi, T., Chang, K. W., Zou, J., Saligrama, V., & Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems*, *29*, 4356–4364.

Bonchi, F., Hajian, S., Mishra, B., & Ramazzotti, D. (2017). Exposing the probabilistic causal structure of discrimination. *International Journal of Data Science and Analytics*, *3*(1), 1–21.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Routledge.

Brier, G. W. (1950). Verification of Forecasts Expressed in Terms of Probability. *Monthly Weather Review*, *78*(1), 1–3.

Brockhaus, S., Melcher, M., Leisch, F., & Greven, S. (2017). Boosting flexible functional regression models with a high number of functional historical effects. *Statistics and Computing*, *27*(4), 913–926.

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, *22*(4), 477–505.

Buolamwini, J. (2018). Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. *Proceedings of Machine Learning Research*, *81*, 1–15.

Caton, S., & Haas, C. (2020). Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053*.

Chang, W. (2021). R6: Encapsulated Classes with Reference Semantics.

Chen, I. Y., Johansson, F. D., & Sontag, D. (2018). Why is my classifier discriminatory? *Advances in Neural Information Processing Systems*, *2018-Decem*, 3539–3550.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794.

Chiappa, S. (2019). Path-specific counterfactual fairness. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 7801–7808.

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

Chouldechova, A., & Roth, A. (2020). A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, *63*(5), 82–89.

Connors, A. F., Dawson, N. V., Desbiens, N. A., Fulkerson, W. J., Goldman, L., Knaus, W. A., Lynn, J., Oye, R. K., Bergner, M., Damiano, A., Hakim, R., Murphy, D. J., Teno, J., Virnig, B., Wagner, D. P., Wu, A. W., Yasui, Y., Robinson, D. K., & Kreling, B. (1995). A controlled trial to improve care for seriously ill hospitalized patients: The study to understand prognoses and preferences for outcomes and risks of treatments (SUPPORT). *JAMA - Journal of the American Medical Association*, *274*(20), 1591–1598.

Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*(2), 187–202.

Cox, D. R., & Oakes, D. (1984). *Analysis of Survival Data*. CRC Press.

D'Agostino, R. B., & Nam, B. H. (2003). Evaluation of the Performance of Survival Analysis Models: Discrimination and Calibration Measures. *Handbook of Statistics*, *23*, 1–25.

Dispenzieri, A., Katzmann, J. A., Kyle, R. A., Larson, D. R., Therneau, T. M., Colby, C. L., Clark, R. J., Mead, G. P., Kumar, S., Melton, L. J., & Rajkumar, S. V. (2012). Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. *Mayo Clinic Proceedings*, *87*(6), 517–523.

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

Ferryman, K., & Pitcan, M. (2018). Fairness in Precision Medicine. *Data & Society*, *February*(February), 58.

Fletcher, R. R., Nakeshimana, A., & Olubeko, O. (2021). Addressing Fairness, Bias, and Appropriate Use of Artificial Intelligence and Machine Learning in Global Health. *Frontiers in Artificial Intelligence*, *3*, 116.

Foulds, J. R., Islam, R., Keya, K. N., & Pan, S. (2020). An intersectional definition of fairness. *Proceedings - International Conference on Data Engineering*, *2020-April*, 1918–1921.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Gajane, P., & Pechenizkiy, M. (2017). On Formalizing Fairness in Prediction with Machine Learning. *arXiv preprint arXiv:1710.03184*.

Gerds, T. A., Kattan, M. W., Schumacher, M., & Yu, C. (2013). Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in Medicine*, *32*(13), 2173–2184.

Gerds, T. A., & Schumacher, M. (2006). Consistent estimation of the expected brier score in general survival models with right-censored event times. *Biometrical Journal*, *48*(6), 1029–1040.

Good, I. J. (1952). Rational Decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, *14*(1), 107–114.

Graf, E., Schmoor, C., Sauerbrei, W., & Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, *18*(17-18), 2529–2545.

Grote, T., & Berens, P. (2020). On the ethics of algorithmic decision-making in healthcare. *Journal of Medical Ethics*, *46*(3), 205–211.

Haider, H., Hoehn, B., Davis, S., & Greiner, R. (2020). Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, *21*, 1–63.

Hardt, M., Price, E., & Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 3323–3331.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., & Rosati, R. A. (1982). Evaluating the Yield of Medical Tests. *JAMA: The Journal of the American Medical Association*, *247*(18), 2543–2546.

Harrell, F. E., Lee, K. L., & Mark, D. B. (2005). Prognostic/Clinical Prediction Models: Multivariable Prognostic Models: Issues in Developing Models, Evaluating Assumptions and Adequacy, and Measuring and Reducing Errors. *Tutorials in Biostatistics, Statistical Methods in Clinical Studies*, *1*(4), 223–249.

Hébert-Johnson, Ú., Kim, M. P., Reingold, O., & Rothblum, G. N. (2018). Multicalibration: Calibration for the (computationally-identifiable) masses. *35th International Conference on Machine Learning, ICML 2018*, *5*, 3087–3103.

Hofner, B., Mayr, A., Robinzonov, N., & Schmid, M. (2014). Model-based boosting in R: A hands-on tutorial using the R package mboost. *Computational Statistics*, *29*(1-2), 3–35.

Hothorn, T., Kneib, T., & Bühlmann, P. (2014). Conditional transformation models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *76*(1), 3–27.

Iosifidis, V., & Ntoutsi, E. (2019). Adafair: Cumulative fairness adaptive boosting. *International Conference on Information and Knowledge Management, Proceedings*, 781–790.

Joseph, M., Kearns, M., Morgenstern, J., & Roth, A. (2016). Fairness in Learning: Classic and contextual bandits. *Advances in Neural Information Processing Systems*, *29*, 325–333.

Jung, C., Lee, C., Pai, M. M., Roth, A., & Vohra, R. (2020). Moment Multicalibration for Uncertainty Estimation.

Kalbfleisch, J. D., & Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data* (Wiley Inte).

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, *33*(1), 1–33.

Kamran, F., & Wiens, J. (2021). Estimating Calibrated Individualized Survival Curves with Deep Learning. *TheThirty-Fifth AAAI Conference on Artificial Intelligence*, *35*(1), 240–248.

Kaplan, E. L., & Meier, P. (1958). Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association*, *53*(282), 457–481.

Kearns, M., Neel, S., Roth, A., & Wu, Z. S. (2018). Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. *35th International Conference on Machine Learning, ICML 2018*, *6*, 4008–4016.

Keya, K. N., Islam, R., Pan, S., Stockwell, I., & Foulds, J. (2021). Equitable Allocation of Healthcare Resources with Fair Survival Models. *Proceedings of the 2021 siam international conference on data mining (sdm)* (pp. 190–198).

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking and Finance*, *34*(11), 2767–2787.

Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., & Schölkopf, B. (2017). Avoiding discrimination through causal reasoning. *Advances in Neural Information Processing Systems*, *2017-Decem*, 657–667.

Kim, M. P., Ghorbani, A., & Zou, J. (2019). Multiaccuracy: Black-box post-processing for fairness in classification. *AIES 2019 - Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 247–254.

Klein, J. P., & Moeschberger, M. L. (2003). *Survival analysis: techniques for censored and truncated data* (Vol. 1230). Springer.

Kleinberg, J., Mullainathan, S., & Raghavan, M. (2017). Inherent trade-offs in the fair determination of risk scores. *Leibniz International Proceedings in Informatics, LIPIcs*, *67*.

Kusner, M., Loftus, J., Russell, C., & Silva, R. (2017). Counterfactual fairness. *Advances in Neural Information Processing Systems*, *2017-Decem*, 4067–4077.

Kusner, M. J., Russell, C., Loftus, J. R., & Silva, R. (2019). Making decisions that reduce discriminatory impact. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 6368–6379.

Kvamme, H., Borgan, O., & Scheel, I. (2019). Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, *20*, 1–30.

Lang, M., Au, Q., Coors, S., & Schratz, P. (2021). mlr3learners: Recommended Learners for 'mlr3'.

Lang, M., Bischl, B., & Surmann, D. (2017). batchtools: Tools for R to work on batch systems. *The Journal of Open Source Software*, *2*(10).

Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L., & Bischl, B. (2019). mlr3: A modern object-oriented machine learning framework in R. *Journal of Open Source Software*, *4*(44), 1903.

Lee, C., Zame, W. R., Alaa, A. M., & van der Schaar, M. (2020). Temporal quilting for survival analysis. *AISTATS 2019 - 22nd International Conference on Artificial Intelligence and Statistics*, 596–605.

Lum, K., & Isaac, W. (2016). To predict and serve? *Significance*, *13*(5), 14–19.

Makhlouf, K., Zhioua, S., & Palamidessi, C. (2020). On the Applicability of ML Fairness Notions. *arXiv preprint arXiv:2006.16745*.

Mason, L., Baxter, J., Bartlett, P., & Frean, M. (2000). Boosting algorithms as gradient descent. *Advances in Neural Information Processing Systems*, *17*, 512–518.

Mayr, A., & Schmid, M. (2014). Boosting the concordance index for survival data - A unified framework to derive and evaluate biomarker combinations. *PLoS ONE*, *9*(1).

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). *A Survey on Bias and Fairness in Machine Learning* (tech. rep.).

Mitchell, S., Potash, E., Barocas, S., D'amour, A., & Lum, K. (2021). Annual Review of Statistics and Its Application Algorithmic Fairness: Choices, Assumptions, and Definitions. *Rev. Stat. Appl. 2021*, *8*, 141–163.

Mogensen, U. B., Ishwaran, H., & Gerds, T. A. (2012). Evaluating Random Forests for Survival Analysis Using Prediction Error Curves. *Journal of Statistical Software*, *50*(11), 1–23.

Mulligan, D. K., Kroll, J. A., Kohli, N., & Wong, R. Y. (2019). This thing called fairness: Disciplinary confusion realizing a value in technology.

Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, *12*(4), 595–600.

Noriega-Campero, A., Garcia-Bulle, B., Bakker, M. A., & Pentland, A. S. (2019). Active fairness in algorithmic decision making. *AIES 2019 - Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 77–83.

Penciana, M. J., & D'Agostino, R. B. (2004). Overall C as a measure of discrimination in survival analysis: Model specific population value and confidence interval estimation. *Statistics in Medicine*, *23*(13), 2109–2123.

Pfisterer, F., Kern, C., Dandl, S., Sun, M., Kim, M. P., & Bischl, B. (2021). mcboost: Multi-Calibration Boosting for R. *Journal of Open Source Software*, *6*(64), 3453.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., & Weinberger, K. Q. (2017). On fairness and calibration. *Advances in Neural Information Processing Systems*, *2017-Decem*, 5681–5690.

R Development Core Team. (2020). A Language and Environment for Statistical Computing.

Ridgeway, G. (1999). The State of Boosting. *Computing Science and Statistics*, *31*, 172–181.

Schmid, M., & Hothorn, T. (2008). Flexible boosting of accelerated failure time models. *BMC Bioinformatics*, *9*.

Schumann, C., Foster, J. S., Mattei, N., & Dickerson, J. P. (2020). We need Fairness and Explainability in Algorithmic Hiring. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, *2020-May*, 1716–1720.

Simoiu, C., Corbett-Davies, S., & Goel, S. (2017). The problem of infra-marginality in outcome tests for discrimination. *Annals of Applied Statistics*, *11*(3), 1193–1216.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2011). Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software*, *39*(5), 1–13.

Sonabend, R., Király, F. J., Bender, A., Bischl, B., & Lang, M. (2021). mlr3proba: an R package for machine learning in survival analysis. *Bioinformatics*.

Steyerberg, E. W., Borsboom, G. J., van Houwelingen, H. C., Eijkemans, M. J., & Habbema, J. D. F. (2004). Validation and updating of predictive logistic regression models: A study on sample size and shrinkage. *Statistics in Medicine*, *23*(16), 2567–2586.

Suresh, H., & Guttag, J. V. (2019). A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle. *arXiv preprint arXiv:1901.10002*.

Therneau, T. M., & Grambsch, P. M. (2000). *Modeling Survival Data: Extending the {C}ox Model*. Springer.

Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., & Wei, L. J. (2011). On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in Medicine*, *30*(10), 1105–1117.

van der Laan, M. J., & Robins, J. M. (2003). *Unified Methods for Censored Longitudinal Data and Causality*. Springer.

Vargo, A., Zhang, F., Yurochkin, M., & Sun, Y. (2021). Individually Fair Gradient Boosting. *arXiv preprint arXiv:2103.16785*.

Verma, S., & Rubin, J. (2018). Fairness definitions explained. *IEEE/ACM International Workshop on Software Fairness (FairWare)*, 1–7.

Wang, P., Li, Y., & Reddy, C. K. (2019). Machine learning for survival analysis: A survey. *ACM Computing Surveys*, *51*(6), 1–36.

Wang, Z., & Wang, C. Y. (2010). Buckley-James boosting for survival analysis with high-dimensional biomarker data. *Statistical Applications in Genetics and Molecular Biology*, *9*(1).

Wright, M. N., & Ziegler, A. (2017). Ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17.

Xiang, A., & Raji, I. D. (2019). On the Legal Compatibility of Fairness Definitions. *arXiv preprint arXiv:1912.00761*.

Yang, F., Cisse, M., & Koyejo, S. (2020). Fairness with overlapping groups. *Advances in Neural Information Processing Systems*, *2020-Decem*.

Zafar, M. B., Valera, I., Rodriguez, M. G., & Gummadi, K. P. (2017). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. *26th International World Wide Web Conference, WWW 2017*, 1171–1180.

Zhang, J., & Bareinboim, E. (2018). Fairness in decision-making the causal explanation formula. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2037–2045.

Zhang, T., & Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, *33*(4), 1538–1579.

List of Figures

List of Tables

A. Details on Boosting

A.1. Pseudo-residual of Log-Loss

$$\begin{aligned}
\tilde{r}(\boldsymbol{x}) = - & \left[\frac{\partial}{\partial f(\boldsymbol{x})} L(y, f(\boldsymbol{x}))\right]_{f=\hat{f}^{[m-1]}} \\
= - & \left[\frac{\partial}{\partial f(\boldsymbol{x})} - yf(\boldsymbol{x}) + \ln(1+\exp(f(\boldsymbol{x})))\right]_{f=\hat{f}^{[m-1]}} \\
= & \left[y - \frac{\exp(f(\boldsymbol{x}))}{1+\exp(f(\boldsymbol{x}))}\right]_{f=\hat{f}^{[m-1]}} \\
= & \left[y - \frac{1}{1+\exp(-f(\boldsymbol{x}))}\right]_{f=\hat{f}^{[m-1]}} \\
= & \left[y - s\left(f\left(\boldsymbol{x}\right)\right)\right]_{f=\hat{f}^{[m-1]}} \\
= & \left[y - \pi\left(\boldsymbol{x}\right)\right]_{f=\hat{f}^{[m-1]}}
\end{aligned}$$

A.2. Anyboost optimizes $L2$-loss internally

It can be shown that

$$\arg\max_{b\in\mathcal{B}} -\langle U\left(\mathbf{x}\right), b(\mathbf{x})\rangle = \arg\min_{b\in\mathcal{B}} \sum_{i=1}^{n} \left(b\left(\mathbf{x}_i\right) - \tilde{r}_i\right)^2,$$

if we assume that

1. $\sum_{i=1}^{n} \left(b\left(\mathbf{x}_i\right)\right)^2$ is constant, i.e. we normalize the predictions. By this assumption, we are only concerned with the direction of the base learner and not its length.

2. The negation of the base learner is still a base learner: $\forall b \in \mathcal{B} \rightarrow \exists - b \in \mathcal{B}$.

$$\begin{aligned}
& \arg\max_{b\in\mathcal{B}} -\langle U\left(\mathbf{x}\right), b(\mathbf{x})\rangle \\
= & \arg\min_{b\in\mathcal{B}} -\langle \tilde{r}, b(\mathbf{x})\rangle \\
= & \arg\min_{b\in\mathcal{B}} -\sum_{i=1}^{n} \tilde{r}_i b\left(\mathbf{x}_i\right) \\
= & \arg\min_{b\in\mathcal{B}} -2\sum_{i=1}^{n} \tilde{r}_i b\left(\mathbf{x}_i\right) \\
= & \arg\min_{b\in\mathcal{B}} \sum_{i=1}^{n} \underbrace{\tilde{r}_i^2}_{\text{constant}} - 2\tilde{r}_i b\left(\mathbf{x}_i\right) + \underbrace{\left(b\left(\mathbf{x}_i\right)\right)^2}_{\text{constant}} \\
= & \arg\min_{b\in\mathcal{B}} \sum_{i=1}^{n} \left(b\left(\mathbf{x}_i\right) - \tilde{r}_i\right)^2
\end{aligned}$$

B. Details on the Experimental Setup

B.1. Data Descriptions and Pre-Processing

Our experimental setup includes four data sets: support, compas, kidtran, and flchain. We performed first individual data pre-processing for every data set and second created one pre-processing pipeline for all data sets. For the individual data sets, we used standard pre-processing steps comparable to other benchmarks in the literature.

(1) The **support** (Study to understand prognoses and preferences for outcomes and risks of treatments, Connors et al., 1995) data is from major research to better understand prognoses, preferences, outcomes, and risks associated with therapy (SUPPORT), in which the survival time of critically sick hospitalized patients was examined. In our benchmark, we used the version of the `R` package `casebase` (Bhatnagar et al., 2020). For the support data set, we discretized the age and created a variable `age_group` that has the following ranges: "$0-14$", "$15-44$", "$45-64$", and ">64". We dropped all rows with missing values. Additionally, we added the two sensitive attributes `sexF` and `raceBlack`, that are not used during modeling.

(2) The **compas** dataset contains information on a system for predicting criminal recidivism that has been challenged for possible bias (Angwin et al., 2016). Angwin et al. (2016) used a Cox model to assess the compas system's effectiveness in predicting future recidivism for African-American defendants. They found that the system significantly overpredicts future recidivism for African-American defendants. We used the data set used in the analysis of Angwin et al. (2016)[2]. We followed their proposed procedure to obtain a survival data set: we calculated the time between start and end date and filtered all data points where there is a time > 0. Additionally, we selected the following variables for modeling: `sex`, `age`, `juv_fel_count`, `decile_score`, `priors_count`, `race`. We deleted all entries which are more than once in the data set. Like in the support data set, we created the two sensitive attributes `sexF` and `raceBlack`, that are not modelled.

(3) The **kidtran** dataset was derived from research examining the time interval between clinically evident infection and death in a group of individuals with renal insufficiency. We used the version in the `R` package `KMsurv` (Klein & Moeschberger, 2003). Here, we dropped all lines with missing values. Also, we added the two sensitive attributes `sexF` and `raceBlack`, that are not used in the modeling process.

[2] https://raw.githubusercontent.com/propublica/compas-analysis/master/cox-parsed.csv

(4) **Flchain** is a publicly available data set created by Dispenzieri et al. (2012) to examine the connection between the serum-free light chain and mortality. The source for the data set is the `R` package `survival` (Therneau & Grambsch, 2000). It considers variables such as age, sex, serum creatinine concentration, and the existence of monoclonal gammopathy. We eliminated all participants with missing variables, and excluded the variable `chapter` describing the death cause. In addition, we created the two sensitive attributes `sexF` and `age_65` that are not utilized throughout the modeling process.

For all data sets, we used a pre-processing pipeline of `mlr3pipelines` (Binder et al., 2021) that conducts the following steps

(1) We imputed all numeric features with their median.

(2) Then, removed all constant features.

(3) Next, we encode all variables which are not numeric with one-hot-encoding.

(4) Finally, we discretized all survival times in the data set in 200 quantiles to reduce the size of the matrix of time and subjects, which is the basis for modeling and evaluation.

B.2. Biased Training Data

		$\mathcal{D}_{train}$ split (in %)				$\mathcal{D}_{val}$ split (in %)			
dataset	**biased?**	**MN**	**MB**	**FN**	**FB**	**MN**	**MB**	**FN**	**FB**
compas	×	39.07	40.45	11.05	9.43	39.05	40.41	11.09	9.45
	✓	41.56	43.04	8.41	6.99	31.58	32.63	19.02	16.77
kidtran	×	50.10	10.49	32.62	6.80	50.00	10.92	32.18	6.90
	✓	56.48	11.18	27.04	5.30	30.81	8.84	48.95	11.40
support	×	48.59	7.69	36.14	7.58	48.55	7.73	36.15	7.57
	✓	55.29	8.73	29.77	6.21	28.43	4.63	55.25	11.69
dataset	**biased?**	**MY**	**MO**	**FY**	**FO**	**MY**	**MO**	**FY**	**FO**
flchain	×	27.33	17.62	28.15	26.90	27.36	17.62	28.12	26.90
	✓	22.25	14.85	32.04	30.86	42.61	25.94	16.46	15.00

Table 5: Splits of the four subgroups are defined on the two sensitive attributes in a biased and unbiased setting. In the first column, the four data sets are listed. The second column marks a bias for the majority groups (✓) or not (×). The subsequent values (in %) are the proportions of the subgroups in the training data set $\mathcal{D}_{train}$ and the test data set $\mathcal{D}_{val}$. The subgroups are denoted by the following abbreviations: M = male, F = female, N = not black, B = black, O = old, Y = young. The proportions in the test data $\mathcal{D}_{test}$ can be derived from the unbiased setting, as the sensitive attributes stratify the data.

B.3. Standard Deviation of the Results

a. Standard Deviation of the S-IBS (RQ)

			unbiased $\mathcal{D}_{train}$		biased $\mathcal{D}_{train}$	
#	data set	model	baseline	McBoostSurv	baseline	McBoostSurv
1	compas	cv_glm	0.0226	0.0085	0.0318	0.0108
2		rf	0.0104	0.0029	0.0088	0.0023
3	flchain	cv_glm	0.0029	0.0023	0.0042	0.0168
4		rf	0.0108	0.0019	0.0026	0.0023
5	kidtran	cv_glm	0.0070	0.0085	0.0056	0.0093
6		rf	0.0238	0.0137	0.0236	0.0109
7	support	cv_glm	0.0049	0.0008	0.0032	0.0014
8		rf	0.0230	0.0053	0.0236	0.0017

b. Standard Deviation of the IBS (Calibration, SQ1)

			unbiased $\mathcal{D}_{train}$		biased $\mathcal{D}_{train}$	
#	data set	model	baseline	McBoostSurv	baseline	McBoostSurv
1	compas	cv_glm	0.0198	0.0088	0.0309	0.0111
2		rf	0.0101	0.0032	0.0094	0.0014
3	flchain	cv_glm	0.0027	0.0019	0.0039	0.0146
4		rf	0.0102	0.0027	0.0030	0.0034
5	kidtran	cv_glm	0.0059	0.0056	0.0027	0.0143
6		rf	0.0181	0.0111	0.0129	0.0104
7	support	cv_glm	0.0049	0.0013	0.0024	0.0018
8		rf	0.0236	0.0046	0.0237	0.0024

c. Standard Deviation of the C-Index (Discrimination, SQ2)

			unbiased $\mathcal{D}_{train}$		biased $\mathcal{D}_{train}$	
#	data set	model	baseline	McBoostSurv	baseline	McBoostSurv
1	compas	cv_glm	0.0077	0.0587	0.0084	0.0547
2		rf	0.0356	0.0128	0.0336	0.0079
3	flchain	cv_glm	0.0075	0.0053	0.0077	0.0733
4		rf	0.0374	0.0091	0.0060	0.0053
5	kidtran	cv_glm	0.0860	0.0980	0.1105	0.0833
6		rf	0.0698	0.0378	0.0446	0.0605
7	support	cv_glm	0.0051	0.0118	0.0048	0.0172
8		rf	0.0347	0.0100	0.0356	0.0091

Table 6: Standard deviation of the measures for the results in Table 4.

Declaration of authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.

Munich, 18.09.2021

.....

Carolin Becker