

# Fairness without Imputation: A Decision Tree Approach for Fair Prediction with Missing Values

Haewon Jeong, Hao Wang, Flavio P. Calmon

Harvard University

{haewon,flavio}@seas.harvard.edu, hao\_wang@g.harvard.edu

## Abstract

We investigate the fairness concerns of training a machine learning model using data with missing values. Even though there are a number of fairness intervention methods in the literature, most of them require a complete training set as input. In practice, data can have missing values, and data missing patterns can depend on group attributes (e.g. gender or race). Simply applying off-the-shelf fair learning algorithms to an imputed dataset may lead to an unfair model. In this paper, we first theoretically analyze different sources of discrimination risks when training with an imputed dataset. Then, we propose an integrated approach based on decision trees that does not require a separate process of imputation and learning. Instead, we train a tree with missing incorporated as attribute (MIA), which does not require explicit imputation, and we optimize a fairness-regularized objective function. We demonstrate that our approach outperforms existing fairness intervention methods applied to an imputed dataset, through several experiments on real-world datasets.

## 1 Introduction

Datasets can contain missing values, i.e., unobserved variables that would be meaningful for analysis if observed (Little and Rubin 2019). In many domains ranging from survey data, electronic health records, and recommender systems, data missingness is so common that it is the norm rather than the exception. This challenge has inspired significant research on methods to handle missing values (Little and Rubin 2019; Schafer and Graham 2002; Buuren and Groothuis-Oudshoorn 2010; Royston 2004; Molenberghs and Kenward 2007). Data missingness is usually modeled as a random process independent of non-missing features. However, when it comes to human-related data, data missingness often correlates with the subject’s sociodemographic group attributes<sup>1</sup>. For example, in medical data, there can be more missing values in low-income patients as they are more likely to refuse to take costly medical tests. In survey data, questionnaires can be less friendly for a certain population, e.g., the fonts are too small for elderly participants, or the language is too difficult for non-native English speakers. We

show a concrete example of different missing patterns in different sociodemographic groups in the high school longitudinal study (HSLS) dataset (Ingels et al. 2011), which contains education-related surveys from students and parents<sup>2</sup>. In the parent survey, there are generally 6-7% more missing values in under-represented minority (URM) students<sup>3</sup> compared to White and Asian students. On the questions related to secondary caregivers, URM students had 15% more missing values, as there are disproportionately more children in a single-parent family in URM (Bureau 2019).

In the machine learning (ML) pipeline, missing values are handled in the preprocessing step (either dropping missing entries or performing imputation) before training a model. When there exist disparate missing patterns across different groups, how missing values are treated can affect the final trained prediction model, not just its accuracy, but also its fairness. Despite the ubiquity of the problem, fairness issues in missing values have gained only limited attention (Fernando et al. 2021; Wang and Singh 2021). The burgeoning literature on fair ML algorithms either overlooks the existence of missing data or assumes that the training data are always complete (see e.g., Calmon et al. 2017; Hardt, Price, and Srebro 2016; Zemel et al. 2013; Zafar et al. 2019; Feldman et al. 2015; Menon and Williamson 2018).

In this work, we connect two well-studied, yet still mostly disjoint research areas—handling missing values and fair ML—and answer a crucial question:

*In order to train a fair model on data with missing values, is it sufficient to first impute the data then apply a fair learning algorithm?*

We address this question through both a theoretical analysis and an experimental study with real-world datasets.

In the first part of the paper, we examine the limitation of the disconnected process of imputing first and training next. We identify three different potential sources of discrimination. First, we show that depending on the missing patterns, the performance of imputation methods can be different per group. As a result, the predictive model trained on imputed data can inherit and propagate biases that exists in the imputed data. Then, we show that even when we

<sup>2</sup>See Section C of the supplementary material for details

<sup>3</sup>In this paper we use the term URM to include Black, Hispanic, Native American and Pacific Islander.

use an imputation method that has an unbiased performance at the training time, if different imputation is employed at the testing time, this can give rise to discriminatory performance of the trained model. Finally, we prove a fundamental information-theoretic result: *there is no universally fair imputation method for different downstream learning tasks*.

To overcome the above-mentioned limitations, we propose an integrated approach, called *Fair MIP Forest*, which learns a fair model without the need for explicit imputation. The Fair MIP Forest algorithm is a decision tree based approach that combines two different ideas: missing incorporated as attribute (MIA) (Twala, Jones, and Hand 2008) for handling missing values and mixed integer programming (MIP) (Bertsimas and Dunn 2017) formulation to optimize a fairness-regularized objective function. By marrying the two ideas, we are able to optimize fairness and accuracy in an end-to-end fashion, instead of finding an optimal solution for the imputation process and the training process separately. Finally, we propose using an ensemble of fair trees instead of training a single tree. This reduces overall training time and improves accuracy. We implement the Fair MIP Forest algorithm and test it on three real-world datasets, including the aforementioned HSLS dataset. The experimental results show that our approach performs favorably compared to existing fair learning algorithms trained on an imputed dataset in terms of fairness-accuracy trade-off.

## 1.1 Related Works

Methods that handle missing values have a long history in statistics (Little and Rubin 2019; Schafer and Graham 2002; Buuren and Groothuis-Oudshoorn 2010; Royston 2004; Molenberghs and Kenward 2007; Stekhoven and Bühlmann 2012; Tang and Ishwaran 2017). The simplest way of handling missing values is dropping rows with missing entries (also known as complete case analysis). However, in the missing values literature, it is strongly advised to use all the available data as even with a small missing rate (e.g. 2-3%), dropping can lead to suboptimal performance and unethical selection bias (Newman 2014). A more desirable way to deal with missing values is imputation (Little and Rubin 2019), where missing entries are replaced with a new value. This includes inserting dummy values, mean imputation, or regression imputation (e.g. k-nearest neighbor (k-NN) regression) (Donders et al. 2006; Zhang 2016; Bertsimas, Pawlowski, and Zhuo 2017). Multiple imputation is a popular class of imputation methods (Rubin 2004; Wulff and Jeppesen 2017; White, Royston, and Wood 2011) that draws a set of possible values to fill in missing values, as opposed to single imputation that substitutes a missing entry with a single value. While our theoretical analysis focuses on single imputation for its simplicity, our proposed Fair MIP Forest algorithm shares conceptual similarities with multiple imputation since we train multiple trees with different random mini batches, each of which treats missing values differently.

Alternatively, some ML models do not require an explicit imputation step. Decision trees, for instance, have several ways to handle missing values directly such as surrogate splits (Breiman et al. 2017), block propagation (Ke et al. 2017), and missing incorporated as attribute (MIA) (Twala,

Jones, and Hand 2008). In the second part of the paper, we focus on decision trees with MIA as it is empirically shown to outperform other missing values methods in decision trees (Kapelner and Bleich 2015; Josse et al. 2019). For this reason, the MIA approach is implemented in the widely-used XGBoost algorithm (Chen and Guestrin 2016). The main challenge of XGBoost is that it requires a twice-differentiable loss function but most group fairness metrics are non-differentiable. Due to this challenge, there is only one workshop paper on training a fair XGBoost model (Ravichandran et al. 2020) that is limited to minimizing the statistical parity metric using a surrogate loss function. In contrast, our framework is applicable to a broad class of group discrimination metrics and minimizes them directly in the mixed-integer program.

We study the problem of learning from incomplete data and its fairness implications. In this regard, our work is related with Kallus, Mao, and Zhou (2021); Fogliato, G’Sell, and Chouldechova (2020); Mehrotra and Celis (2021) which consider the case where the sensitive attributes or labels are missing or noisy (e.g. due to imputation). In contrast, we focus on the case where the input features are missing and may thus impact performance in downstream prediction tasks. The works that are most relevant to our work are Fernando et al. (2021); Wang and Singh (2021) as they examine the intersection of general data missingness and fairness. While Fernando et al. (2021) presents a comprehensive investigation on the relationship between fairness and missing values, their analyses are limited to observational and empirical studies on how different ways of handling missing values can affect fairness. Wang and Singh (2021) proposes reweighting scheme that assigns lower weight to data points with missing values by extending the preprocessing scheme given in Calmon et al. (2017). The question we address in this work is fundamentally different from these two works. We examine if we can rectify fairness issues that arise from missing values by applying existing fair learning algorithms after imputation. Unlike the previous works that are limited to empirical evaluations, we demonstrate the potential fairness risk of learning from imputed data from a theoretical perspective. Our theoretical analysis is inspired by a line of works (see e.g., Chen, Johansson, and Sontag 2018; Feldman et al. 2015; Zhao and Gordon 2019) which aim to quantify and explain algorithmic discrimination. Furthermore, our solution to learning a fair model from missing values is an in-processing approach that explicitly minimizes widely-adopted group fairness measures such as equalized odds or accuracy parity, in contrast to the preprocessing approach in Wang and Singh (2021).

Our proposed Fair MIP Forest algorithm is inspired by Aghaei, Azizi, and Vayanos (2019) that introduced training a fair decision tree using mixed integer linear programming. Framing decision tree training as an integer program was first suggested in Bertsimas and Dunn (2017), and adapted and improved in Verwer and Zhang (2017, 2019). Unlike conventional decision tree algorithms where splitting at each node is determined in a greedy manner, this formulation produces a tree that minimizes a specified objective function. We follow the optimization formulation given in Bert-

simas and Dunn (2017), but we add MIA in the optimization to decide the optimal way to send missing entries at each branching node. To the best of our knowledge, our work is the first one to include missing value handling in the integer optimization framework. Furthermore, unlike Aghaei, Azizi, and Vayanos (2019) which only had statistical parity as a fairness regularizer, we implement four different fairness metrics (FPR/FNR/accuracy difference and equalized odds) in the integer program. Finally, we go one step further from training a single tree, and propose using an ensemble of weakly-optimized trees. Raff, Sylvester, and Mills (2018) also studies a fair ensemble of decision trees. However, their approach follows the conventional greedy approach and does not consider missing values.

## 2 Framework

**Supervised learning and disparate impact.** Consider a supervised learning task where the goal is to predict an outcome random variable  $Y \in \mathcal{Y}$  using an input random feature vector  $X \in \mathcal{X}$ . For a given loss function  $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ , the performance of an ML model  $h : \mathcal{X} \rightarrow \mathcal{Y}$  can be measured by a population risk  $L(h) \triangleq \mathbb{E}[\ell(h(X), Y)]$ . Some typical choices of the loss function are 0-1 loss  $\ell_{0-1}(\hat{y}, y) = \mathbb{I}[\hat{y} = y]$  for classification task or mean squared loss  $\ell_2(\hat{y}, y) = (\hat{y} - y)^2$  for regression. Since the underlying distribution  $P_{X,Y}$  is unknown, one usually minimizes an empirical risk instead using a dataset of samples drawn from  $X$  and  $Y$  given by  $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ :

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \ell(h(\mathbf{x}_i), y_i), \quad (1)$$

where  $\mathcal{H}$  is a class of predictive models (e.g., neural networks or logistic regression).

A model  $h$  exhibits disparate impact (Barocas and Selbst 2016) if its performance varies across population groups, defined by a group attribute  $S$ . For simplicity, we assume that  $S$  is binary, but our framework can be extended to a more general intersectional setting. For a given loss function  $\ell$ , we measure the performance of  $h$  on group  $s$  by  $L_s(h) \triangleq \mathbb{E}[\ell(h(X), Y) | S = s]$  and say that  $h$  is discriminatory if  $L_0(h) \neq L_1(h)$ . We define the *discrimination risk* as:

$$\text{Disc}(h) \triangleq |L_0(h) - L_1(h)|. \quad (2)$$

We refer the readers to Dwork et al. (2018); Donini et al. (2018) for a discussion on how to recover some commonly used group fairness measures, such as equalized odds (Hardt, Price, and Srebro 2016), by selecting different loss functions in (2). In this paper, we say a model ensures fairness when (2) is small or zero for a chosen loss function.

**Data missingness.** In practice, data may contain features with missing values. We assume that the complete data are composed of two components: observed variables  $X_{\text{obs}}$  and a missing variables  $X_{\text{ms}}$ . For the purpose of our theoretical analysis, we assume that only single variable is missing—a common simplifying assumption made in the statistics literature (Little and Rubin 2019). We introduce a binary variable  $M$  for indicating whether  $X_{\text{ms}}$  is missing (i.e.,  $M = 1$

if and only if  $X_{\text{ms}}$  is missing). Finally, we define the incomplete feature vector by  $\tilde{X} = (X_{\text{obs}}, \tilde{X}_{\text{ms}}) \in \tilde{\mathcal{X}}$  where

$$\tilde{X}_{\text{ms}} = \begin{cases} X_{\text{ms}} & \text{if } M = 0 \\ * & \text{otherwise.} \end{cases}$$

Note that we drop the assumption that only one variable is missing when developing the Fair MIP Forest in Section 4.

Missing data can be categorized into three types (Little and Rubin 2019) based on the relationship between missing pattern and observed variables:

- Missing completely at random (MCAR) if  $M$  is independent of  $X$ ;
- Missing at random (MAR) if  $M$  depends only on the observed variables  $X_{\text{obs}}$ ;
- Missing not at random (MNAR) if neither MCAR nor MAR holds.

Even though most missing patterns in the real world are MNAR, the theoretical studies on imputation methods often rely on the MCAR (or MAR) assumption. However, when the missing pattern varies across groups, even if the each group satisfies MCAR/MAR, it is possible that entire population does not, and vice versa. This is formalized in the following lemma.

**Lemma 1.** *There exist data distributions such that (i) each group satisfies MCAR (or MAR) but the entire population does not or (ii) the entire population satisfies MCAR (or MAR) but each group does not.*

**Data imputation.** Using incomplete feature vectors for solving the empirical risk minimization in (1) may be challenging since most optimizers, such as gradient-based methods, only take real-valued inputs. To circumvent this issue, one can impute missing values by applying a mapping  $f_{\text{imp}} : \tilde{\mathcal{X}} \rightarrow \mathcal{X}$  on the incomplete feature vector  $\tilde{X}$ .

## 3 Risks of Training with Imputed Data

A model trained on imputed data may manifest disparate impacts for various reasons. For example, the imputation method can be inaccurate for the minority group, leading to a biased imputed dataset. As a result, models trained on the imputed dataset may inherit this bias. In this section, we identify three different ways imputation can fail to produce fair models: (i) the imputation method may have different performance per group, (ii) there can be a mismatch between the imputation methods used during training and testing time, and (iii) the imputation method may be applied without the knowledge of the downstream predictive task. For the last point, we prove that there is no universal imputation method that ensures fairness across all downstream learning tasks.

### 3.1 Biased Imputation Method

To quantify the performance of an imputation method and how it varies across different groups, we introduce the following definition.

**Definition 1.** The performance of an imputation method  $f_{\text{imp}}$  on group  $s$  is measured by

$$L_s(f_{\text{imp}}) \triangleq \mathbb{E} \left[ \|f_{\text{imp}}(\tilde{X}) - X\|_2^2 \mid M = 1, S = s \right]. \quad (3)$$

Furthermore, we define the *discrimination risk* of  $f_{\text{imp}}$  by

$$\text{Disc}(f_{\text{imp}}) \triangleq |L_0(f_{\text{imp}}) - L_1(f_{\text{imp}})|. \quad (4)$$

Next, we show that even when data satisfy the simplest assumption of MCAR within each group, the optimal imputation  $f_{\text{imp}}^*$  can still exhibit discrimination risk.

**Theorem 1.** Assume that data from each group are MCAR. For the sake of illustration, we let  $X_{\text{obs}} = \emptyset$  and the optimal imputation method be

$$f_{\text{imp}}^* = \underset{f_{\text{imp}}}{\operatorname{argmin}} \mathbb{E} \left[ (f_{\text{imp}}(\tilde{X}) - X)^2 \mid M = 1 \right].$$

We can decompose its discrimination risk as

$$\begin{aligned} \text{Disc}(f_{\text{imp}}^*) &= |(p_1^{ms} - p_0^{ms})(m_1 - m_0)^2 \\ &\quad + (\text{Var}[X|S=0] - \text{Var}[X|S=1])| \end{aligned}$$

where  $p_s^{ms} \triangleq \Pr(S=s|M=1)$  and  $m_s \triangleq \mathbb{E}[X|S=s]$  for  $s \in \{0, 1\}$ .

This theorem reveals three factors which may cause discrimination in data imputation: (i) different proportion of missing data,  $p_1^{ms} - p_0^{ms}$ , (ii) difference in group means,  $m_1 - m_0$ , and (iii) different variance per group,  $\text{Var}[X|S=0] - \text{Var}[X|S=1]$ . The first factor could be caused by different missing pattern across groups. Since  $X_{\text{obs}} = \emptyset$ , an optimal imputation method would be imputing a constant, and the optimal constant is the population mean for the L2 loss.<sup>4</sup> Hence, the second factor measures the gap between the per-group optimal imputation methods. Finally, the variance quantifies the performance of the mean imputation on each group. Therefore, the last factor indicates how suitable the mean imputation is for each group.

**Remark 1.** We briefly discuss how to improve the fairness of imputation based on the factors identified in Theorem 1. First, if the bias mainly comes from the difference  $p_1^{ms} - p_0^{ms}$ , then one can collect more samples from the minority group which has more missing data or resample the training dataset. Second, if the mean values of each group are significantly different (i.e.,  $m_1 - m_0$  is large), then one can impute the missing values for each group separately, assuming it is legal and ethical to do so (Dwork et al. 2018; Wang et al. 2021). Lastly, if the variance difference  $\text{Var}[X|S=0] - \text{Var}[X|S=1]$  is large, then one may consider using a different imputation mechanism for each group instead of mean imputation.

### 3.2 Mismatched Imputation Methods

Imputation is an unavoidable process to train many classes of ML models (e.g. logistic regression) in the presence of missing values. However, a user of a trained model might

not have information on what type of imputation was performed for training. People who disseminate a trained model might omit the details about the imputation process. In some cases, it is not possible for a model developer to disclose all the details about the imputation due to privacy concerns. Take mean imputation as an example. Releasing the sample mean can leak private information in the training set, and a user might have to estimate the sample mean from the testing data, which would not precisely match that of training data. In the following theorem, we provide how this imputation mismatch between training and testing can aggravate the discrimination risk.

**Theorem 2.** For a predictive model  $h$  and an imputation method  $f_{\text{imp}}$ , the performance on group  $s$  is measured by

$$L_s(h \circ f_{\text{imp}}) \triangleq \mathbb{E} \left[ \ell(h \circ f_{\text{imp}}(\tilde{X}), Y) \mid S = s \right]. \quad (5)$$

Assume that the loss function  $\ell$  is bounded by a constant  $K > 0$  and data from group  $s \in \{0, 1\}$  are MCAR with probability  $p_s$ . Let  $f_{\text{imp}}^{\text{train}}$  and  $f_{\text{imp}}^{\text{test}}$  be the imputation methods used in the training and testing time, respectively. Then

$$\begin{aligned} &|L_0(h \circ f_{\text{imp}}^{\text{test}}) - L_1(h \circ f_{\text{imp}}^{\text{test}})| \\ &\leq |L_0(h \circ f_{\text{imp}}^{\text{train}}) - L_1(h \circ f_{\text{imp}}^{\text{train}})| \\ &\quad + K \sum_s p_s D_{\text{TV}}(P_s^{\text{train}} \| P_s^{\text{test}}). \end{aligned} \quad (6)$$

where  $D_{\text{TV}}(\cdot \| \cdot)$  is the total variation distance and  $P_s^{\text{train}}$ ,  $P_s^{\text{test}}$  are the probability distributions of  $(f_{\text{imp}}^{\text{train}}(\tilde{X}), Y) \mid M = 1, S = s$  and  $(f_{\text{imp}}^{\text{test}}(\tilde{X}), Y) \mid M = 1, S = s$ , respectively. Finally, there exist a data distribution, a predictive model, and an imputation method such that the equality in (6) is achieved.

The second term in the upper bound in (6) shows that even if discrimination risk is completely eliminated at training time, a mismatched imputation method can still give rise to an overall discrimination risk at testing time.

### 3.3 Imputation Without Being Aware of the Downstream Tasks

Imputing missing values and training predictive model are closely intertwined. On the one hand, the performance of the predictive model relies on how missing data are imputed. On the other hand, if data are imputed blindly without taking the downstream tasks into account, the predictive model produced from the imputed data can be unfair. We ask a fundamental question on whether there exists a universally good imputation method that can guarantee fairness and accuracy regardless of which model class is used in the downstream. We address this question by borrowing notions from learning theory.

Many existing fairness intervention algorithms (see e.g., Donini et al. 2018; Zafar et al. 2019; Celis et al. 2019; Wei, Ramamurthy, and Calmon 2020; Alghamdi et al. 2020) can be understood as solving the following optimization problem (or its various approximations) for a given  $\epsilon \geq 0$

$$\begin{aligned} &\min_{h \in \mathcal{H}} L(h) \\ &\text{s.t. } |L_0(h) - L_1(h)| \leq \epsilon. \end{aligned} \quad (7)$$

<sup>4</sup>See Section A.2 in the supplementary material.

Since the imputation method changes the data distribution, it affects the solution of the optimization problem in (7) as well. The following definition characterizes the class of imputation methods that ensure the existence of an accurate solution of (7).

**Definition 2.** For a given hypothesis class  $\mathcal{H}$ , a distribution of  $(S, \tilde{X}, Y)$ , and constants  $\epsilon, \delta \geq 0$ , we call an imputation method  $(\epsilon, \delta)$ -conformal if the minimal value of (7) under the imputed data distribution is upper bounded by  $\delta$ .

Although the above definition does not say anything about how an optimal solution can be reached, a conformal imputation guarantees the existence of a fair solution. On the other hand, if non-conformal method is applied to impute data, solving (7) will always return an unfair (or inaccurate) solution, no matter which optimizer is used.

By definition, whether an imputation method is conformal relies on the hypothesis class. Hence, one may wonder if there is a universally conformal imputation method that can be applied to missing values regardless of the hypothesis class. However, the following theorem states that such a method, unfortunately, does not exist.

**Theorem 3.** *There is no universally conformal imputation method. Specifically, for  $\epsilon, \delta < 0.5$ , there exists a data distribution of  $(S, \tilde{X}, Y)$  and two hypothesis classes  $\mathcal{H}_1$  and  $\mathcal{H}_2$  such that their  $(\epsilon, \delta)$ -conformal imputation methods are disjoint.*

The previous theorem suggests that imputing missing values and training a predictive model cannot be treated separately. Otherwise, it may result in a dilemma where there is no fair model computed from the imputed data. On the other hand, it is often hard to find a conformal imputation method that is tailored to a particular hypothesis class. In fact, even verifying that a given imputation method is conformal is non-trivial without solving the optimization problem in (7).

## 4 Fair Decision Tree with Missing Values

The previous section reveals several ways discrimination risks can be introduced in the process of training with imputed data. Motivated by the limitations of decoupling imputation from model training, we propose an integrated approach for learning a fair model with missing values. The proposed approach is based on decision trees. We exploit the “missing incorporated in attribute” (MIA) (Twala, Jones, and Hand 2008), which uses data missingness to compute the splitting criteria of a tree without performing explicit imputation. We combine MIA with the integer programming approach for training a fair decision tree. The details of the algorithm is described next.

### 4.1 Fair MIP Forest Algorithm

In this section, we propose *Fair MIP Forest* algorithm for binary classification where missing value handling is embedded in a training process that minimizes both classification error and discrimination risk. The algorithm is designed by marrying two different ideas: MIA and mixed integer programming (MIP) formulations for fitting fair deci-

sion trees. In order to mitigate the high computational cost and the overfitting risk of the integer programming method, we propose using an ensemble of under-optimized trees. We begin with providing a brief background on MIA and the MIP formulation for decision trees.

**Missing Incorporated in Attribute (MIA).** MIA is a method that naturally handles missing values in decision trees by using missingness itself as a splitting criterion. It treats missing values as a separate category,  $\{\ast\}$ , and at each branching node, it sends all missing values to the left or to the right. I.e., there are two ways we can split at a branch that uses the  $j$ -th feature:

- $\{X_j \leq q \text{ or } X_j = \ast\} \text{ vs } \{X_j > q\}$ ,
- $\{X_j \leq q\} \text{ vs } \{X_j > q \text{ or } X_j = \ast\}$ .

Note that by setting  $q = -\infty$  in the first case, we can make the split:  $\{X_j = \ast\} \text{ vs } \{X_j \neq \ast\}$ .

**Mixed Integer Programming (MIP) for Decision Trees.** To set up the MIP formulation, let us introduce a few notations (see Figure 1 for an illustration on a depth-2 decision tree). We consider a decision tree  $\mathcal{T}$  of fixed depth  $D$ , and for simplicity we assume a full tree. This can be described with an ordered set of branching nodes  $\mathcal{V}$  and an ordered set of leaf nodes  $\mathcal{L}$ . Note that  $|\mathcal{V}| = 2^D - 1$  and  $|\mathcal{L}| = 2^D$ . Learning a decision tree with missing values corresponds to learning four variables  $\mathcal{T} \triangleq (\mathbf{P}, \mathbf{q}, \mathbf{c}, \mathbf{u})$ : the feature we split on ( $\mathbf{P}$ ), splitting threshold ( $\mathbf{q}$ ), and whether to send missing values to the left or to the right ( $\mathbf{c}$ ) at each branching node in  $\mathcal{V}$ , and the prediction we make at each leaf node ( $\mathbf{u}$ ). More details on each of these variables are given below.

Recall that  $n$  is the number of samples in the training set  $\mathcal{D}$  and  $d$  is the number of features. At each branching node  $v \in \mathcal{V}$ , we split on one feature specified by the one-hot-encoded vector  $\mathbf{p}_v \in \{0, 1\}^d$ , and we let  $\mathbf{P} \in \{0, 1\}^{|\mathcal{V}| \times d}$  be the matrix where each row is  $\mathbf{p}_v$ . Since  $\mathbf{p}_v$  is a one-hot encoded vector,  $\sum_j p_{v,j} = 1$  for all  $v \in \mathcal{V}$ . When the value is not missing, we split at the threshold  $q_v$ , i.e., if the feature selected at the node  $v$  is  $j$ , a data point that has  $x_{i,j} \leq q_v$  will go to the left and a data point with  $x_{i,j} > q_v$  will go to the right. When  $x_{i,j}$  is missing, then it will go to the left if  $c_v = 1$  and to the right branch if  $c_v = 0$ . Finally, we assume that we give the same prediction to all data points in the same leaf node, i.e., for a leaf node  $l \in \mathcal{L}$ , the prediction will be  $u_l \in \mathcal{Y}$ .

How a tree  $\mathcal{T}$  makes a prediction on an input data point,  $(\mathbf{x}_i, y_i) \in \mathcal{D}$ , can be described with two more variables:  $\mathbf{w}_i \in \{0, 1\}^{|\mathcal{V}|}$  and  $\mathbf{z}_i \in \{0, 1\}^{|\mathcal{L}|}$  ( $i \in [n]$ ).  $\mathbf{w}_i$  represents where the data point goes to at each node, i.e.,  $w_{i,v} = 1$  means that the data point goes to the left branch at the branching node  $v$ .  $\mathbf{z}_i$  is an one-hot-encoding vector that represents the final destination leaf node of the data point, i.e.,  $z_{i,l} = 1$  means that the data point goes to the leaf node  $l \in \mathcal{L}$ , and we assign  $\hat{y}_i = u_l$ .

Under this setting, we minimize the following fairness-regularized objective function:

$$\ell(\mathcal{D}) + \lambda \cdot \ell_{\text{fair}}(\mathcal{D}). \quad (8)$$

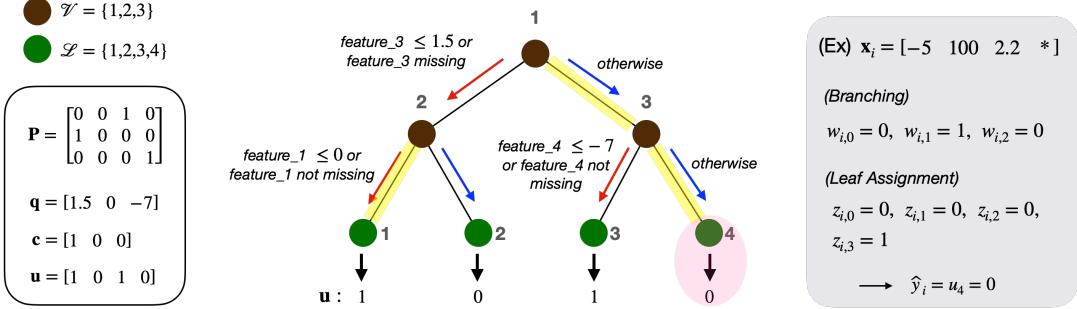


Figure 1: Demonstration of MIP notations for decision trees with MIA using a depth-2 tree example. The data dimension  $d = 4$ , and  $\mathbf{P}$  is a 3-by-4 matrix where each row dictates which feature a branching node uses for split. In the first row of  $\mathbf{P}$ , the third element is one, so feature\_3 is used for splitting at branch node 1. The first element in  $\mathbf{q}$  and  $\mathbf{c}$  are used as a splitting threshold missing values splitting at branch node 1. We also show how prediction is made for the example data point  $\mathbf{x}_i = [-5 \ 100 \ 2.2 \ *]$ . The branching decisions  $w_{i,v}$ 's are computed for all  $v$  (highlighted in yellow). From the computed  $w_{i,v}$ 's, the algorithm decides the leaf node it belongs to, i.e.,  $\mathbf{z}_i$ . In this case  $\mathbf{z}_i = [0 \ 0 \ 0 \ 1]$ . Hence,  $\hat{y}_i = u_4 = 0$ .

We use 0-1 loss for  $\ell(\mathcal{D})$  and implement four different  $\ell_{\text{fair}}$ : accuracy difference, FNR difference, FPR difference, and equalized odds. Section B in the supplementary material describe in detail how we encode these into a mixed integer linear program.

**Fair MIP Forest.** We now describe our Fair MIP Forest algorithm. We take an ensemble approach to train multiple trees from different mini-batches, and employ early termination. Solving an integer program is a computationally intensive and time-consuming process. However, we observed that in the early few iterations, we can find a reasonably well-performing model, but it takes a long time to get to the (local) optimum, making minute improvements over iterations. Based on this observation, we terminate the optimization for training each tree early after some time limit ( $t_{\text{limit}}$ ), and we use multiple weakly-optimized trees. We observed that this not only reduces training time but also improves prediction accuracy. Termination time ( $t_{\text{limit}}$ ) and the number of trees ( $n_{\text{tree}}$ ) are hyperparameters that can be tuned. Furthermore, we initialize each tree with the tree obtained in the previous iteration, to reduce the time required in the initial search. After we obtain  $\{\mathcal{T}^{(i)}\}_{i=1,\dots,n_{\text{tree}}}$ , for prediction on a new data point, we perform prediction on each tree and make a final decision using the majority rule.

## 5 Experimental Results

We present the implementation of the Fair MIP Forest algorithm. Through experiments on three datasets, we compare our proposed algorithm and existing fair learning algorithms coupled with different imputation methods.

**Datasets.** We test our Fair MIP Forest algorithm on three datasets: COMPAS (Angwin et al. 2016), Adult (Dua and Graff 2017), and high school longitudinal study (HSLS) dataset (Ingels et al. 2011). For Adult and COMPAS datasets, we generate artificial missing values as the original datasets do not contain any missing values. For missing value generation, we chose a set of features to erase and for a given feature, we erased randomly with different missing

probabilities per group, which are chosen so that the baseline results (i.e., no fairness intervention) have enough disparities either in FPR or FNR (see Section C in the supplementary material). For the HSLS dataset, there was no artificial missing value generation as the original dataset already contains a substantial amount of missing values. The sensitive attribute in Adult dataset was gender (0: Female, 1: Male), in COMPAS dataset was race (0: Black, 1: White), and in HSLS was also race (0:URM, 1:White/Asian).

**Setup.** We implemented the the Fair MIP Forest algorithm with Gurobi 9.1.2 for solving the integer program. All experiments were run on a cluster with 32 CPU cores and 32GB memory. For each dataset, we tune hyperparameters for the Fair MIP Forest algorithm: tree depth, number of trees, batch size, time limit for each tree training (see Section C in the supplementary material for details). We vary  $\lambda$  in (8) to obtain different points on the fairness-accuracy curve. We compare Fair MIP Forest with the baseline, which performs simple imputation (mean or k-NN) and then trains a decision tree. We compare our method with three fair learning methods coupled with imputation: the exponentiated gradient algorithm (Agarwal et al. 2018), disparate mistreatment algorithm (Zafar et al. 2019), and equalized odds algorithm (Hardt, Price, and Srebro 2016). We refer to these by the first author's name. All experiments are run 10 times with different random train-test splits to measure the variance in both accuracy and fairness metrics. For COMPAS and HSLS datasets, we choose FNR difference as a fairness metric because the difference in FPR was already very small in the baseline classifier without any fairness intervention. Similarly, we only regularize FPR difference in Adult dataset as FNR difference in the baseline was negligible. For Agarwal and Zafar, we vary hyperparameters to get different fairness-accuracy trade-off. Hardt does not have tunable hyperparameters.

**Discussion.** The experimental results are summarized in Figure 2. With the COMPAS dataset, we observe that the Fair MIP Forest algorithm achieves a better accuracy for

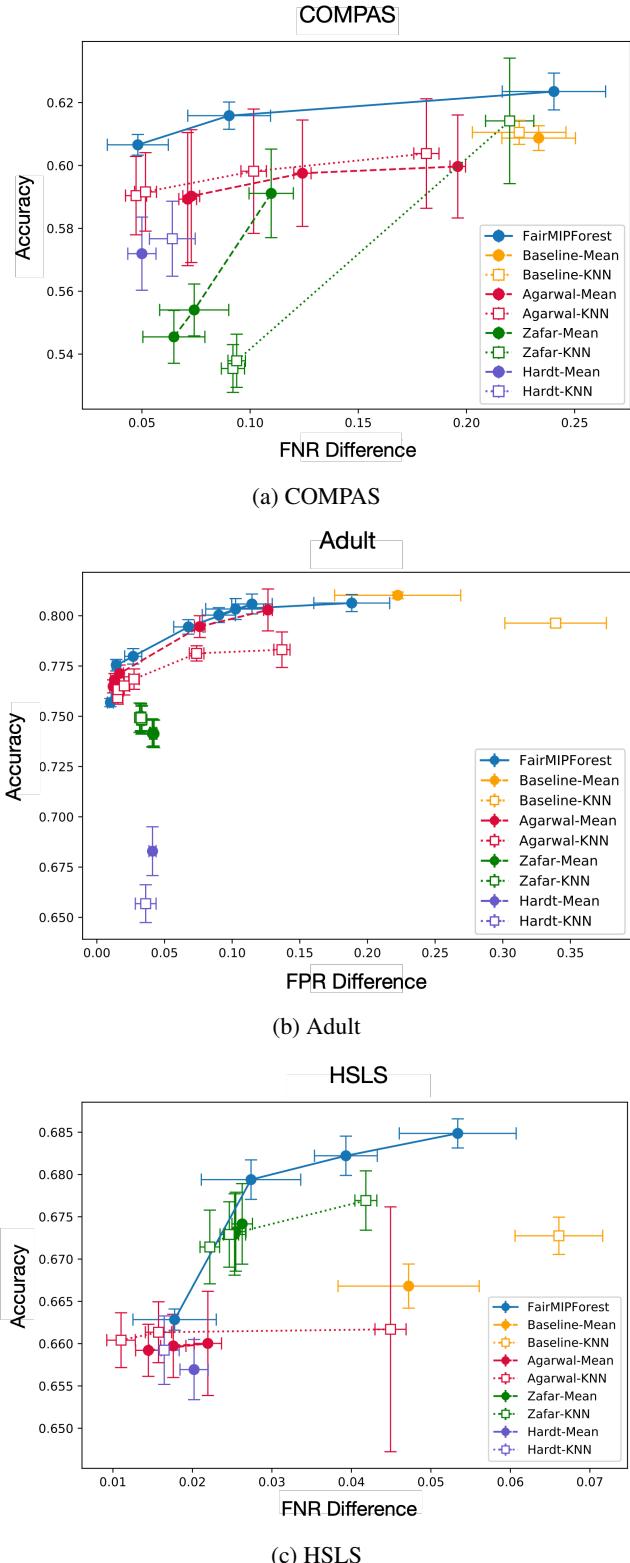


Figure 2: Comparison of Fair MIP Forest with existing fairness intervention approaches (Zafar et al. 2019; Hardt, Price, and Srebro 2016; Agarwal et al. 2018) coupled with mean or k-NN imputations. Baseline indicates the result of training a decision tree without any fair intervention. Error bars show the standard error after 10 runs with different train-test splits.

the same FNR difference, compared to all the other existing works we tested. In the Adult dataset, Fair MIP Forest and Agarwal coupled with mean imputation showed the best performance. In the HSLS experiments, Fair MIP Forest showed superior accuracy for higher FNR difference values although Agarwal was able to achieve smaller FNR difference (close to 0.01).

We also observe that even with the same dataset, there is no one imputation method that has better performance in all fair learning algorithms. For example, in Adult, while Agarwal clearly performs better with mean imputation, Zafar has slightly better fairness and accuracy with k-NN imputation. Also, notice that the performance of a fair learning algorithm depends heavily on which imputation method was used. In the COMPAS dataset, Zafar with mean imputation performs significantly better than Zafar with k-NN imputation. In the Adult dataset, Agarwal with mean imputation had better performance than Agarwal with k-NN. This suggests that how to pair which imputation with which fair learning method for a given dataset is not a straightforward question. With the Fair MIP Forest algorithm, we can sidestep such question and simply apply it to any dataset with missing values.

For all experiments, we set  $t_{\text{limit}}$  to 60 seconds, which means that training 30 trees takes roughly 1,800 seconds. Compared to what is reported in Aghaei, Azizi, and Vayanos (2019) – 15,000+ seconds for training a tree for COMPAS and Adult datasets – this is more than 8x time saving.

## 6 Discussion and Future Work

In this work, we analyze different sources of fairness risks, in terms of commonly-used group fairness metrics, when we train a model from imputed data. Extension of our analysis to multiple imputation (e.g., MICE (Van Buuren and Groothuis-Oudshoorn 2011)) and to other fairness metrics (e.g., individual fairness (Dwork et al. 2012), preference-based fairness (Zafar et al. 2017), or rationality (Ustun, Liu, and Parkes 2019)) would be an interesting future direction. We then introduce our solution to training a fair model with missing values, that utilizes decision trees. While tree-based algorithms are a preferred choice in many settings for their interpretability and ability to accommodate mixed data types (categorical and real-valued), we hope our work can inspire the development of fair handling of missing values in other supervised models, such as neural networks. Finally, the general question of how to design a fair imputation procedure is a widely open research problem.

## 7 Acknowledgement

This material is based upon work supported by the National Science Foundation under grants CAREER 1845852, IIS 1926925, and FAI 2040880.

## References

- Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; and Wallach, H. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*, 60–69. PMLR.
- Aghaei, S.; Azizi, M. J.; and Vayanos, P. 2019. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1418–1426.
- Alghamdi, W.; Assoodeh, S.; Wang, H.; Calmon, F. P.; Wei, D.; and Ramamurthy, K. N. 2020. Model projection: Theory and applications to fair machine learning. In *2020 IEEE International Symposium on Information Theory (ISIT)*, 2711–2716. IEEE.
- Angwin, J.; Larson, J.; Mattu, S.; Kirchner, L.; and ProPublica. 2016.
- Barocas, S.; and Selbst, A. D. 2016. Big data’s disparate impact. *Calif. L. Rev.*, 104: 671.
- Bellamy, R. K.; Dey, K.; Hind, M.; Hoffman, S. C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; Mojsilovic, A.; et al. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*.
- Bertsimas, D.; and Dunn, J. 2017. Optimal classification trees. *Machine Learning*, 106(7): 1039–1082.
- Bertsimas, D.; Pawlowski, C.; and Zhuo, Y. D. 2017. From predictive methods to missing data imputation: an optimization approach. *The Journal of Machine Learning Research*, 18(1): 7133–7171.
- Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 2017. *Classification and regression trees*. Routledge.
- Bureau, U. S. C. 2019. 2019 The American Community Survey 1-year Public Use Microdata Samples. Technical report, Washington D.C.
- Buuren, S. v.; and Groothuis-Oudshoorn, K. 2010. mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, 1–68.
- Calmon, F. P.; Wei, D.; Vinzamuri, B.; Ramamurthy, K. N.; and Varshney, K. R. 2017. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 3995–4004.
- Celis, L. E.; Huang, L.; Keswani, V.; and Vishnoi, N. K. 2019. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, 319–328.
- Chen, I.; Johansson, F. D.; and Sontag, D. 2018. Why is my classifier discriminatory? In *Advances in Neural Information Processing Systems*, 3539–3550.
- Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794.
- Donders, A. R. T.; Van Der Heijden, G. J.; Stijnen, T.; and Moons, K. G. 2006. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10): 1087–1091.
- Donini, M.; Oneto, L.; Ben-David, S.; Shawe-Taylor, J.; and Pontil, M. 2018. Empirical risk minimization under fairness constraints. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2796–2806.
- Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.
- Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.
- Dwork, C.; Immorlica, N.; Kalai, A. T.; and Leiserson, M. 2018. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, 119–133.
- Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 259–268.
- Fernando, M.-P.; Cèsar, F.; David, N.; and José, H.-O. 2021. Missing the missing values: The ugly duckling of fairness in machine learning. *International Journal of Intelligent Systems*.
- Fogliato, R.; G’Sell, M.; and Chouldechova, A. 2020. Fairness Evaluation in Presence of Biased Noisy Labels. *arXiv preprint arXiv:2003.13808*.
- Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*.
- Ingels, S. J.; Pratt, D. J.; Herget, D. R.; Burns, L. J.; Dever, J. A.; Ottem, R.; Rogers, J. E.; Jin, Y.; and Leinwand, S. 2011. High School Longitudinal Study of 2009 (HSLS: 09): Base-Year Data File Documentation. NCES 2011-328. *National Center for Education Statistics*.
- Josse, J.; Prost, N.; Scornet, E.; and Varoquaux, G. 2019. On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931*.
- Kallus, N.; Mao, X.; and Zhou, A. 2021. Assessing algorithmic fairness with unobserved protected class using data combination. *Management Science*.
- Kapelner, A.; and Bleich, J. 2015. Prediction with missing data via Bayesian additive regression trees. *Canadian Journal of Statistics*, 43(2): 224–239.
- Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30: 3146–3154.
- Little, R. J.; and Rubin, D. B. 2019. *Statistical analysis with missing data*, volume 793. John Wiley & Sons.
- Mehrotra, A.; and Celis, L. E. 2021. Mitigating Bias in Set Selection with Noisy Protected Attributes. In *Proceedings of*

- the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 237–248.
- Menon, A. K.; and Williamson, R. C. 2018. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, 107–118. PMLR.
- Molenberghs, G.; and Kenward, M. 2007. *Missing data in clinical studies*, volume 61. John Wiley & Sons.
- Newman, D. A. 2014. Missing data: Five practical guidelines. *Organizational Research Methods*, 17(4): 372–411.
- Polyanskiy, Y.; and Wu, Y. 2019. Lecture notes on information theory. *Lecture Notes for 6.441 (MIT)*, ECE 563 (UIUC), STAT 364 (Yale).
- Raff, E.; Sylvester, J.; and Mills, S. 2018. Fair forests: Regularized tree induction to minimize model bias. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 243–250.
- Ravichandran, S.; Khurana, D.; Venkatesh, B.; and Edakunni, N. U. 2020. FairXGBoost: Fairness-aware Classification in XGBoost. *arXiv preprint arXiv:2009.01442*.
- Royston, P. 2004. Multiple imputation of missing values. *The Stata Journal*, 4(3): 227–241.
- Rubin, D. B. 2004. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons.
- Schafer, J. L.; and Graham, J. W. 2002. Missing data: our view of the state of the art. *Psychological methods*, 7(2): 147.
- Stekhoven, D. J.; and Bühlmann, P. 2012. MissForest—nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1): 112–118.
- Tang, F.; and Ishwaran, H. 2017. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6): 363–377.
- Twala, B.; Jones, M.; and Hand, D. J. 2008. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, 29(7): 950–956.
- Ustun, B.; Liu, Y.; and Parkes, D. 2019. Fairness without harm: Decoupled classifiers with preference guarantees. In *International Conference on Machine Learning*, 6373–6382. PMLR.
- Van Buuren, S.; and Groothuis-Oudshoorn, K. 2011. mice: Multivariate imputation by chained equations in R. *Journal of statistical software*, 45(1): 1–67.
- Verwer, S.; and Zhang, Y. 2017. Learning decision trees with flexible constraints and objectives using integer optimization. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 94–103. Springer.
- Verwer, S.; and Zhang, Y. 2019. Learning optimal classification trees using a binary linear program formulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1625–1632.
- Wang, H.; Hsu, H.; Diaz, M.; and Calmon, F. P. 2021. To split or not to split: The impact of disparate treatment in classification. *IEEE Transactions on Information Theory*.
- Wang, Y.; and Singh, L. 2021. Analyzing the impact of missing values and selection bias on fairness. *International Journal of Data Science and Analytics*, 1–19.
- Wei, D.; Ramamurthy, K. N.; and Calmon, F. 2020. Optimized Score Transformation for Fair Classification. In *International Conference on Artificial Intelligence and Statistics*, 1673–1683. PMLR.
- White, I. R.; Royston, P.; and Wood, A. M. 2011. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4): 377–399.
- Wulff, J. N.; and Jeppesen, L. E. 2017. Multiple imputation by chained equations in praxis: guidelines and review. *Electronic Journal of Business Research Methods*, 15(1): 41–56.
- Zafar, M. B.; Valera, I.; Gomez-Rodriguez, M.; and Gummadi, K. P. 2019. Fairness Constraints: A Flexible Approach for Fair Classification. *J. Mach. Learn. Res.*, 20(75): 1–42.
- Zafar, M. B.; Valera, I.; Rodriguez, M.; Gummadi, K.; and Weller, A. 2017. From Parity to Preference-based Notions of Fairness in Classification. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *International conference on machine learning*, 325–333. PMLR.
- Zhang, Z. 2016. Missing data imputation: focusing on single imputation. *Annals of translational medicine*, 4(1).
- Zhao, H.; and Gordon, G. 2019. Inherent tradeoffs in learning fair representations. *Advances in neural information processing systems*, 32: 15675–15685.

## A Omitted Proofs

### A.1 Proof of Lemma 1

*Proof.* We assume that  $X_{\text{obs}} = \emptyset$ . In this case, MCAR and MAR are equivalent so we only focus on MCAR in what follows. First, we construct a probability distribution such that each group satisfies MCAR but the entire population does not. Let  $S \sim \text{Bernoulli}(0.5)$  and

$$\begin{aligned} X|S=0 &\sim \text{Bernoulli}(0.1), & X|S=1 &\sim \text{Bernoulli}(0.9), \\ M|S=0 &\sim \text{Bernoulli}(0.1), & M|S=1 &\sim \text{Bernoulli}(0.9). \end{aligned}$$

By construction, we know each group satisfies MCAR. However,

$$\begin{aligned} &\Pr(M=1, X=1) \\ &= \sum_s \Pr(M=1, X=1|S=s) \Pr(S=s) \\ &= \sum_s \Pr(M=1|S=s) \Pr(X=1|S=s) \Pr(S=s) \\ &= 0.1 \times 0.1 \times 0.5 + 0.9 \times 0.9 \times 0.5 = 0.41, \end{aligned}$$

and

$$\begin{aligned} &\Pr(M=1) \Pr(X=1) \\ &= \sum_s \Pr(M=1|S=s) \Pr(S=s) \sum_s \Pr(X=1|S=s) \Pr(S=s) \\ &= (0.1 \times 0.5 + 0.9 \times 0.5) \times (0.1 \times 0.5 + 0.9 \times 0.5) = 0.25. \end{aligned}$$

Hence,  $\Pr(M=1, X=1) \neq \Pr(M=1) \Pr(X=1)$  which means that the entire population does not satisfy MCAR.

Next, we construct a probability distribution such that the entire population satisfies MCAR but each group does not. Let  $S \sim \text{Bernoulli}(0.5)$ ,

$$\begin{aligned} \Pr(M=0, X=0|S=0) &= 0.1, & \Pr(M=0, X=1|S=0) &= 0.3, \\ \Pr(M=1, X=0|S=0) &= 0.4, & \Pr(M=1, X=1|S=0) &= 0.2, \end{aligned}$$

and

$$\begin{aligned} \Pr(M=0, X=0|S=1) &= 0.4, & \Pr(M=0, X=1|S=1) &= 0.2, \\ \Pr(M=1, X=0|S=1) &= 0.1, & \Pr(M=1, X=1|S=1) &= 0.3. \end{aligned}$$

By construction,

$$\begin{aligned} \Pr(M=0, X=0) &= 0.25, & \Pr(M=0, X=1) &= 0.25, \\ \Pr(M=1, X=0) &= 0.25, & \Pr(M=1, X=1) &= 0.25. \end{aligned}$$

As a result, the entire population satisfies MCAR but each group does not satisfy this assumption.  $\square$

### A.2 Proof of Theorem 1

*Proof.* We denote  $\alpha \triangleq f_{\text{imp}}(\ast)$  and let  $p_s^{\text{ms}} \triangleq \Pr(S=s|M=1)$ ,  $m_s \triangleq \mathbb{E}[X|S=s]$  for  $s \in \{0, 1\}$ . Then

$$\begin{aligned} &\mathbb{E}[(f_{\text{imp}}(\tilde{X}) - X)^2 | M=1] \\ &= \sum_s \Pr(S=s|M=1) \mathbb{E}[(\alpha - X)^2 | M=1, S=s] \\ &= \sum_s p_s^{\text{ms}} \mathbb{E}[(\alpha - X)^2 | S=s]. \end{aligned}$$

Consequently, we have

$$\begin{aligned} \alpha^* &= \operatorname{argmin}_{\alpha} \sum_s p_s^{\text{ms}} \mathbb{E}[(\alpha - X)^2 | S=s] \\ &= \operatorname{argmin}_{\alpha} \sum_s p_s^{\text{ms}} (\alpha - m_s)^2 \\ &= \frac{p_0^{\text{ms}} m_0 + p_1^{\text{ms}} m_1}{p_0^{\text{ms}} + p_1^{\text{ms}}} \\ &= p_0^{\text{ms}} m_0 + p_1^{\text{ms}} m_1. \end{aligned}$$

Therefore, the optimal imputation method is unique and has a closed-form expression:

$$f_{\text{imp}}^*(\tilde{x}) = \begin{cases} \tilde{x} & \text{if } \tilde{x} \in \mathcal{X} \\ \alpha^* & \text{if } \tilde{x} = *. \end{cases}$$

The performance of the optimal imputation method on group  $s$  is

$$\begin{aligned} \mathbb{E} [(f_{\text{imp}}^*(\tilde{X}) - X)^2 \mid M = 1, S = s] &= \mathbb{E} [(\alpha^* - X)^2 \mid S = s] \\ &= (p_0^{\text{ms}} m_0 + p_1^{\text{ms}} m_1 - m_s)^2 + \text{Var}[X \mid S = s] \\ &= (m_1 - m_0)^2 (p_{1-s}^{\text{ms}})^2 + \text{Var}[X \mid S = s]. \end{aligned}$$

Finally, we can compute the discrimination risk of the optimal imputation method:

$$\begin{aligned} &\left| \mathbb{E} [(f_{\text{imp}}^*(\tilde{X}) - X)^2 \mid M = 1, S = 0] - \mathbb{E} [(f_{\text{imp}}^*(\tilde{X}) - X)^2 \mid M = 1, S = 1] \right| \\ &= |(p_1^{\text{ms}} - p_0^{\text{ms}})(m_1 - m_0)^2 + (\text{Var}[X \mid S = 0] - \text{Var}[X \mid S = 1])|. \end{aligned}$$

□

### A.3 Proof of Theorem 2

*Proof.* Since data from each group are MCAR, the quantity  $L_s(h \circ f_{\text{imp}}^{\text{test}})$  is equal to

$$\begin{aligned} &\mathbb{E} [\ell(h \circ f_{\text{imp}}^{\text{test}}(\tilde{X}), Y) \mid M = 1, S = s] p_s + \mathbb{E} [\ell(h \circ f_{\text{imp}}^{\text{test}}(\tilde{X}), Y) \mid M = 0, S = s] (1 - p_s) \\ &= \mathbb{E} [\ell(h \circ f_{\text{imp}}^{\text{test}}(\tilde{X}), Y) \mid M = 1, S = s] p_s + \mathbb{E} [\ell(h(X), Y) \mid S = s] (1 - p_s). \end{aligned} \quad (9)$$

Now we can rewrite the first term as

$$\mathbb{E} [\ell(h \circ f_{\text{imp}}^{\text{test}}(\tilde{X}), Y) \mid M = 1, S = s] = \int \ell(h(x), y) dP_s^{\text{test}}(x, y) \quad (10)$$

where  $P_s^{\text{test}}$  is the probability distribution of  $(f_{\text{imp}}^{\text{test}}(\tilde{X}), Y) \mid M = 1, S = s$ . Combining (9) with (10) yields

$$L_s(h \circ f_{\text{imp}}^{\text{test}}) = p_s \int \ell(h(x), y) dP_s^{\text{test}}(x, y) + \mathbb{E} [\ell(h(X), Y) \mid S = s] (1 - p_s). \quad (11)$$

Similarly, we have

$$L_s(h \circ f_{\text{imp}}^{\text{train}}) = p_s \int \ell(h(x), y) dP_s^{\text{train}}(x, y) + \mathbb{E} [\ell(h(X), Y) \mid S = s] (1 - p_s) \quad (12)$$

where  $P_s^{\text{train}}$  is the probability distribution of  $(f_{\text{imp}}^{\text{train}}(\tilde{X}), Y) \mid M = 1, S = s$ . Since the loss function is bounded between 0 and  $K$ , the variational representation of total variation distance (see Section 6.3 in Polyanskiy and Wu 2019) implies

$$\left| \int \ell(h(x), y) dP_s^{\text{train}}(x, y) - \int \ell(h(x), y) dP_s^{\text{test}}(x, y) \right| \leq K D_{\text{TV}}(P_s^{\text{train}} \| P_s^{\text{test}}). \quad (13)$$

By the triangle inequality and (11–13), we have

$$|L_0(h \circ f_{\text{imp}}^{\text{test}}) - L_1(h \circ f_{\text{imp}}^{\text{test}})| \leq |L_0(h \circ f_{\text{imp}}^{\text{train}}) - L_1(h \circ f_{\text{imp}}^{\text{train}})| + K \sum_s p_s D_{\text{TV}}(P_s^{\text{train}} \| P_s^{\text{test}}). \quad (14)$$

Finally, we prove that the above inequality is tight. Let the loss function be the 0-1 loss and  $X = (X_{\text{obs}}, X_{\text{ms}}) \in [0, 1]^2$ ,  $Y \in \{0, 1\}$ . Consider a binary classifier  $h(x_1, x_2) = \mathbb{I}[x_2 \geq 0.5]$  and different imputation methods during training and testing time:  $f_{\text{imp}}^{\text{train}}(x_{\text{obs}}, *) = (x_{\text{obs}}, 0)$  and  $f_{\text{imp}}^{\text{test}}(x_{\text{obs}}, *) = (x_{\text{obs}}, \mathbb{I}[x_{\text{obs}} \leq 0.5])$ . Furthermore, we let the missing probability  $p_s = 1$  and  $P_{\tilde{X}, Y \mid M=1, S=0} = \delta_{(0, *), 0}$ ,  $P_{\tilde{X}, Y \mid M=1, S=1} = \delta_{(1, *), 0}$ . In this case,  $L_0(h \circ f_{\text{imp}}^{\text{train}}) = L_1(h \circ f_{\text{imp}}^{\text{train}}) = 0$ ,  $L_0(h \circ f_{\text{imp}}^{\text{test}}) = 1$ ,  $L_1(h \circ f_{\text{imp}}^{\text{test}}) = 0$ . Consequently, the LHS and RHS of (14) are both 1. □

#### A.4 Proof of Theorem 3

*Proof.* Let the loss function used for defining  $L(h)$  and  $L_s(h)$  be the 0-1 loss and  $X = (X_{\text{obs}}, X_{\text{ms}}) \in [0, 1]^2$ ,  $Y \in \{0, 1\}$ . Furthermore, we let  $\Pr(S = 0) = 0.5$ ,  $\Pr(M = 1) = 1$ , and  $P_{\tilde{X}, Y|M=1, S=0} = \delta_{(0, *), 1}$ ,  $P_{\tilde{X}, Y|M=1, S=1} = \delta_{(1, *), 0}$ . Now consider two hypothesis classes:

$$\begin{aligned}\mathcal{H}_1 &\triangleq \{\mathbb{I}[x_1 \geq a] \mid a \in [0, 1]\}, \\ \mathcal{H}_2 &\triangleq \{\mathbb{I}[x_2 \geq a] \mid a \in [0, 1]\}.\end{aligned}$$

For  $\epsilon < 0.5$ , the minimum in (7) under the imputed data distribution and hypothesis class  $\mathcal{H}_1$  is always 1.0 no matter which imputation method is used. In other words, for  $\epsilon, \delta < 0.5$ , the class of  $(\epsilon, \delta)$ -conformal imputation methods under  $\mathcal{H}_1$  is empty. However, for another hypothesis class,  $f_{\text{imp}}(x_{\text{obs}}, *) = (x_{\text{obs}}, \mathbb{I}[x_{\text{obs}} \leq 0])$  is a  $(\epsilon, \delta)$ -conformal imputation method and  $h(x_1, x_2) = \mathbb{I}[x_2 \geq 0.5] \in \mathcal{H}_2$  gives a perfect solution of (7).  $\square$

## B Mixed Integer Programming for Fair Decision Tree with Missing Values

The full integer program for training a fair decision tree with MIA is given in Program 1. We first explain the variables and then walk through each constraint in the program.

As explained in Section 4.1,  $p_{v,j}, q_v, c_v, u_l$  are parameters that determine the tree ( $v \in \mathcal{V}, j \in [d], l \in \mathcal{L}$ ).  $w_{i,v}^{(1)}, w_{i,v}^{(2)}, w_{i,v}^{(\text{nm})}$  and  $z_{i,l}$  are parameters associated with the training data point  $(\mathbf{x}_i, y_i)$ .  $w_{i,v} \in \{0, 1\}$  determines whether the  $i$ -th data point goes to the left or to the right at the branching node  $v$ . This is computed for all  $v \in \mathcal{V}$  regardless of whether the node is on the path to its destination leaf node or not.  $w_{i,v}^{(1)}, w_{i,v}^{(2)}, w_{i,v}^{(\text{nm})}$  are auxiliary variables for computing  $w_{i,v}$ .  $\mathbf{z}_i$  is an one-hot-encoded vector of length  $|\mathcal{L}|$  that encodes the destination leaf node of the  $i$ -th data point, i.e.,  $z_{i,l} = 1$  indicates that the leaf  $l$  is the destination leaf node for the  $i$ -th data point.  $f_{l,0}, f_{l,1}$ , and  $\text{loss}(l)$  are variables used to compute  $\ell(\mathcal{D})$  and  $\ell_{\text{fair}}(\mathcal{D})$  in the objective.

The first constraint in (17) enforces the one-hot-encoding of  $\mathbf{p}_v$  and  $\mathbf{z}_i$ . Constraints in (18)–(23) are used to obtain  $w_{i,v}$ . (18) and (19) encode the following logical constraint:

$$w_{i,v}^{(\text{nm})} = \begin{cases} 1, & \text{if } q_v \geq \sum_{j \in [d]} p_{v,j} (1 - m_{i,j}) x_{i,j}, \\ 0, & \text{otherwise.} \end{cases}$$

$M \in \mathbb{R}$  in (18) is a constant chosen to be large enough so that the left hand side (LHS) is always smaller than  $M$ , and  $\epsilon$  is a small constant to close to zero (e.g., 0.001). Having  $\epsilon$  allows for numerical errors in the real-number representation of binary variables. Notice that  $w_{i,v}^{(\text{nm})}$  is 1 when the selected feature at node  $v$  is not missing and smaller than the threshold  $q_v$ . However, when the feature is missing, the right hand side (RHS) is always zero. Then,  $w_{i,v}^{(\text{nm})}$  becomes equivalent to a logical variable that represents if  $q_v \geq 0$ . This is a condition does not say anything about where the given data point should go, i.e., the condition encoded by  $w_{i,v}^{(\text{nm})}$  is relevant only when the value is not missing. We introduce another variable  $w_{i,v}^{(1)}$  that is 1 only if  $w_{i,v}^{(\text{nm})} = 1$  and also the value is not missing:

$$w_{i,v}^{(1)} = \begin{cases} 1, & \text{if } \left( \sum_{j \in [d]} p_{v,j} \cdot (1 - m_{i,j}) > 0 \right) \text{ AND } \left( w_{i,v}^{(\text{nm})} > 0 \right), \\ 0, & \text{otherwise,} \end{cases}$$

and this is obtained through constraints (20) and (21). When the feature is missing, we use  $w_{i,v}^{(2)}$  to determine the splitting:

$$w_{i,v}^{(2)} = \begin{cases} 1, & \text{if } \left( \sum_{j \in [d]} p_{v,j} \cdot m_{i,j} > 0 \right) \text{ AND } (c_v > 0), \\ 0, & \text{otherwise,} \end{cases}$$

and this is computed through line (22). Finally,  $w_{i,v}$  can be obtained from:

$$w_{i,v} = w_{i,v}^{(1)} \text{ OR } w_{i,v}^{(2)} \quad (\text{line (23) in Program 1}).$$

This can be interpreted as: the data point  $(\mathbf{x}_i, y_i)$  will follow the left branch if the value is not missing and satisfies the threshold to go to the left ( $x_i \leq q_v$ ), or if the value is missing and  $c_v = 1$ . Otherwise, it goes to the right branch. When we have  $w_{i,v}$  for all  $v \in \mathcal{V}$ ,  $z_{i,l}$  can be determined through the conditions given in (24).

As we consider binary classification problem, we choose  $u_l$ , prediction at leaf  $l$ , based on the majority rule:

$$u_l = \begin{cases} 1, & \text{if } \sum_{i \in [n]} y_i z_{i,l} \geq \sum_{i \in [n]} (1 - y_i) z_{i,l}, \\ 0, & \text{otherwise.} \end{cases}$$

This is modeled into constraints (25),(26). For the loss function  $\ell(\mathcal{D})$ , we use 0-1 loss,  $\ell(\mathcal{D}) = \frac{\sum_{l \in \mathcal{L}} \text{loss}(l)}{n}$ , where

$$\text{loss}(l) = \begin{cases} \sum_{i \in [n]} (1 - y_i) z_{i,l}, & \text{if } u_l = 1, \\ \sum_{i \in [n]} y_i z_{i,l}, & \text{if } u_l = 0. \end{cases} \quad (15)$$

Since this is nonlinear with respect to the variables, we model it through constraints given in (27),(28).

For the fairness regularizer, we use group fairness metrics based on confusion matrix: FNR difference, FPR difference, equalized odds (i.e., both FNR and FPR difference), and accuracy difference. We first describe how to implement accuracy difference (i.e.,  $|\Pr(Y \neq \hat{Y}|S=0) - \Pr(Y \neq \hat{Y}|S=1)|$  as  $\ell_{\text{fair}}$ ). To compute this, we introduce auxiliary variables  $f_{l,0}$  and  $f_{l,1}$  ( $l \in \mathcal{L}$ ) that denote the number of misclassified points per leaf, for group 0 and group 1, respectively. They can be written as:

$$f_{l,0} = \begin{cases} \sum_{i \in [n]} (1 - y_i)(1 - s_i) z_{i,l}, & \text{if } u_l = 1, \\ \sum_{i \in [n]} y_i(1 - s_i) z_{i,l}, & \text{if } u_l = 0. \end{cases}$$

$$f_{l,1} = \begin{cases} \sum_{i \in [n]} (1 - y_i)s_i z_{i,l}, & \text{if } u_l = 1, \\ \sum_{i \in [n]} y_i s_i z_{i,l}, & \text{if } u_l = 0. \end{cases}$$

$f_{l,0}$ 's are computed through (29)–(31) and  $f_{l,1}$ 's are computed through (32)–(34). Then, the accuracy difference between group 0 and group 1 is given as:

$$\ell_{\text{fair}}(\mathcal{D}) = \left| \frac{\sum_{l \in \mathcal{L}} f_{l,0}}{\sum_{i \in [n]} (1 - s_i)} - \frac{\sum_{l \in \mathcal{L}} f_{l,1}}{\sum_{i \in [n]} s_i} \right| \quad (16)$$

Although the term inside the absolute value is a linear combination of the variables, taking the absolute value makes this non-linear. Hence, instead of having (16) in the objective function directly, we model this into constraints (35) and (36).

Now we describe how this can be modified to regularize to FPR or FNR difference. To use FPR difference, we set the first terms in (31), (34) to zero, i.e.,

$$\begin{aligned} f_{l,0} &\geq 0 - M u_l, \quad f_{l,1} \geq 0 - M u_l, \quad \forall l \in \mathcal{L} \\ f_{l,0} &\leq 0 + M u_l + \epsilon, \quad f_{l,1} \leq 0 + M u_l + \epsilon, \quad \forall l \in \mathcal{L}. \end{aligned}$$

Additionally, we modify  $\ell_{\text{fair}}$  to:

$$\ell_{\text{fair}}(\mathcal{D}) = \left| \frac{\sum_{l \in \mathcal{L}} f_{l,0}}{\sum_{i \in [n]} (1 - s_i)(1 - y_i)} - \frac{\sum_{l \in \mathcal{L}} f_{l,1}}{\sum_{i \in [n]} s_i(1 - y_i)} \right|$$

Similarly, to use FNR difference as a regularizer, we set the first terms in (29), (30), (32),(33) to zero:

$$\begin{aligned} f_{l,0} &\geq 0 - M(1 - u_l), \quad f_{l,1} \geq 0 - M(1 - u_l), \quad \forall l \in \mathcal{L}, \\ f_{l,0} &\leq 0 + M(1 - u_l) + \epsilon, \quad f_{l,1} \leq 0 + M(1 - u_l) + \epsilon, \quad \forall l \in \mathcal{L}, \text{ and} \end{aligned}$$

set  $\ell_{\text{fair}}$  to:

$$\ell_{\text{fair}}(\mathcal{D}) = \left| \frac{\sum_{l \in \mathcal{L}} f_{l,0}}{\sum_{i \in [n]} (1 - s_i)y_i} - \frac{\sum_{l \in \mathcal{L}} f_{l,1}}{\sum_{i \in [n]} s_i y_i} \right|$$

To use equalized odds as a regularizer, we can use above formulas to compute the FPR difference and FNR difference separately, and regularize:

$$|\Pr(Y \neq \hat{Y}|S=0, Y=0) - \Pr(Y \neq \hat{Y}|S=1, Y=0)| + |\Pr(Y \neq \hat{Y}|S=0, Y=1) - \Pr(Y \neq \hat{Y}|S=1, Y=1)|.$$

Major differences in our formulation from the previous works (Bertsimas and Dunn 2017; Aghaei, Azizi, and Vayanos 2019) are:

- We implement fairness regularizers that are based on the performance difference between two groups – FNR difference, FPR difference, equalized odds, and accuracy difference – as compared to Aghaei, Azizi, and Vayanos (2019), which only considered statistical parity. We add leaf-wise fairness risk variables  $f_{l,0}$  and  $f_{l,1}$  to implement this.
- In our formulation, we have to search for an optimal  $\mathbf{c}$  (missing value splitting criteria) in addition to  $\mathbf{P}$  and  $\mathbf{q}$ , which are used for conventional non-missing splitting. A straightforward way to translate this into integer programming leads to a quadratic program. To make it linear, we add multiple intermediate variables, such as  $w^{(1)}$ ,  $w^{(2)}$  and  $w^{(nm)}$ .

## C Experiment Details

### C.1 Datasets

We use three different datasets for evaluation: COMPAS, Adult, and High School Longitudinal Study (HSLS). While COMPAS and Adult are widely used datasets in the fair machine learning literature, we believe that this paper is the first work to study the fair ML aspect of the HSLS dataset. We first give brief introduction to the dataset and then illustrate how missing patterns in the real-world survey data can have disparate missing patterns.

**Description of HSLS Dataset.** This dataset consists of 23,000+ participants from 944 high schools who were followed from the 9th through 12th grade. It includes surveys from students, parents, and teachers, student demographic information, school information, and students academic performance across several years. The goal is to predict student’s 9th-grade math test performance from relevant variables collected prior to the test. The original dataset has thousands of features, but for our analysis, we only utilize a set of 11 features:

VARIABLE	DESCRIPTION	TYPE
$XIRACE$	Student’s race/ethnicity	Categorical
$XIMTHID$	Student’s mathematics identity	Continuous
$XIMTHUTI$	Student’s mathematics utility	Continuous
$XIMTHEFF$	Student’s mathematics self-efficacy	Continuous
$XIPAR2EDU$	Secondary caregiver’s highest level of education	Categorical
$XIFAMINCOME$	Total family income	Continuous
$XIPIRELATION$	Relationship between student and the primary caregiver	Categorical
$XIPARIEMP$	Primary caregiver’s employment status	Categorical
$XISCHOOLBEL$	Student’s sense of school belonging	Continuous
$XISTU30OCC2$	Student desired occupation at age 30	Categorical
$XITXMSCR$	Student’s mathematics standardized test score	Continuous

Table 1: Description of features used in the HSLS dataset

We use  $XIRACE$  to generate a binary group attribute: White/Asian (WA) and under-represented minority (URM) that includes Black, Hispanic, Native American, and Pacific Islanders. We create a binary label from the continuous test score  $XITXMSCR$ , to perform a binary classification on whether a student belongs to the top 50% performers or the bottom 50% performers. As we do not consider the case when the group attribute or the label are missing, we drop data points that are missing  $XIRACE$  or  $XITXMSCR$ . We scale every variable to be between 0 and 1.

**Illustration of disparate missing patterns in the HSLS dataset.** We show a real-world example of disparate missing patterns in the HSLS dataset. Between male and female students, male students consistently had 1-2% more missing values in all variables. In Table 2, we summarize missing probabilities of some variables between different demographic groups: male vs. female and WA vs. URM. Between WA students and URM students, URM students always had higher missing probabilities in all variables. However, the difference in missing probabilities varied widely depending on the variables. Within the student survey variables,  $XIMTHID$  had only 3% difference between WA and URM, and  $XIMTHEFF$  had about 6% difference. For parent survey variables, URM consistently had 6-8% higher missing probabilities. However, for the questions related to secondary caregivers, URM had a significantly higher missing rate, e.g. for  $XIPAR2EDU$ , the difference was more than 15%. On the other hand, between genders, while the questions on secondary caregivers have a higher missing probability in general, the difference between males and females was not significant. Without an in-depth analysis, it is not straightforward to detect how missing patterns will vary depending on which group attributes.

**Description of COMPAS and Adult datasets.** For COMPAS dataset, we use eight features:  $age\_cat\_25-45$ ,  $age\_cat\_Greater-than-45$ ,  $age\_cat\_Less-than-25$ ,  $race$ ,  $sex$ ,  $priors\_count$ ,  $c\_charge\_degree$ ,  $two\_year\_recid$ . We use  $race$  as a group attribute (0: Black, 1: White) and  $two\_year\_recid$  as a binary label (1 indicates recidivating within two years and 0 otherwise). We balance the dataset to have equal number of Black and White subjects to isolate the issue of data imbalance out of our analysis. After balancing, we had a total of 4206 data points.

For Adult dataset, we use the following features:  $age$ ,  $workclass$ ,  $education$ ,  $education-num$ ,  $marital-status$ ,  $occupation$ ,  $relationship$ ,  $race$ ,  $gender$ ,  $capital-gain$ ,  $capital-loss$ ,  $hours-per-week$ ,  $native-country$ , and  $income$ . We use  $gender$  as a group attribute (0: female, 1: male) and  $income$  as a label (0: income  $\leq$  50K, 1: > 50K). For Adult, we balance the dataset both in

VARIABLE	DESCRIPTION	MISSING PROBABILITIES (%)			
		MALE	FEMALE	WA	URM
<i>XIMTHID</i>	Student’s mathematics identity	10.6 ± 0.3	9.3 ± 0.3	4.8 ± 0.2	7.8 ± 0.3
<i>XIMTHEFF</i>	Student’s mathematics self-efficacy	21.2 ± 0.4	19.1 ± 0.4	14.0 ± 0.3	20.1 ± 0.4
<i>SIAPCALC</i>	9th grader plans to enroll in an Advanced Placement (AP) calculus course	14.2 ± 0.3	12.1 ± 0.3	7.6 ± 0.2	12.1 ± 0.3
<i>XIPAR1EDU</i>	Primary caregiver’s highest level of education	29.6 ± 0.4	27.5 ± 0.4	22.8 ± 0.4	29.8 ± 0.5
<i>XIPAR2EDU</i>	Secondary caregiver’s highest level of education	44.5 ± 0.4	43.4 ± 0.5	35.8 ± 0.4	<b>51.0 ± 0.5</b>

Table 2: Data missing probabilities of different variables between demographic groups: male vs. female or white vs. under-represented minority (URM) in the HSLS dataset

terms of the group attribute and the label, as the original dataset is highly imbalanced, having substantially more data points with label 0. After balancing, we had a total of 7834 data points.

As these two datasets do not contain any missing values, we artificially created missing values. The missing statistics we created is summarized below:

Dataset	Feature	$p_0^{\text{ms}}$	$p_1^{\text{ms}}$
Adult	marital-status	0.0	0.4
	hours-per-week	0.0	0.3
	race	0.2	0.2
COMPAS	priors_count	0.4	0.1
	sex	0.6	0.2

Table 3: Missing statistics for Adult and COMPAS datasets we generated for our experiments.

As the goal of the paper is to examine the fairness issues and their remedy, we tested different combinations of features with varying  $p_0$  and  $p_1$ , where  $p_0 = \Pr(M = 1|S = 0)$  and  $p_1 = \Pr(M = 1|S = 1)$  for each missing variable. We varied  $p_0$  and  $p_1$  from 0.0 to 0.9, and chose the missing pattern that had considerable difference in FNR of FPR between the two groups defined by the group attribute.

## C.2 Hyperparameters

For the Fair MIP Forest algorithm, there are four hyperparameters we can choose: tree depth ( $D$ ), number of trees ( $n_{\text{tree}}$ ), time limit for training a single tree ( $t_{\text{limit}}$ ), and the batch size. We chose  $D = 3$  for all experiments as we did not see much improvement in accuracy when going from  $D = 3$  to  $D = 4, 5$ . We have tried  $n_{\text{tree}} = 10, 20, 30, 40$ ,  $t_{\text{limit}} = 60, 90, 120$  (s), batch size = 200, 400, 800, and picked the ones with best performance. If the performance was similar, we chose parameters that have smaller computational cost (e.g. smaller  $n_{\text{tree}}$ ). The chosen hyperparameters are summarized below:

Dataset	$t_{\text{limit}}$	$n_{\text{tree}}$	batch_size	$\lambda$
COMPAS	60	30	200	{0.1, 0.5, 1.0}
Adult	60	30	200	{0.1, 0.14, 0.17, 0.5, 0.8, 2.0 }
HSLS	60	30	400	{ 0.1, 1.0 ,3.0 }

Table 4: Summary of hyperparameters used in Fair MIP Forest.

We varied  $\lambda$  in Fair MIP Forest algorithm to moderate how much we regularize fairness metrics. We varied  $\lambda$  from 0.01 to 20. The final lambda values plotted in Figure 2 are given in Table 4. For Zafar (Zafar et al. 2019), we varied  $\tau$  to get different points on the fairness-accuracy trade-off plot. The set of  $\tau$  values we tried is: {0.001, 0.01, 0.1, 1, 10, 100}. For Agarwal (Agarwal et al. 2018), we varied  $\epsilon$  to achieve different fairness-accuracy trade-off points, and the set of values used is: {0.001, 0.005, 0.01, 0.02, 0.05, 0.1}. For all methods, we drop points under the convex curve and only keep the best performing points on the plot.

For Agarwal and Hardt, we train a decision tree classifier with the same parameters as the baseline (i.e. decision trees with depth 3). For Zafar, we use logistic regression, as it requires a distance-based classifier.

### C.3 Implementation Details

Agarwal and Hardt are implemented with AIF360 ([Bellamy et al. 2018](#)). For Zafar, we use the code in <https://github.com/mbilalzafar/fair-classification>.

---

Program 1: MIP Formulation for Fair Decision Tree Training with MIA

---

$$\begin{aligned}
& \text{minimize} \quad \ell(\mathcal{D}) + \lambda \cdot \ell_{\text{fair}}(\mathcal{D}) \\
& q_v, \text{loss}(l), f_{l,0}, f_{l,1}, \ell_{\text{fair}}(\mathcal{D}) \in \mathbb{R} \quad \text{for } v \in \mathcal{V}, l \in \mathcal{L} \\
& w_{i,v}, w_{i,v}^{(1)}, w_{i,v}^{(2)}, w_{i,v}^{(\text{nm})}, p_{v,j}, z_{i,l}, c_v, u_l \in \{0, 1\} \quad \text{for } i \in [n], v \in \mathcal{V}, j \in [d], l \in \mathcal{L}
\end{aligned}
\tag{17}$$

$$\text{subject to: } \sum_{j \in [d]} p_{v,j} = 1, \sum_{l \in \mathcal{L}} z_{i,l} = 1 \tag{17}$$

$$q_v - \sum_{j \in [d]} p_{v,j} (1 - m_{i,j}) x_{i,j} \leq M w_{i,v}^{(\text{nm})} - \epsilon (1 - w_{i,v}^{(\text{nm})}) \tag{18}$$

$$q_v - \sum_{j \in [d]} p_{v,j} (1 - m_{i,j}) x_{i,j} \geq -M (1 - w_{i,v}^{(\text{nm})}) \tag{19}$$

$$w_{i,v}^{(1)} + 1 \geq (1 - \sum_{j \in [d]} p_{v,j} m_{i,j}) + w_{i,v}^{(\text{nm})} \tag{20}$$

$$w_{i,v}^{(1)} \leq (1 - \sum_{j \in [d]} p_{v,j} m_{i,j}), \quad w_{i,v}^{(1)} \leq w_{i,v}^{(\text{nm})} \tag{21}$$

$$w_{i,v}^{(2)} + 1 \geq \sum_{j \in [d]} p_{v,j} m_{i,j} + c_v, \quad w_{i,v}^{(2)} \leq \sum_{j \in [d]} p_{v,j} m_{i,j}, \quad w_{i,v}^{(2)} \leq c_v \tag{22}$$

$$w_{i,v} \geq w_{i,v}^{(1)}, \quad w_{i,v} \geq w_{i,v}^{(2)} \tag{23}$$

$$z_{i,l} \leq w_{i,v}, \quad \forall l \in \mathcal{L}^L(v), \quad z_{i,l} \leq 1 - w_{i,v}, \quad \forall l \in \mathcal{L}^R(v) \tag{24}$$

$$\sum_{i \in [n]} 2(y_i - 1) z_{i,l} \leq M u_l - \epsilon (1 - u_l) \tag{25}$$

$$\sum_{i \in [n]} 2(y_i - 1) z_{i,l} \geq M (1 - u_l) - \epsilon (1 - u_l) \tag{26}$$

$$\text{loss}(l) \leq \sum_{i \in [n]} (1 - y_i) z_{i,l}, \quad \text{loss}(l) \leq \sum_{i \in [n]} y_i z_{i,l} \tag{27}$$

$$\text{loss}(l) \geq \sum_{i \in [n]} (1 - y_i) z_{i,l} - M (1 - u_l), \quad \text{loss}(l) \geq \sum_{i \in [n]} y_i z_{i,l} - M u_l \tag{28}$$

$$f_{l,0} \geq \sum_{i \in [n]} (1 - y_i) (1 - s_i) z_{i,l} - M (1 - u_l) \tag{29}$$

$$f_{l,0} \leq \sum_{i \in [n]} (1 - y_i) (1 - s_i) z_{i,l} + M (1 - u_l) + \epsilon \tag{30}$$

$$f_{l,0} \geq \sum_{i \in [n]} y_i (1 - s_i) z_{i,l} - M u_l, \quad f_{l,0} \leq \sum_{i \in [n]} y_i (1 - s_i) z_{i,l} + M u_l + \epsilon \tag{31}$$

$$f_{l,1} \geq \sum_{i \in [n]} (1 - y_i) s_i z_{i,l} - M (1 - u_l) \tag{32}$$

$$f_{l,1} \leq \sum_{i \in [n]} (1 - y_i) s_i z_{i,l} + M (1 - u_l) + \epsilon \tag{33}$$

$$f_{l,1} \geq \sum_{i \in [n]} y_i s_i z_{i,l} - M u_l, \quad f_{l,1} \leq \sum_{i \in [n]} y_i s_i z_{i,l} + M u_l + \epsilon, \tag{34}$$

$$\ell_{\text{fair}}(\mathcal{D}) \geq \frac{\sum_{l \in \mathcal{L}} f_{l,0}}{\sum_{i \in [n]} (1 - S_i)} - \frac{\sum_{l \in \mathcal{L}} f_{l,1}}{\sum_{i \in [n]} s_i} \tag{35}$$

$$\ell_{\text{fair}}(\mathcal{D}) \geq - \left( \frac{\sum_{l \in \mathcal{L}} f_{l,0}}{\sum_{i \in [n]} (1 - S_i)} - \frac{\sum_{l \in \mathcal{L}} f_{l,1}}{\sum_{i \in [n]} S_i} \right) \tag{36}$$

( $\forall v \in \mathcal{V}, \forall j \in [d], \forall l \in \mathcal{L}, \forall i \in [n]$ , except line (24))

---