On the Effects of Missing Data Imputation on Classification Fairness

Anonymous Authors (from Computer Science and Data Science)

Abstract

In recent years, a great deal of research has focused on bias in machine learning, and in areas such as criminal sentencing, such bias may result in catastrophic consequences. One aspect of fairness that has been understudied is the effect of missing data imputation on classification fairness. In real-world applications, data for training is often incomplete, and to fill the missing values, data imputation methods are used before training a model. Here, we perform a comprehensive study of the effects of data imputation on classification fairness. We show that- as with imputation and classification accuracy- the interaction between data imputation and fairness is complex, but that the imputation strategy known as multiple imputation often results in the lowest bias. Next, we propose a novel data imputation heuristic known as Impute From Opposite, and show that it generally results in similar classification accuracy and lower bias than its counterparts.

Introduction

Over the last few decades, machine learning algorithms have evolved rapidly, and have penetrated into almost all aspects of society. However, as is now well known, use of such algorithms may result in harmful bias against groups of individuals. For example, as was famously discussed by ProPublica in (Angwin et al. 2016) (and challenged in (Larson and Angwin 2016)), criminal sentencing instruments may be systematically biased against African Americans. In response to such examples, the work on *fair machine learning* seeks to identify causes of bias and propose remedies to such bias. However, in real world applications, datasets are often incomplete. Thus, when training a machine learning model, *data imputation* is often necessary to 'fill in' missing values in a dataset so that a classifier can then be trained.

In this paper, we (a) examine the effect of several popular data imputation techniques on the fairness of the predictions made by machine learning classifiers, and (b) propose a simple heuristic that can easily be used together with these data imputation techniques to reduce unfairness in the final classifications. We consider three popular data imputation techniques: imputation by **Mean**- which replaces each missing value with the mean value for that feature, imputation by **Similar** (also referred to as k-nearest neighbors imputation)- which identifies other data points similar to the one with a missing value, and imputes the missing value with the mean of the corresponding feature value of those data points, and **Multiple** imputation- which, for each missing value, creates a set of imputed values with the goal of properly capturing properties of the distribution of feature values.

We evaluate these data imputation techniques on six popuar machine learning algorithms, ranging from the simple **Logistic Regression** to the **Multilayer Perceptron** neural network algorithm, on six real datasets commonly used in algorithmic fairness research. We demonstrate that, as with imputation and accuracy, the relationship between imputation and fairness is complex, but that **Multiple** imputation generally results in lower bias than the other methods.

We then describe **Impute From Opposite** imputation, a framework that imputes missing values by looking at values from other protected groups. We show that when used with **Similar** or **Multiple** imputation, for missing data fractions below 50%, **Impute From Opposite** generally results in lower bias and similar accuracy to the original method.

To our knowledge, this is the first work to comprehensively study the effect of various data imputation techniques on classification fairness. While there is an existing work by Martinez-Plumed *et al.* that looks at data imputation from a fairness perspective, the goal of that paper is primarily to explore the fairness of the dataset with and without missing values, and only considers **Mean** imputation (Martínez-Plumed et al. 2019).

The major contributions of our work are as follows:

1. We perform a comprehensive study of the effect of data imputation on classification fairness. Our study includes three data imputation algorithms, six real-world datasets, and six classification algorithms.
2. We propose **Impute From Opposite**, a novel fairness-focused data imputation framework that can be used in conjunction with existing data imputation algorithms. We show that at values of missingness below 50%, **Impute From Opposite** gives similar accuracy but lower bias than its standard counterparts.

The rest of this paper is structured as follows: first, we give an overview of related work. Next, we provide a brief background on data imputation. We then give a description of our experimental setup, and present the results of our analysis. Next, we propose a data imputation framework that can accommodate existing data . Finally, we conclude our work by summarizing several takeaway messages and discuss avenues for future work.

Related Work

In recent years, topics related to fairness in machine learning have received a great deal of attention. This literature typically assumes the existence of protected groups defined on the basis of a protected attribute such as race or gender. ‘Unfairness’ occurs when these protected groups are systematically and wrongly treated differently by an algorithm (e.g., receive different classification outcomes at rates not justified by differences in the underlying distributions). Algorithmic unfairness can be caused by many different factors, including bias in the data (Mehrabi et al. 2019), underrepresentation of protected groups in the data (Mehrabi et al. 2019), etc. To measure unfairness, there are a number of metrics that may be used, including disparate impact, equal opportunity difference, and others (Mehrabi et al. 2019). These metrics generally look at the number (fraction) of individuals from each protected group who receive positive or negative classifications, and the accuracy of those classifications.

The fairness metric used in this paper is based on *Equal Opportunity*, which states that the True Positive Rate should be equal for 2 protected groups (Hardt, Price, and Srebro 2016a). A full description of our metric is provided in Section .

Because real datasets often have missing values, it is of interest to study *missing data imputation* in the context of fairness. Data imputation is an important and active research area in statistics and machine learning. Simple techniques for data imputation include filling in missing values by the mean of the column (feature) or finding rows that are similar with respect to other features and taking the average of the missing feature value from those rows (De Leeuw 2001). The current state-of-the-art in the statistics community are the many variations of *multiple imputation*, which uses a multivariate distribution to fit the whole dataset, and draws from that distribution to fill in missing values. This procedure is repeated many time, generating multiple complete datasets, to replicate the variances and covariances that exist in the multivariate distribution. Common approaches to multiple imputation include drawing from conditional models for each individual feature (van Buuren and Groothuis-Oudshoorn 2011) and drawing from a multivariate Gaussian (Honaker et al. 2011; Lee and Carlin 2010). Further discussion on data imputation is provided in the Background Section. Although considered to be a data preprocessing step, multiple imputation can have a profound effect on the final results of a model. Existing work (Kropko et al. 2014) has examined the effect of an imputation approach on the accuracy of missing value estimates, analytic model parameters, and prediction probabilities.

Data imputation techniques have primarily been studied by statisticians, who are interested in overall properties of a distribution (so that one can do things like significance testing). Multiple imputation is considered to be the best method primarily because it preserves distribution properties, even if individual imputed values are inaccurate. However, in classification settings, where the objective is the accuracy of predictions, the accuracy of individual values may be more important than the accuracy of the distribution of feature values. There is some literature on the effect of imputation on classification accuracy (Farhangfar, Kurgan, and Dy 2008; Acuna and Rodriguez 2004). Many of these works only considered single imputation methods (such as **Mean** and **Similar**). Among other results, these works show that **Similar** works well when features are not highly correlated (Batista and Monard 2003). Farhangfar *et al.* also consider multiple imputation, and show that across classifiers, there is no single best imputation method: results are highly inconsistent (Farhangfar, Kurgan, and Dy 2008). In classification settings, particularly with complex non-linear classifiers, the properties of the datasets and the relationships between features determine which imputation method works best. In the ideal case, a data imputation will strengthen differences between classes; in the worst case, these differences become blurred.

No study as yet examined the effect of imputation on classification fairness. By far the closest (and only similar) work to ours is that of Martinez-Plumed *et al.*, who study fairness and missing values (Martínez-Plumed et al. 2019). However, their work focuses primarily on analyzing fairness with respect to which values are missing, and while they explore data imputation and fairness, they only examine the simple data imputation method of filling in by the column mean.

Background: Data Imputation

Here, we give a brief overview of data imputation. Data imputation refers to the process of completing missing values in a dataset, and has its roots in statistics, rather than machine learning. Missing data may be present in the training data, and thus must be completed before a classifier can be trained, or may be present in the test data, and so must be completed before predictions can be made (or both). Naturally, the choice of data imputation technique affects the classification outcomes, and so it is important to study the effect of various imputation techniques, both from the perspective of accuracy as well as fairness.

Data Imputation Methods

In our work, we consider the following three data imputation techniques, which are the three most commonly used in practice (Gelman and Hill 2006).

Mean Imputation is the simplest method of data imputation. In this method, a missing value is filled in with the average value from the column. This method is computationally very fast, and although naive, it can sometimes give good results in classification tasks (Farhangfar, Kurgan, and Dy 2008). It does not preserve feature value distributions or relationships between features: correlations can be either increased or decreased.

Similar Imputation finds the k most similar data samples in the dataset, and fills in the missing value by the average of those samples. The method in this research uses the *KNNImputer* implemented by *Scikit-Learn*. Similarity between two data samples is computed using features that are known for both samples. The *Scikit-Learn* implementation uses k-Nearest Neighbors to find the most similar samples. In this work, we set $k = 5$.

Multiple Imputation produces multiple complete datasets by applying a multivariate imputer to the missing values multiple times. The imputer used in this research is the *IterativeImputer* from Scikit-Learn. The idea is based on the Multivariate Imputation by Chained Equations (MICE) algorithm (van Buuren and Groothuis-Oudshoorn 2011). MICE is based on the idea that to model the joint distribution of an entire dataset, we need only model the distribution of each feature conditional on all of the other features in the data. Then, based on the type of feature, a predetermined model is used to model each feature. Logistic regression is used for binary features, ordered logistic is used for ordered-categorical features, and so on, where each model includes all other features in the linear predictor. The process is iterative: first starting values are chosen for the missing values in the data. Then the conditional models are trained on these complete data, and new imputed values are drawn out-of-sample for each feature's conditional model. These new imputed values are used to retrain the conditional models, and the process is repeated until convergence in the imputed values is observed. Once the process converges, imputed values are drawn with randomness from the conditional models several times, generating many complete datasets which together express the variance due to uncertainty in the imputation model. To use this method in our study, we produce multiple complete datasets, perform classification on each, and report the majority of the resulting predictions.

Simulating Missing Data

When researchers study data imputation, they generally simulate missing data: this allows for accuracy-based comparisons, as the real values are known. The three most common ways of simulating missing data are the (confusingly-named) *Missing Completely At Random (MCAR)*, *Missing At Random (MAR)*, and *Not Missing At Random (NMAR)*.

The primary focus of this paper is on MCAR missingness, which is the simplest model, making no assumptions about relationships or dependencies between features. The other missingness models are also interesting, and will be the focus of future work.

In the ***Missing Completely At Random (MCAR)*** scenario, values are missing at random without dependencies on other features. The missingness probability can vary by feature, though we do not do that in this work. The MCAR assumption is somewhat artificial (it might occur if, say, a surveyor randomly skips questions, or a elements in a database are randomly corrupted), but it is also the most amenable to analysis and unlike the other two methods, makes no assumptions about dependencies between variables. As this is the first work to examine the effect of data imputation on classification, we choose to begin with the most straightforward missingness scenario. Given an expected missing ratio for the input dataset, we compute a uniform distribution of missing values across the columns and rows of the dataset, so that the percentage of missing values across rows and columns are approximately the same.

In the ***Missing At Random (MAR)*** scenario, the probability that a value is missing is a function of other feature values (Kropko et al. 2014). One example of this would be if people in a survey have some probability of refusing to reveal their income, but older individuals are more likely to refuse than younger people. In this example, the probability that the 'income' feature is missing is dependent on the 'age' feature.

The ***Not Missing At Random (NMAR)*** scenario is similar to *Missing at Random*, except the probability that a value is missing is dependent on that value itself or on the missing values of other features (Kropko et al. 2014). For example, using the income example from earlier, if low-income individuals are less likely to reveal their income than high-income individuals, then this value is *Not Missing At Random*.

In our simulations, we begin with a dataset with no missing values and then delete a selected set of values. We simulate MCAR missingness, and select values to delete uniformly at random according to a fixed probability.

Assumptions behind Data Imputation

In real-world settings, it can be difficult to use data imputation algorithms in a principled way. The alternative approaches to imputation are generally predicated on an atheoretical model of the joint distribution of the dataset: either by making the infeasible assumption that a real-world dataset is generated by the multivariate Gaussian distribution, or using naive conditional distributions. Furthermore, all of these approaches require either the MCAR or MAR assumption to guarantee that the imputed values do not bias the results of a final analytical model. There is no test, however, of the validity of the MAR assumption (Potthoff et al. 2006), so the argument that the data conform to an MAR missingness process must be made using theory and subject matter knowledge.

Experimental Overview

In this section, we discuss our experimental setup, including the machine learning algorithms used, a description of the datasets and pre-processing steps, the bias measurement used to evaluate fairness in classification, and other details of our study.

Machine Learning Algorithms

We evaluate six popular machine learning algorithms:

k-Nearest Neighbors (kNN) is a classic machine learning algorithm. In this algorithm, each unlabeled data point is assigned a label by taking a majority vote over the k nearest labeled data points (Cover and Hart 1967). k is defined by the user, or can be identified through a parameter search process. We set $k = 5$.

The **Support Vector Machine (SVM)** algorithm maps the data into a high-dimensional space, and computes support vectors to identify the boundaries between different classes in that space. The implementation we use in this research is the linear SVM with hinge loss, solved using the LibLinear library (Fan et al. 2008)), which converges quickly on large datasets.

Logistic Regression (LR) uses a statistical approach to compute the probability of a label for a given data point. The original algorithm can only produce binary classifications, since its output is between 0 and 1. However, it can also be adapted to multi-label classification by using the one-to-rest method (Hosmer Jr, Lemeshow, and Sturdivant 2013). We use the *Scikit-Learn* implementation, which by default applies regularization.

Decision Tree (Tree) generates a tree-like internal structure that directs each sample to a predicted label. The leaves at the end of the tree are the target labels, and the internal nodes are the conditions that decides which branch the prediction will follow (Safavian and Landgrebe 1991).

Random Forest (Forest) uses ensembles of multiple decision trees, and produces an output from the the average of outputs from the decision trees. By ensembling different random decision trees, Random Forest better avoids overfitting (Liaw, Wiener et al. 2002).

Multi-layer Perceptron (MLP) is a neural network approach to classification problems. The model creates a neural network that has multiple layers of perceptrons: one input layer, one output layer, and several hidden layers. Each perceptron is a binary function that controls data flow through the network. The model will be trained with an optimizer that adjusts the weight on each node of the network, in order to best fit the input data.

Datasets

We use six datasets that are commonly used in algorithmic fairness research. Each dataset has a target value that the classifier aims to predict, and we additionally denote one feature as the 'protected' feature for each. For each dataset, we removed irrelevant features (like name), as well as those features with extremely high missingness rates in the original dataset. A summary of each dataset is shown in Table 1. This table shows the target variable, protected attribute (which is part of the training feature set), and the remaining features.

The **Adult** dataset is obtained from the UCI Machine Learning Repository (Dua and Graff 2017). It contains information for 32,561 people, and has a target feature of whether the person earns more than $50,000 per year, where 1 means ">$50K" and 0 means "≤$50K".

The **COMPAS** dataset is the well-known crime recidivism dataset used in ProPublica's analysis. We obtained a copy of this dataset from ProPublica's Github page (Larson et al. 2016). The file that we use in our analysis is titled 'compas-scores.csv'. This dataset contains information from 11,757 defendants from Broward County, Florida. The target value is `is_recid`, which indicates whether the individual recidivates. As in ProPublica's analysis, we use `race` as the protected feature, and for simplicity, only consider 2 groups: 'Caucasian' and 'African-American'.

The **Titanic** dataset records survivals and deaths from the *Titanic* maritime disaster. It describes basic information of the people on board the ship, and the target feature is whether or not a person survived. The data we are using is downloaded from a Kaggle.com Contest (Titanic: Machine Learning from Disaster), and we use the file name 'train.csv', describing 891 people. We use `Sex` as the protected feature (which in this dataset is binary).

Next, we use the **German Credit** dataset from the UCI Machine Learning Repository (Dua and Graff 2017). The dataset contains information of 1000 individuals, and has a target of whether the individual is a good or bad credit risk. We use the `Age` attribute as the protected attribute, and divide it into two groups: those with age ≥ 26 ('older') and those with age < 26 ('younger').

The **Communities and Crime** dataset is also retrieved from the UCI Machine Learning Repository (Dua and Graff 2017). Unlike our other datasets, each row here represents a community. We compute a categorical target feature of 'More Violent Crimes' or 'Fewer Violent Crimes' using the dataset feature `ViolentCrimesPerPop`, where communities with an above average number of per capita violent crimes belong to the first category, and the rest belong to the second category. We then identify the majority racial group of each community, and use this as the protected attribute. As with the **COMPAS** dataset, we only consider 'Caucasian' and 'African-American'.

Finally, we use the **Bank Marketing** dataset, which is also retrieved from the UCI Machine Learning Repository (Dua and Graff 2017). We use the data in the 'bank-full.csv' file, which contains information of 45,211 people collected by a Portuguese banking institution, and has a target of whether the client will subscribe a term deposit. As with the **German Credit** dataset, we convert the `age` attribute to categorical by defining those with `age` ≥ 35 as 'older', and the rest as 'younger'. (For both this dataset and the earlier dataset, we used a threshold that resulted in roughly-balanced groups.)

Measuring Bias

We use a bias definition based on the Equal Opportunity fairness concept (Hardt, Price, and Srebro 2016b). For a binary classifier, and a protected attribute A of 2 groups, equal opportunity exists if $\Pr\{\hat{Y} = 1|A = 0, Y = 1\} = \Pr\{\hat{Y} = 1|A = 1, Y = 1\}$. In other words, equal opportunity is fulfilled when the TPR (True Positive Rate) for 2 groups are equal. Based on this notion, noting that $TPR = 1 - FNR$ and $TNR = 1 - FPR$, we define our bias measurement as:

$$\text{Bias} = |\text{FPR}_A - \text{FPR}_B| + |\text{FNR}_A - \text{FNR}_B|$$

where A and B are 2 groups of the protected attributes, FNR is False Negative Rate, FPR is False Positive Rate. FNR and FPR are defined as:

$$\text{FPR}_A = \frac{\text{FP}_A}{\text{FP}_A + \text{TN}_A} = 1 - \text{TNR}_A$$

$$\text{FNR}_A = \frac{\text{FN}_A}{\text{FN}_A + \text{TP}_A} = 1 - \text{TPR}_A$$

For example, in the context of the *COMPAS* dataset, FPR corresponds to the case where a defendant is mistakenly predicted as recidivating, and FNR corresponds to defendants who recidivated, but were not predicted to do so. We measure the absolute differences of these two metrics on the two protected groups as an indication of bias in classification.

Adult	**COMPAS**	**Titanic**	**German**	**Communities**	**Bank**
Dataset Size					
30,718	9388	712	1000	1993	45,211
Target Feature					
Whether a person makes over 50K a year	Whether a person recidivates	Whether a person survived	Whether a person has good credit risk	Whether the total number of violent crimes per 100K population in community is above average	Whether a client will subscribe a term deposit
Target Class Sizes					
0: 23,068 1: 7650	0: 6201 1: 3187	0: 424 1: 288	0: 700 1: 300	0: 1285 1: 708	0: 39,922 1: 5289
Protected Feature					
Sex	Race	Sex	Age	Race	Age
Protected Class Sizes					
Male: 20,788 **Female**: 9930	**African-American**: 4672 **Caucasian**: 3253 Hispanic: 817 Other: 571 Asian: 48 Native American: 27	**Male**: 453 **Female**: 259	**Older**: 810 **Younger**: 190	**Caucasian**: 1572 **African-American**: 218 Hispanic: 115 Asian: 88	**Older**: 30,198 **Younger**: 15,013
Other Features					
age workclass education education-num marital-status occupation relationship race hours-per-week	age age_cat (Age Category) c_charge_degree priors_count juv_misd_count (Juvenile Offenses Count) juv_fel_count (Juvenile Offenses Count) juv_other_count (Juvenile Offenses Count) days_b_screening_arrest sex length_of_stay	Pclass (Ticket Class) Age SibSp (Number of Siblings/Spouses Aboard) Parch (Number of Parents/Children Aboard) Fare Embarked	Status_account Duration_month Credit_history Purpose Credit_amount Savings_account Employment_since Installment_rate Personal_status Debtors_guarantors Residence_since Property Installment_plans Housing Number_credits Job Num_liable_people Telephone Foreign	Population in Community Household Age Population in Urban Areas Median Income Race Percentage Race-based Income Poverty Level Education Employment Status Marital Status Immigrants Speaking Language Birth Location Land Area Population Density Public Transit Usage	job marital education default (Has Credit in Default) balance housing loan contact (Communication Type) day month duration campaign (Number of Contacts in Campaign) pdays (Number of Days Passed) previous (Number of Contacts before Campaign) poutcome (Outcome of Previous Campaign)

Table 1: Dataset Details

Simulating Missingness in Data

To simulate MCAR missingness, we first fix a target missingness fraction p, and select p fraction of the elements from the dataset uniformly at random for deletion. In our experimental setup, we tested missing fractions from 0.05 to 0.95 in 0.05 increments.

Cross-Validation Details

We use 10-fold cross-validation in our experiments. For each 9:1 combination, we use 9 folds as the training data and the remaining 1 fold as the test data. We first run the selected imputation method on the full dataset (with the exception of multiple imputation, as described below) to fill in the missing values, and apply One-Hot encoding to expand categorical features to binary states. Next, we apply SMOTE (Chawla et al. 2002) on the training data to achieve class balance (with respect to the target variable). Before training the classifiers, we apply a StandardScaler on the training data as well as test data to standardize the mean and scale to unit variance.

Then for each classifier, we fit a model on the training data, run a prediction on the test data, and compute a confusion matrix for each protected group From the confusion matrix values, we compute bias as described earlier, as well as the accuracy of the classification. The final values for bias and accuracy are computed by taking the average of the 10-fold outputs.

For multiple imputation, because the output from the data imputation method is multiple complete datasets (10 in our implementation), we used majority voting to combine the results. Training data and test data are imputed separately, and we get 10 versions of each. Then for each type of classifier, we train 10 classifiers on the training data, and run prediction on each test data, and get 100 predictions. To combine predictions, we take the majority predicted value for each sample as the final prediction. The rest of the overall process is the same.

Additional Details of Experimental Setup

In our experiments, missingness is induced using the MCAR method, and we assess the bias and accuracy of data imputation methods.

In order to find the best parameter for each classifier-dataset combination, we use GridSearchCV from the `sklearn` package to search parameters in a defined parameter space. All preexisting missing values are dropped prior to parameter search. Detailed parameter settings can be found in appendices.

In this research, we use `Python3` as programming language for all experiments. We mainly use `Scikit-Learn` for machine learning algorithms, `imbalanced-learn` for SMOTE, `NumPy` and `Pandas` for dataset processing, and `matplotlib` for generating plots. Source code and results for all experiments can be found at https://anonymous.4open.science/r/bf71f0c3-39d1-4af1-bc5e-97a3c48f3195/ .

Experimental Analysis: Unfairness vs. Accuracy

Here, we present results demonstrating the effect of missingness and data imputation on accuracy and unfairness in classification. To perform this analysis, we compute accuracy and bias in the results of six classification algorithms as missingness varies from 0% to 95%, under the MCAR scenario. In all of these experiments, the protected attribute was excluded from the feature set. It is known that the induction of missingness will typically damage the accuracy of a classification model, no matter what data imputation method is used, but beyond that, the relationship between imputation and accuracy is complicated (Farhangfar, Kurgan, and Dy 2008). We see that the relationship between imputation and bias is also complicated.

Results are shown in Figures 1- 3. Due to space constraints, results for **Logistic Regression** and **Decision Tree** are not shown, but **Logistic Regression** is very similar to **Linear SVM** and **Decision Tree** is very similar to **Random Forest**.

The discussion in this section is structured as follows: First, we discuss general patterns in the relationship between bias and imputation. Next, we discuss patterns across different classifier algorithms. Finally, we discuss behaviors of the specific imputation methods.

Bias vs. Missingness

We observe that dataset/classifier combinations fall into one of three categories:

(1) As seen in Figure 1, accuracy and bias may both decrease roughly monotonically as missingness increases. In such cases, by damaging the predictive properties of the data, one also damages its potential for bias: each individual row becomes less distinctive, making it harder to predict the target variable, but also making it less obviously a member of its protected group. In this category, both the target variable and protected attribute are associated with the other attributes, and as those other attributes become less accurate, classification accuracy and bias decrease accordingly.

(2) As seen in Figure 2, bias may be very low to start with. In such cases, bias may slightly vary (e.g., decrease, as in the case of k-**NN**, or even show a 'hill' shape, as with **SVM** and **LR**), but changes are extremely small and not meaningful. This 'hill' shape occurs very rarely.

(3) As seen in Figure 3, the bias may stay relatively flat. This occurs when the protected attribute is not associated with the other non-target attributes. For example, the dataset shown in Figure 3 is the *Titanic* dataset. The protected attribute is 'sex', which is independent of the other non-target attributes (which include things like ticket class, etc.). Interestingly, this attribute *is* strongly associated with the target variable (whether or not the individual survived the disaster), but the target variable is not used in the imputation process. In this case, imputing from known values has nothing to do with the protected attribute.

Performance of Classification Algorithms

On the whole, the only major pattern that we see for the different classifiers is that the linear algorithms (**Linear SVM**, **Logistic Regression**) show slightly more bias and less accuracy than the others. However, all classifiers can give results fitting the three categories described above: the bias-related behavior as missingness increases seems to be much more a function of dataset than classification algorithm.

On some datasets, such as *Titanic* (Figure 3), all classifiers perform almost identically: accuracy declines roughly linearly with missingness, and bias is relatively flat for **multiple** imputation, or flat with a sharp decline at the end for **mean** and **similar** imputation. In contrast, in the *Adult* dataset (Figure 1), although the classifiers are similar with respect to accuracy (with **Linear SVM** slightly worse than the others at high rates of missingness), the linear algorithms such as **Linear SVM** show larger bias overall, while methods whose decision boundaries are non-linear, such as k-**NN** and **Random Forest** show less bias. Interestingly, on the *Bank* dataset, shown in Figure 2, the accuracy of most methods is fairly stable until very high amounts of missingness, while **Linear SVM** shows a gradual decline.

As with the relationship between imputation and accuracy, the relationships here are too complex and dataset/feature-specific to draw deeper general conclusions.

Performance of Imputation Methods

In most cases, **Multiple** imputation is the clear winner with respect to bias (one exception here is on the *Titanic* dataset at unrealistically high amounts of missingness). **Mean** imputation is the worst, and **Similar** imputation is in between (sometimes these latter two are swapped).

Again, these results are fairly complicated to interpret, as they are very dataset, classifier, and feature dependent. With respect to accuracy, our results confirm those seen in earlier works: it is surprising, but already known, that in some cases the simple **Mean** imputation can give the best results; while in other cases, the more sophisticated **Multiple** imputation method gives the highest accuracy.

With respect to bias, though, there are two consistent results: **Mean** imputation generally performs poorly, and **Multiple** imputation performs comparatively well. Just as with accuracy, the reasons for these observations can be complex and many. Two possible explanations include:

(1) If classes are imbalanced, then **Mean** imputation may strengthen the patterns characterizing the majority class and weaken patterns of the minority class.[1] This increases the predictive power associated with the features of the minority class, resulting in an increase in the FPR and FNR of that class, and a decrease in the FPR and FNR of the majority class. This can result in an overall increase in bias. Note that this explanation assumes that the two classes have sufficiently different mean feature values, not just different distributions.

(2) **Multiple** imputation generates multiple imputed values for each missing values (and thus generates multiple imputed datasets). The goal of **Multiple** imputation is to generate values that match the statistical properties of the underlying distribution, not to generate individually-accurate single imputed values. Counterintuitively, this inaccuracy may help when it comes to bias: instead of strengthening the patterns associated with a protected group, because each individual prediction is deliberately inaccurate, those patterns are weakened.

The main conclusion that we can draw from our results is that **Multiple** imputation gives the lowest bias of the three imputation methods considered, but often does poorly with respect to accuracy.

Proposed Data Imputation Method: Impute-From-Opposite

Results from the previous section demonstrate that in certain datasets, bias is either very low or unaffected by data imputation (the second and third categories described in Section). However, in other datasets (the first category described earlier), bias is affected by imputation, and this raises the question: can we design a data imputation method that results in a greater decrease in bias, while maintaining high accuracy?

In this section, we propose a heuristic to modify existing data imputation methods with the goal of imputing data so as to minimize unfairness in the end classification while preserving accuracy. We refer to this method as **Impute From Opposite (IFO)**. **IFO** assumes that the protected attribute is known (even if it is not used in classification).

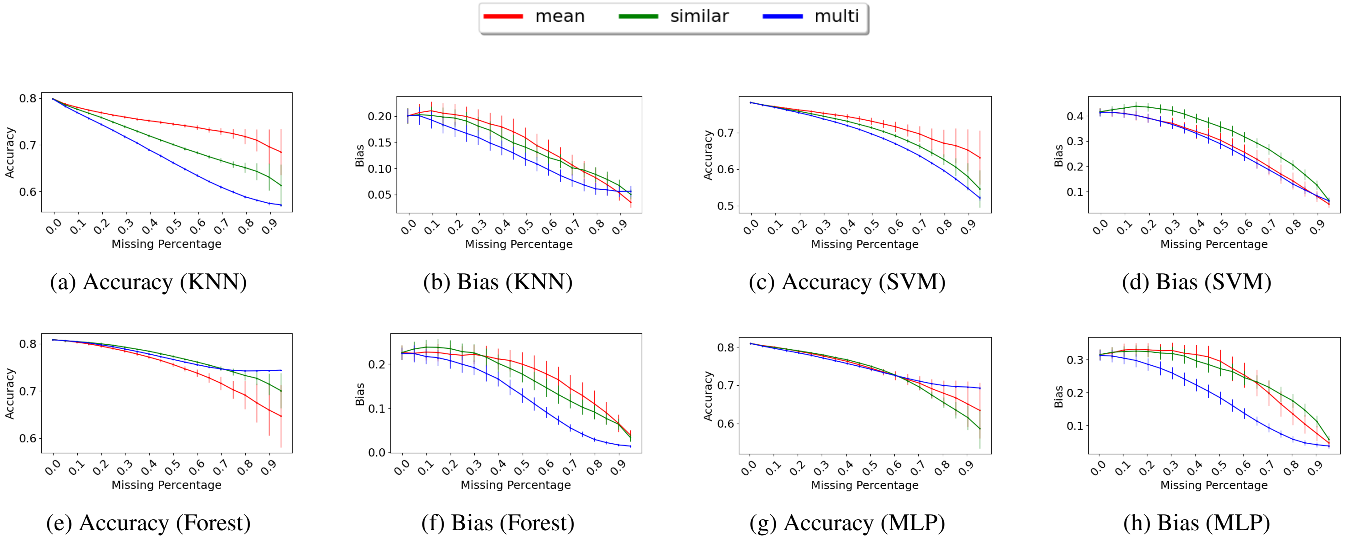
IFO can be explained simply: Instead of computing a missing value based on all the available data, compute it based on the data only from the opposite group(s). For example, in the *Adult* dataset, if a 'Female' data point has a missing value in the 'age' column, and we wish to use **Mean** Imputation, instead of computing the averaged age for the entire dataset, we only consider take the average over those data points where the 'sex' is 'Male' (i.e., the opposite group). In cases of a non-binary categorical attribute, one can instead use all protected groups other than the target row's. IFO works extremely well to reduce bias when amounts of missingness are low (under approximately 50%, which is realistic). For higher values of missingness, a standard method may be more appropriate. We discuss why this is later in this section.

Figures 4 and 5 show results for the various data imputation methods, both standard and with the IFO modification. We present results on the *Adult* and *COMPAS* datasets. These are the two datasets that exhibited behavior Results for other datasets are shown in the Appendix. The experimental setup is the same as used earlier.

As we can see from the figure, the accuracy of the IFO methods are generally the same as that of the original methods. (Though one could imagine that if some protected group were extremely small, it may be different for the opposite group to accurately impute from that very small protected group.)

When examining bias, we see that for fractions of missingness below 50%, use of each IFO method with **Similar** or **Multiple** imputation results in better (lower) bias scores

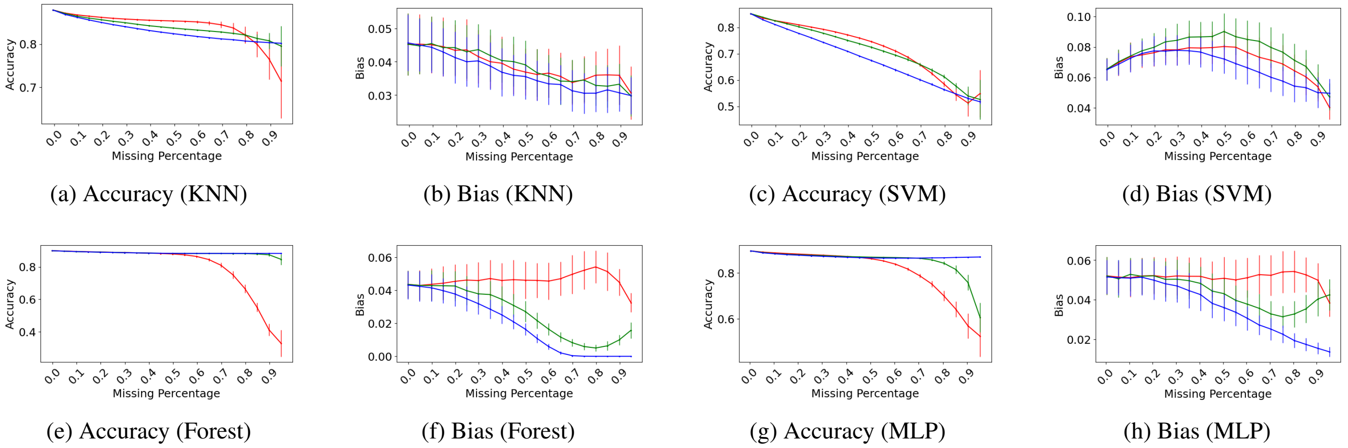
[1] Recall that we apply SMOTE to balance classes in the training data, but this is done before classification, not imputation.



(a) Accuracy (KNN) (b) Bias (KNN) (c) Accuracy (SVM) (d) Bias (SVM)

(e) Accuracy (Forest) (f) Bias (Forest) (g) Accuracy (MLP) (h) Bias (MLP)

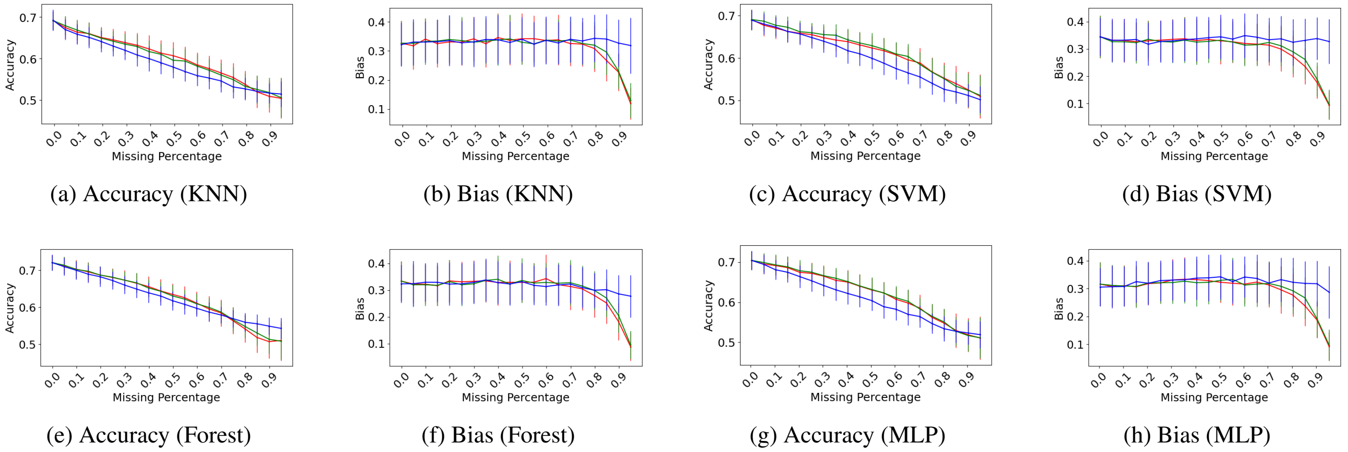
Figure 1: Results on Adult dataset.



(a) Accuracy (KNN) (b) Bias (KNN) (c) Accuracy (SVM) (d) Bias (SVM)

(e) Accuracy (Forest) (f) Bias (Forest) (g) Accuracy (MLP) (h) Bias (MLP)

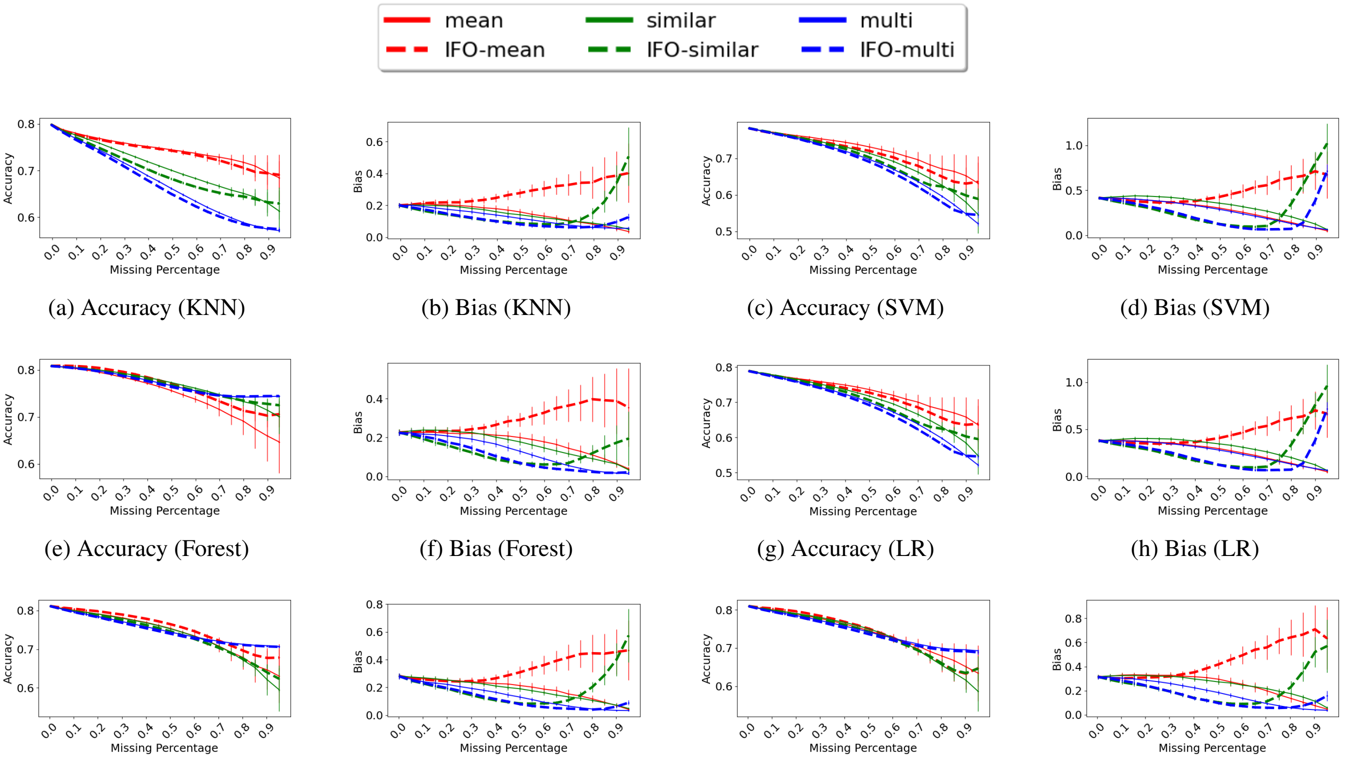
Figure 2: Results on Bank dataset.



(a) Accuracy (KNN) (b) Bias (KNN) (c) Accuracy (SVM) (d) Bias (SVM)

(e) Accuracy (Forest) (f) Bias (Forest) (g) Accuracy (MLP) (h) Bias (MLP)

Figure 3: Results on Titanic dataset.



(a) Accuracy (KNN)

(b) Bias (KNN)

(c) Accuracy (SVM)

(d) Bias (SVM)

(e) Accuracy (Forest)

(f) Bias (Forest)

(g) Accuracy (LR)

(h) Bias (LR)

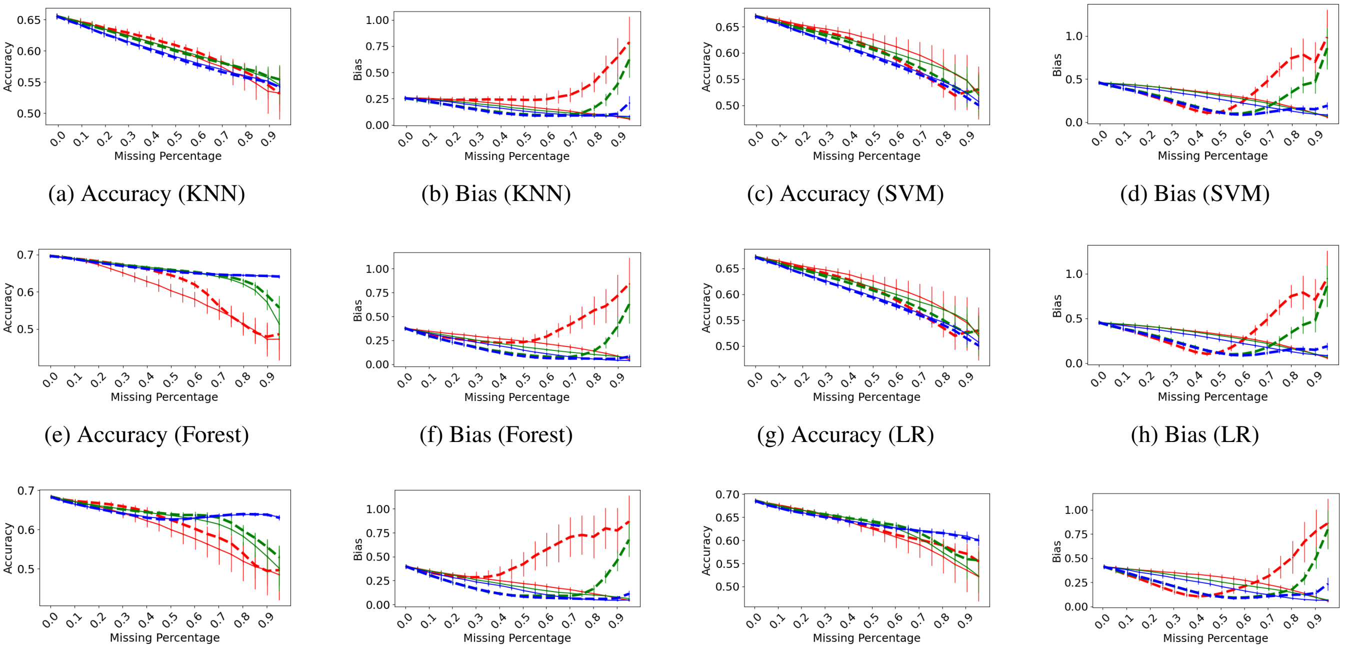
(i) Accuracy (Tree)

(j) Bias (Tree)

(k) Accuracy (MLP)

(l) Bias (MLP)

Figure 4: Adult Dataset (Combined Imputation Methods)



(a) Accuracy (KNN)

(b) Bias (KNN)

(c) Accuracy (SVM)

(d) Bias (SVM)

(e) Accuracy (Forest)

(f) Bias (Forest)

(g) Accuracy (LR)

(h) Bias (LR)

(i) Accuracy (Tree)

(j) Bias (Tree)

(k) Accuracy (MLP)

(l) Bias (MLP)

Figure 5: COMPAS Dataset (Combined Imputation Methods)

than use of the corresponding original method. In contrast, **IFO-Mean** generally does worse than **Mean** with respect to bias. There is an obvious explanation for this: **Mean** imputation imputes the same value for all missing elements in the same column, and similarly, **IFO-Mean** imputes the same value for all missing values in the same column from the same protected group. This gives an immediate proxy variable for the protected attribute, allowing additional bias to enter the system. **IFO-Mean** is thus always a poor strategy, since it introduces a protected attribute signal into the data even when none existed before, and should not be used.

The **IFO-Multiple** and **IFO-Similar** methods sometimes show an interesting 'U'-shaped curve: at around 50-70% missingness, bias will *increase*. This is very different than for standard methods, where the bias decreases as missingness increases. This is because for the standard methods, as missingness approaches 100%, the protected classes become identical, resulting in low accuracy and low bias. In contrast, the IFO methods exchange the characteristics of the two groups. The properties that lead to bias still exist, just with different labels. However, note that it would be unusual for a real dataset to have a missingness fraction high enough for this phenomenon to occur; and if so much data were missing, the dataset would be problematic for many other reasons, too. We also note that for values of missingness below roughly 50%, bias drops much faster as missingness increasxes for the IFO versions than the standard versions.

These results suggest that at realistic amounts of missingness, the IFO heuristic is an excellent alternative to standard methods of data imputation. There is no additional running time cost (and in fact, imputation is done over a smaller dataset), and accuracy is similar.

Discussion

Here, we list the key takeaway messages from our study.

Observation 1: As missingness increases, accuracy and bias decrease. It is well known that accuracy will decrease as missingness increases, and for the same reasons, it makes sense that bias does as well. As the information allowing the classifier to find patterns relevant to the target variable decreases, so too does that allowing the classifier to (inadvertently) find patterns correlated with the protected attribute.

Observation 2: Different machine learning algorithms show different performance with respect to accuracy and bias as missingness varies. Linear models such as **Logistic Regression** and **Linear SVM** have larger bias values, and accuracy reduces faster as missingness increases. In contrast, k**-Nearest Neighbors**, **Random Forest**, and **Decision Tree** show lower bias overall. **Observation 3: Of standard imputation algorithms, results for accuracy vs. missingness are mixed, but Multiple imputation generally shows the lowest bias as missingness increases.** In many cases, the simple **Mean** imputation method shows the highest accuracy overall (as is consistent with prior literature), but in other cases, **Multiple** imputation does best. Consistently, though, **Multiple** imputation shows dramatically lower bias than **Mean** and **Similar** imputation. This may be due to class imbalance issues, or may be because **Multiple** imputation generates a set of imputed values for each missing value.

Observation 4: The Impute-From-Opposite modification preserves accuracy while providing a sharp decrease in bias for values of missingness below 50% for Similar and Multiple imputation. With **IFO-Multiple** and **IFO-Similar** imputation, by using only opposite group(s) to impute missing values, one reduces the tendency of standard methods to reinforce 'stereotypical' features associated with a group, but still allows for a high degree of accuracy. With **Mean** imputation, however, because all missing values in the same column from the same protected group are imputed with the same value, **IFO-Mean** introduces a very strong proxy attribute into the data, even when none existed before. **IFO-Mean** thus should not be used. However, even for **IFO-Multiple** and **IFO-Similar**, for (unrealistically) large amounts of missingness, we see a 'U'-shaped bias pattern. This is because when so much data is missing, imputing from the opposite group makes an individual start to resemble that other group, thus allowing bias to re-enter.

Observation 5: For the Random Forest, Decision Tree, and MLP algorithms, when missingness is induced by MCAR, IFO-Multiple shows the best performance with respect to accuracy and bias. Across all classifiers, IFO-Similar provides a good tradeoff between accuracy and bias. For example, in Figure 4, **IFO-Multiple** Imputation (blue dashed lines) has the lowest bias for all classifiers listed, while maintaining a reasonable amount of accuracy as missingness increases. We can also observe that the bias under **IFO-Multiple** Imputation decreases much faster than other imputation methods. **IFO-Similar** performs moderately well with respect to both bias and accuracy.

Observation 6: The experimental results showed in this paper suggest that, in order to reduce bias, it may possible to proactively induce missingness in a complete dataset and then apply imputation methods. There are a number of issues that must be examined, including how such an approach would work with other methods for fair machine learning, and we plan to explore this in future work.

Conclusion and Future Work

In this work, we have examined the effect of data imputation on classification fairness. Although previous work has studied data imputation and classification accuracy, this is the first to explore data imputation in the context of bias with respect to protected groups. We find that the relationship between data imputation and fairness is difficult to generally characterize, but that **Multiple** imputation generally gives lower bias scores than **Mean** and **Similar** imputation. We proposed a novel heuristic, **Impute From Opposite**, and show that it gives similar accuracy but lower bias than its counterparts. Interestingly, our work suggests that a potential approach to mitigating unfairness in data/classification is to proactively induce missingness and then impute values. and we plan to explore this in future work.

References

Acuna, E.; and Rodriguez, C. 2004. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, 639–647. Springer.

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine Bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Batista, G.; and Monard, M. C. 2003. A study of k-nearest neighbour as an imputation method. In *In HIS*. Citeseer.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: Synthetic Minority over-Sampling Technique. *J. Artif. Int. Res.* 16(1): 321–357. ISSN 1076-9757.

Cover, T.; and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13(1): 21–27.

De Leeuw, E. D. 2001. Reducing missing data in surveys: An overview of methods. *Quality and Quantity* 35(2): 147–160.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9: 1871–1874.

Farhangfar, A.; Kurgan, L.; and Dy, J. 2008. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition* 41(12): 3692–3705.

Gelman, A.; and Hill, J. 2006. *Missing-data imputation*, 529–544. Analytical Methods for Social Research. Cambridge University Press. doi:10.1017/CBO9780511790942.031.

Hardt, M.; Price, E.; and Srebro, N. 2016a. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413* .

Hardt, M.; Price, E.; and Srebro, N. 2016b. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, 3315–3323.

Honaker, J.; King, G.; Blackwell, M.; et al. 2011. Amelia II: A program for missing data. *Journal of statistical software* 45(7): 1–47.

Hosmer Jr, D. W.; Lemeshow, S.; and Sturdivant, R. X. 2013. *Applied logistic regression*, volume 398. John Wiley & Sons.

Kropko, J.; Goodrich, B.; Gelman, A.; and Hill, J. 2014. Multiple imputation for continuous and categorical data: Comparing joint multivariate normal and conditional approaches. *Political Analysis* 497–519.

Larson, J.; and Angwin, J. 2016. Technical Response to Northpointe. https://www.propublica.org/article/technical-response-to-northpointe.

Larson, J.; Mattu, S.; Kirchner, L.; and Angwin, J. 2016. How We Analyzed the COMPAS Recidivism Algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

Lee, K. J.; and Carlin, J. B. 2010. Multiple Imputation for Missing Data: Fully Conditional Specification Versus Multivariate Normal Imputation. *American Journal of Epidemiology* 171: 624–632.

Liaw, A.; Wiener, M.; et al. 2002. Classification and regression by randomForest. *R news* 2(3): 18–22.

Martínez-Plumed, F.; Ferri, C.; Nieves, D.; and Hernández-Orallo, J. 2019. Fairness and Missing Values. *CoRR* abs/1905.12728. URL http://arxiv.org/abs/1905.12728.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2019. A Survey on Bias and Fairness in Machine Learning. *CoRR* .

Potthoff, R. F.; Tudor, G. E.; Pieper, K. S.; and Hasselblad, V. 2006. Can one assess whether missing data are missing at random in medical studies? *Statistical methods in medical research* 15(3): 213–234.

Safavian, S. R.; and Landgrebe, D. 1991. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics* 21(3): 660–674.

van Buuren, S.; and Groothuis-Oudshoorn, K. 2011. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software* 45(3): 1–67. URL https://www.jstatsoft.org/v45/i03/.