Fair Forests: Regularized Tree Induction to Minimize Model Bias

Edward Raff
Booz Allen Hamilton
Raff_Edward@bah.com

Jared Sylvester
Booz Allen Hamilton
Sylvester@bah.com

Steven Mills
Booz Allen Hamilton
Mills_Steven@bah.com

ABSTRACT

The potential lack of fairness in the outputs of machine learning algorithms has recently gained attention both within the research community as well as in society more broadly. Surprisingly, there is no prior work developing tree-induction algorithms for building fair decision trees or fair random forests. These methods have widespread popularity as they are one of the few to be simultaneously interpretable, non-linear, and easy-to-use. In this paper we develop, to our knowledge, the first technique for the induction of fair decision trees.We show that our "Fair Forest" retains the benefits of the tree-based approach, while providing both greater accuracy and fairness than other alternatives, for both "group fairness" and "individual fairness." We also introduce new measures for fairness which are able to handle multinomial and continues attributes as well as regression problems, as opposed to binary attributes and labels only. Finally, we demonstrate a new, more robust evaluation procedure for algorithms that considers the dataset in its entirety rather than only a specific protected attribute.

CCS CONCEPTS

• **Computing methodologies → Classification and regression trees**; • **Social and professional topics** → *Codes of ethics*; *User characteristics*;

ACM Reference Format:
Edward Raff, Jared Sylvester, and Steven Mills. 2018. Fair Forests: Regularized Tree Induction to Minimize Model Bias. In *2018 AAAI/ACM Conference on AI, Ethics, and Society (AIES '18), February 2–3, 2018, New Orleans, LA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3278721.3278742

1 INTRODUCTION

As applications of Machine Learning becomes more pervasive in society, it is important to consider the fairness of such models. We consider a model to be fair with respect to some protected attribute a_p (such as age or gender), if it's predicted label $\hat{y}$ with respect to a datumn x is unaffected by changes to a_p. Removing a_p from x is not sufficient to meet this goal in practice, as a_p's effect is still present as a latent variable [23]. In this work, we look at adapting decision trees, specifically Random Forests, to this problem. Given an attribute a_p that we wish to protect, we will show how to induce a "Fair Forest" that provides improved fairness and accuracy compared to existing approaches.

Decision Trees have become one of the most widely used classes of machine learning algorithms. In particular, C4.5 [24] and CART [5] tree induction approaches, combined with ensembling approaches like Random Forests [3] and Gradient Boosting [13], have proven to be potent and effective across a broad spectrum of needs and tasks. These methods are one of the few to be simultaneously interpretable, non-linear, and easy-to-use.

Random Forests have proven to be particularly effective. In a study of over one-hundred datasets, Random Forests were found to be one of the best performing approaches — even when no hyperparameter tuning is done [12]. XGBoost, a variant of gradient boosting, has been used in the winning solutions to over half of recent Kaggle competitions [8].

Tree-based algorithms also provide a rare degree of interpretability. Single trees within an ensemble can be printed in a human-readable form, allowing the immediate extraction of the decision process. Further still, there are numerous ways to extract feature importance scores from any tree-based approach [4, 21]. Being able to understand how a model reaches its decision is of special utility when we desire fair decision algorithms, as it gives us a method to double-check that the model appears to be making reasonable judgments. This interpretability has already been exploited in prior work to understand black-box models [15].

Given the wide-ranging benefits and successes of tree-based learning, it is surprising that no prior work has focused on designing fair decision tree induction methods. Other methods for constructing fair models will be reviewed in section 2. In section 3 we propose, to the best of our knowledge, the first fair decision tree induction method. Our design is simple, requiring only minimal changes to existing tree induction code, thereby retaining the desirable property that the trees tend to "just work" without hyperparameter tuning. Our experimental methodology is discussed in section 4, including the introduction of novel fairness measures which are suitable for use with multinomial and continuous attributes. Finally, experimental results are summarized in section 5, including a new experimental procedure to evaluate fair algorithms against all possible features rather than single protected attributed. We end with our conclusions in section 6.

2 RELATED WORK

One approach to building fair classifiers is based on data alteration, where the original corpus is altered to remove or mask information about the protected attribute. Some of the first work in fairness learning followed this approach, and attempted to make the minimum number of changes that removed the discriminative protective information [17]. Others have attempted to re-label the data points to ensure a fair determination [22].

Another approach is to regularize the model in such a way that it is penalized for keeping information that allows it to discriminate against the protected feature. Some of the earliest work was to

AIES '18, February 2–3, 2018, New Orleans, LA, USA

ACM ISBN 978-1-4503-6012-8/18/02…$15.00
https://doi.org/10.1145/3278721.3278742

develop a fair version of Naive Bayes algorithm [7]. Others have taken to creating a differentiable regularization term, and applying it to models such as Logistic and Linear Regression [1, 2, 6, 18]. Our new fair induction algorithm is a member of this group of regularization-based approaches, but unlike prior works has no parameters to tune.

One final group of related approaches is to build new representations, which mask the protected attribute [9]. The use of neural networks have become popular for this task, such as variational auto encoders [20] and adversarial networks [11]. One of the seminal works in this field used an autoencoder with three separate terms in the loss [27], and provides one of the largest comparisons on three now-standard datasets. We replicate their evaluation procedure in this work.

There is an important commonality in all of these prior works. The research is done with respect to datasets and attributes where there is a prior normative expectation of fairness. These are problems usually of social importance, and protected attributes are intrinsic characteristics like age, gender and nationality. But what if focusing on such problems has inadvertently biased the development of fair research? The mechanism for inducing fairness should work for any attribute, not just those that align with current societal norms, and must not be over-fit to the protected attributes used in research. We evaluate our approach with respect to every possible feature choice, to ensure that the mechanism of producing fairness is not over-fit to the data.

3 FAIR FORESTS

We propose a simple regularization approach to constructing a fair decision tree induction algorithm. This is done by altering the way we measure the information gain $G(T, a)$, where T is a set of training examples, and a is the attribute to split on. We will denote the set of points in each of the k branchs of the tree as $T_{i\ldots k}$. This normally is combined with an impurity measure $I(T)$, to give us

$$G(T, a) = I(T) - \sum_{\forall T_i \in \text{splits}(a)} \frac{|T_i|}{|T|} \cdot I(T_i) \tag{1}$$

The information gain scores the quality of a splitting attribute a by how much it reduces impurity compared to the current impurity. The larger the gain, the more pure the class labels have become, and thus, should improve classification performance. In the CART algorithm, the Gini impurity (2) is normally used for categorical targets.

$$I_{\text{Gini}}(T) = 1 - \sum_{\forall T_i \in \text{splits(label)}} \left(\frac{|T_i|}{|T|}\right)^2 \tag{2}$$

This greedy approach to feature selection has proven effective for decades, helping to cement the place of tree-based algorithms as one of the most popular learning methods. However, this does not take into account any notion of fairness, which we desire to add. In this work we do so by altering the information gain scoring itself, leaving the whole of the tree induction process unaltered.

We begin by noting we need to make two slight alterations for our approach. First, we will use the Impurity score to measure both the class label, and now additionally the protected attribute under consideration. We will denote these two cases as I^l, and I^a, and the Gain with respect to the label and protected attribute as G^l and G^a respectively. Additionally, we will impose the constraint that the impurity measure must return a value normalized to the range of $[0, 1]$. For the Gini measure this becomes

$$I^a_{\text{Gini}}(T) = \frac{1 - \sum_{\forall T_i \in \text{splits}(a)} \left(\frac{|T_i|}{|T|}\right)^2}{1 - |\text{splits}(a)|^{-1}} \tag{3}$$

We require that the impurity score $I^a(\cdot)$ produce a normalized score so that we can compare scores on a similar scale range, regardless of which features are selected. We then use this to define a new fair gain measure $G_{\text{fair}}(T, a)$, which seeks to balance predictive accuracy with the fairness goal with respect to some protected attribute a_f.

$$G_{\text{fair}}(T, b) = G^l(T, b) - G^{a_f}(T, b) \tag{4}$$

Intuitively, (4) will discourage the selection of any feature correlated with both the protected attribute and the target label. It remains possible for such a feature to still be selected if no other feature is better suited.

3.1 Gain for Numeric Features

To our knowledge, no work has yet explored making a continuous feature the protected attribute. We can derive this naturally in our new fair induction framework. In CART, trees' numeric target variables are optimized by finding the binary split that minimizes the weighted variance between each split. We use this same notion to define a gain $G_r(T, a)$ that is used when either the predictor or protected attribute is continuous.

Because we are interested in fairness, we look at changes in the mean value of the splits compared to their parent. Even if variances differ, if they retain similar means the impact on the fairness is minimal. To produce a scaled value, we look at the number of standard deviations from the previous mean is for each of the new splits, and assume that being more than three standard deviations is the maximum violation. This gain is defined in (5), where σ_{b,T_i} indicates the standard deviation of attribute b for all datums in the set T_i, and μ_{b,T_i} has the same meaning but for the mean of the subset.

$$G_r(T, b) = 1 - \frac{1}{3} \sum_{T_i \in \text{split}} \frac{|T_i|}{|T|} \min\left(\frac{|\mu_{b,T} - \mu_{b,T_i}|}{\sigma_{b,T}}, 3\right) \tag{5}$$

We emphasize that the standard deviation of the parent T is used, not that of any sub-population T_i. This is because we want to measure drift with respect to the current status. Re-writing the continuous splitting criteria in this fashion also produces a score normalized to the range $[0, 1]$. We can now continue to use the $G_{\text{fair}}(T, b)$ function with continuous attributes as either the label target, or the protected attribute.

This framework now gives us a means to induce decision trees, and thus build Random Forests, for all scenarios: classification and regression problems, and protected features either nominal or numeric. We emphasize that this approach to regularizing the information gain has no tunable parameters as given. This is to keep with the general utility of decision trees in that they often "just work."

While adjusting hyperparameters such as maximum tree depth may be used to improve classification accuracy, the results of a decision tree are often effective without any kind of parameter tunning. This is important for practical use and adoption. Many fairness based systems require an additional two to three hyperparameters to tune [1, 18, 27], on top of whatever hyperparameters come with the original model. This increases the computational requirements in practice, especially when used with a classic grid-search approach.

4 METHODOLOGY

There is currently considerable discussion about what it means for a machine learning model to be fair, which metrics should be used, and whether or not they can be completely optimized [14, 16, 26].

We choose to use the same evaluation procedure laid out by Zemel et al. [27]. This makes our results comparable with a larger body of work, as their approach and metrics have been widely used through the literature [1, 6, 10, 19]. We present both of their metrics — Discrimination and Inconsistency[1] — in a manner compatible with both classification and regression problems, while also extending Discrimination to a broader set of scenarios. We will also discuss the datasets used, their variants tested, and the models we will evaluate.

4.1 Metrics

The first metric we will consider is the Discrimination of the model, measured by the average difference between the average predicted scores for each attribute value.

$$\text{Discrimination} = \left| \frac{\sum_{x_i \in T_{a_p}} \hat{y}_i}{|T_{a_p}|} - \frac{\sum_{x_i \in T_{\neg a_p}} \hat{y}_i}{|T_{\neg a_p}|} \right| \tag{6}$$

Discrimination measures a macro-level quality of fairness, as such it is sometimes termed “group fairness.” However, the definition in (6) is limited to only binary protected attributes. For this work, we will also look at a generalization of Discrimination to k-way categorical variables. This is done by re-formulating Discrimination to consider the sub-population differences from the global mean. This is equivalent to the original definition when $k = 2$, and is given by (7). (See the Appendix for a proof of equivalence.)

$$\text{Discrimination} = \frac{2}{k} \sum_{i=1}^{k} \left| \frac{\sum_{x_j \in T} \hat{y}_j}{|T|} - \frac{\sum_{x_j \in T_i} \hat{y}_j}{|T_i|} \right| \tag{7}$$

We will also consider the discrimination with respect to a continuous variable. With a_p denoting a protected continuous attribute, let $x_i(a_p)$ be the value of feature a_p for datum x_i. We will then define our new Maximum Discrimination (MaxD) metric as the largest discrimination score achieved for some binary split of a_p by some threshold t. This is given in equation (8), and gives us a concise definition extending Discrimination to regression tasks. When a continuous attribute is manually discretized into a binary problem, as is done in prior work, we obtain by definition that MaxD ≥ Discrimination.

$$\text{MaxD} = \arg\max_{t} \left| \frac{\sum_{x_i(a_p) < t} \hat{y}_i}{|x_i(a_p) < t|} - \frac{\sum_{x_i(a_p) \geq t} \hat{y}_i}{|x_i(a_p) \geq t|} \right| \tag{8}$$

Given our novel extensions of the Discrimination scores (7) and (8), we can evaluate this property for any feature. Importantly though, these metrics are population level measures of fairness. Satisfying the Discrimination metric does not guarantee that no bias exists. To measure the potential for bias within sub-populations of the data set, we look at the Inconsistency metric (9).

$$\text{Inconsistency} = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{y}_i - \frac{1}{k} \sum_{j \in k\text{-NN}(x_i)} \hat{y}_j \right| \tag{9}$$

Inconsistency compares the prediction of the model with that of nearby points, and is sometimes referred to as “individual fairness.” This is under the assumption that nearby points should produce similar predictions, and is optimized when the score is as close to zero as possible.

Discrimination and Inconsistency are both evaluating the fairness of a model, and hence do not consider the true supervised label y. Maximizing fairness involves minimizing these two scores, at a potential cost to the model’s predictive utility. We measure the predictive utility of each model with accuracy or Root Mean Squared Error (RMSE) for classification and regression problems respectively. For classification problems, we also consider the *Delta* metric, where Delta = Accuracy − Discrimination.

For corpora with a test set, these metrics will all be evaluated on the given test set. Otherwise, we will evaluate these scores on 10-fold cross validation. For Inconsistency, we will measure it using nearest neighbors from all folds, but using the predicted scores obtained from cross validation. This is in keeping with prior work [27].

4.2 Data Sets

To evaluate our work, we will use three classification datasets used by Zemel et al. [27], the German Credit, Adult, and Heritage Health datasets. For regression we will also use the Health dataset, which was originally a regression problem (how many days will someone stay in the hospital?) that was converted to classification (will they stay one or more days?).

Table 1 summarizes the size, protected attribute, feature count, and task type for each dataset. For the German and Health datasets, the protected attribute *age* is originally encoded as a numeric feature, but, because prior work did not support continuous protected attributes, they converted it to a binary categorical feature. We replicate this in our work, but will also investigate using the original continuous version of *age*.

Table 1. **Summary of the datasets used.**

Dataset	Samples	Protected	Features	Task
German Credit	1,000	Age ≥ 25	20	Good/Bad Credit
Adult Income	45,222	Male/Female	14	Income ≥ 50k
Heritage Health	147,471	Age ≥ 65	149	Stay ≥ 1 day
Heritage HealthR	147,471	Age ≥ 65	149	Days in stay

[1] Zemel et al. refer to their metric as ‘consistency,’ but define it in a way that only makes sense for classification. We use Inconsistency = 1 − Consistency. This form is applicable to both classification and regression tasks.

Table 2. **For each classification task, we show Accuracy, Delta, Discrimination, and Inconsistency, in that order. Scores are for our new method and prior work. Best results shown in bold, second best in *italics.***

	German				Adult				Health			
	Acc	Delta	Discrim	Incon	Acc	Delta	Discrim	Incon	Acc	Delta	Discrim	Incon
DT	0.6890	0.6509	0.0381	0.2140	0.8364	0.4801	0.3563	0.4417	0.8404	0.8196	0.0207	0.2062
DT^F	*0.6990*	0.6908	0.0082	*0.0070*	0.7511	0.7444	0.0067	*0.0033*	**0.8474**	*0.8473*	*0.0001*	*0.0001*
RF	0.6970	*0.6911*	0.0059	0.0020	**0.8501**	0.5463	0.3038	0.3944	0.8472	0.8464	0.0007	0.0005
RF^F	**0.7000**	**0.7000**	**0.0**	**0.0**	0.7530	*0.7530*	**0.0**	**0.0**	**0.8474**	**0.8474**	**0.0**	**0.0**
NB^F	0.6888	0.6314	0.0574	0.3132	*0.7847*	**0.7711**	0.0136	0.4366	0.6878	0.5678	0.1200	0.4107
LR	0.6790	0.5517	0.1273	0.3050	0.6787	0.4895	0.1892	0.2703	0.7547	0.6482	0.1064	0.2767
LR^F	0.5953	0.5842	0.0111	0.1284	0.6758	0.6494	0.0264	0.2234	0.7212	0.7038	0.0174	0.3777
LFR	0.5909	0.5867	0.0042	0.0592	0.7023	0.7018	*0.0006*	0.1892	0.7365	0.7365	**0.0000**	**0.0000**

4.3 Models Evaluated

When listing results, we will compare with standard CART decision trees (DT) and Random Forests (RF). Our fair variants of these methods will be denoted as DT^F and RF^F.

Since our new fair tree induction can directly protect the original non-discretized form, we also evaluate in that manner. Models DT$_c^F$ and RF$_c^F$ indicate a fair decision tree and Random Forest trained to protect the continuous age attribute. When we do this, we will continue to evaluate the models' Discrimination with the originally proposed threshold.

From Zemel et al. [27], we compare against their proposed Learning Fair Representations (LFR) approach and their baseline approaches: Logistic Regression, fair Logistic Regression (LR^F) [18] and fair Naive Bayes (NB^F) [17].

5 EXPERIMENTS

In this section we present the results of our experiments. We remind the reader that for all experiments, we perform no parameter turning for any of our tree-based models. This is in line with practical use, and is a benefit for users in both runtime and simplicity. In these experiments we will show our Fair Forests can be used in the standard classification scenario with a binary protected attribute. In addition, we can use a continuous protected attribute and achieve similar results, and apply both methods to a numeric prediction target. All code was written in Java using the JSAT library [25].

5.1 Binary Target, Binary Protected

For the classification tasks, the results for our various decision tree variants can be seen compared to the baselines in Table 2. We can see that our new Fair Forests win in almost every metric.

Looking at just the tree-based results, we can make two interesting observations. First, that the ensembling and random feature sub-sampling used by Random Forests appears to improve the fairness of CART trees, both when they do and do not consider our fairness regularization. This is a positive indication for the general use of Random Forests compared to single decision trees. Second, that our fairness regularizer can actually *improve* accuracy. This was observed on the German and Health datasets. We do not observe this phenomena with any other fairness approach. While a positive result, we caution that this should not be a general expectation. It is always possible that the protected attribute may truly be predictive of the target task. In such cases we would expect performance to decrease, which we observe on the Adult income dataset.

This result is also important when we contrast to the Logistic Regression model and its fair variant. On every dataset tested, the fair variant of LR has worse predictive accuracy than the standard model. Our fair trees do not suffer in the same way, indicating they are a more robust approach to building fair models.

The baseline results shown from Zemel et al. [27] required a grid-search, and were selected to maximize the Delta score. In this regard our Fair Forests almost dominate the table. The Fair Forest is second best only once to NB^F on the Adult Income corpus, with a relative difference of merely 2.3%. NB^F achieves this by obtaining higher accuracies, but also a higher Discrimination.

On both measures of fairness, Discrimination and Inconsistency, our fair Random Forest dominates the table with empirical zeros. The best non-tree approach in this regard is the LFR algorithm, which obtains empirical zeros on the Health dataset and near-zeros for Discrimination on the German and Adult datasets. However, LFR's Inconsistency increases to 0.06 and 0.19 for each respectively.

5.2 Binary Target, Continuous Protected

In all prior literature we are aware of, the protected attribute is always presented as a binary feature. Our fair tree induction approach allows us to mark a continuous feature as protected directly, without first having to discretize it. We can test this ability with the German and Health datasets, where the protected attribute (*age*) is originally a numeric feature. The results comparing this approach with the classic binary *age* attribute are shown in Table 3. In this table Discrimination is based on the original thresholds used to binarize the *age*.

Here we can see that the Random Forest using the continuous *age* (RF$_c^F$) and the one using binary *age* (RF^F) have equivalent performance. This appears to be a net effect of the added fairness Random Forests naturally provide. In this case it becomes more informative to look at the results from the standard decision tree, where non-zero Discrimination still occurs.

For both DT^F and DT$_c^F$, we can see they continue to reduce the Discrimination and Inconsistency with respect to the original decision tree approach. In these cases, DT$_c^F$ appears to uniformly outperform DT^F in regards to the fairness metrics, with only a 0.003

Table 3. **Classification results on German and Health datasets, where the protected *age* attribute is left as a numeric feature, rather than being converted to a binary categorical one. Best results in bold, second best in *italics*.**

	German				Health			
	Acc	MaxD	Discrim	Incon	Acc	MaxD	Discrim	Incon
DT^F	*0.6990*	0.0216	0.0082	0.0070	**0.8474**	*0.0001*	*0.0001*	*0.0001*
DT$_c^F$	0.6960	*0.0054*	*0.0047*	*0.0040*	**0.8474**	**0.0**	**0.0**	**0.0**
RF^F	**0.7000**	**0.0**	**0.0**	**0.0**	**0.8474**	**0.0**	**0.0**	**0.0**
RF$_c^F$	**0.7000**	**0.0**	**0.0**	**0.0**	**0.8474**	**0.0**	**0.0**	**0.0**

change in accuracy. On the German dataset we can also see that DT$_c^F$ has improved upon the MaxD score from the DT^F, dropping from 0.0216 down to 0.0054. This is reasonable to expect, as DT^F is optimizing fairness with respect to a specific value of *age*, where DT$_c^F$ is attempting to be fair with respect to all *age* values.

Further reading of the table indicates the MaxD score for DT$_c^F$ (0.005) is smaller than the Discrimination score of the DT^F approach (0.008). This means DT$_c^F$ has a greater degree of fairness with respect to age for all possible age splits, than DT^F does with regard to its single age split of interest. We explain this result by noting that DT^F's single split focus at $age \geq 25$ means discrimination can occur in nearby age ranges (e.g., 26-30, or 21-24), and this permissible "border" discrimination can generalize into the test set. Ultimately, while protecting the binary *age* attribute works well compared to the naive DT in Table 2, these results demonstrate the benefit of protecting the original numeric attribute: we can provide better fairness with respect to the threshold of interest, as well as every possible other threshold.

5.3 Continuous Target

One of the benefits of Decision Trees or Random Forests is that they can be applied to both classification and regression problems. In this section we will show that our fair induction strategy improves fairness in such scenarios, both when protecting on a continuous or a binary attribute. We do this using the original version of the Health dataset (see Table 4).

Table 4. **Results using the Regression version of the Health dataset, where Mean Squared Error is the target metric. Best results shown in bold, second best in *italics*.**

	MSE	Discrim	MaxD	Incon
LR	2.722	0.0011	0.0024	0.4163
DT	2.904	0.0006	0.0057	1.0285
DT^F	**2.662**	*0.0005*	**0.0005**	0.0964
DT$_c^F$	*2.663*	**0.0003**	*0.0007*	**0.0695**
RF	2.735	0.0021	0.0022	1.0464
RF^F	2.664	0.0006	0.0006	0.1366
RF$_c^F$	2.664	0.0007	0.0010	*0.0690*

When phrased as a regression problem, we see lower Discrimination scores for both the standard Decision Tree and the Random Forest, which leaves little room for improvement. When comparing Discrimination against the binary *age* threshold ($age \geq 65$), and the Maximum Discrimination against *age*, we see the fair variants of our algorithms perform better than their non-fair counterparts. While the DT^F and DT$_c^F$ happen to perform slightly better than their counterparts RF^F and RF$_c^F$, the differences are in an epsilon range. Either way, these results show that we can use our approach for regression problems and protect both categorical and continuous attributes.

5.4 Visualizing the Impact of Fairness

One of the benefits of tree-based approaches to prediction is the ability to interpret the models. In particular, we note that one can measure the relative importance of a feature using a variety of approaches. Using the Mean Decrease in Impurity (MDI) measure [21], we show the relative importance of features on the German and Adult datasets. These are shown in Figure 1 and Figure 2 respectively, where "Fair" is the relative importance of features used by our Fair Forest induction algorithm and "Standard" indicates the normal Random Forest induction process using CART-style trees. These results allow us to see that our simple regularizer can have a wide range of impact, depending on the dataset and protected attribute.

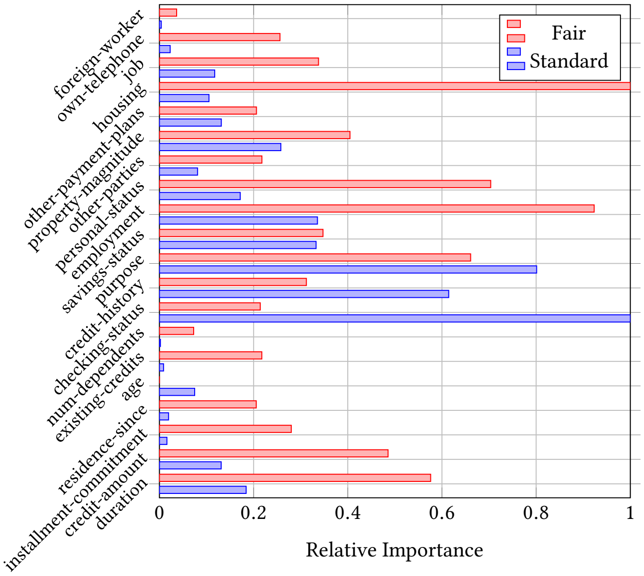

Figure 1. **Feature importance from German dataset.**

On the German dataset in Figure 1, we see a dramatic change in what the model considers importance, with the the most important variable being *checking-status* under the Standard model but *housing* under the Fair model. For almost all features in this corpus, we see a reversing of importance: if it was important under the naive model, it becomes less-so under the Fair model, and vice versa. The only exception to this being the *savings-status* attribute, and to some degree, *property-magnitude*.

The Adult dataset has a markedly different and surprising behavior. Under both the Fair and Naive model, the *relationship* attribute continues to be the most important. However, the Fair model dramatically reduces the relative importance of most other features. Many of these (e.g. *capital-loss*, *capital-gain*, *education*) would likely

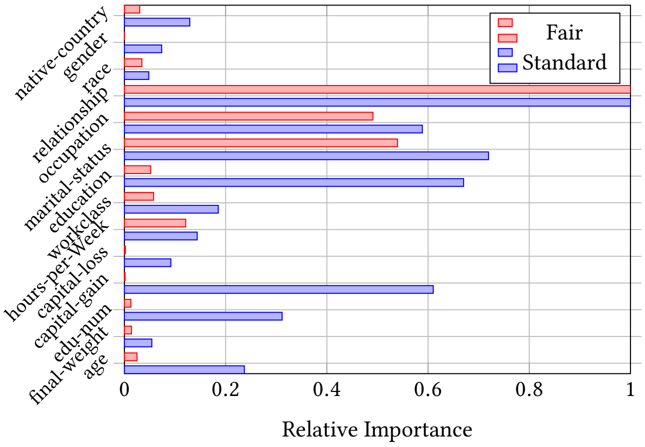

Figure 2. **Feature importance from Adult dataset.**

be features we expect to reliably predict the target attribute, Income. While our intuition may be that these variables should be unbiased and naturally fair predictors, the underlying distribution of *this dataset* indicates they were too highly correlated with the protected Gender attribute, and thus were rarely selected for use.

We expect that the ability to perform such investigation into feature importance pre/post fairness will become a valuable tool for those who wish to build fair models in production environments. Changes in feature importance can give us underlying insights into non-linear correlations that would escape simple analysis. The information itself may allow a decision maker to discover deficiencies or unintended biases in their data collection process, based on these unexpected changes. For example, the non-use of the *capital-gain/loss* features may tell us that we need to collect more data specifically from women with capital investments.

5.5 Fairness vs the Mechanism

We now evaluate the ability of our model to reduce Discrimination for every attribute individually, across each dataset. This helps us to determine that our approach is not overly specific to the choice of attributes such as age and gender. To our knowledge this is the first such evaluation in the fairness literature.

First we train a standard Random Forest, and measure the Discrimination for each attribute using (7) or (8) as appropriate. From these we record the average and standard deviation of the "Raw" discrimination. Then we train a new Fair Forest D times for D features, testing the model when each feature is selected as the protected attribute. We then measure the Discrimination of the protected feature and the accuracy of the resulting model. The mean and standard deviation are then calculated from the protected feature Discriminations. The results of this are shown in Table 5.

Across all three datasets and every feature, the Fair Forest approach was always able to decrease the Discrimination with respect to the protected attribute. For the German and Health datasets, it is able to reduce the Discrimination to zero for all features, and always results in the same accuracy. For the Adult dataset, the original protected attribute of Gender was the only attribute which could be reduced to a Discrimination of zero. The Adult dataset is the only one producing a wide impact in the amount of Discrimination removed, and the resulting accuracy of the model (decreasing from 0.85 down to 0.80 on average).

Table 5. **Discrimination statistics for all features in each dataset. First row is the Discrimination without any protection. The second row shows Discrimination when protecting each feature individually, and third row shows the associated model accuracy.**

	German		Adult		Health	
	μ	σ	μ	σ	μ	σ
Raw Discrim	0.0081	0.0137	0.2971	0.1652	0.0066	0.0101
Prot. Discrim	0.0000	0.0000	0.1253	0.0776	0.0000	0.0000
Prot. Accuracy	0.7000	0.0000	0.8044	0.0108	0.8474	0.0000

6 CONCLUSION

We have developed, to the best of our knowledge, the first fair variant of the Random Forest algorithm. This Fair Forest can be used for classification and regression problems, and protected k-category features as well as numeric attributes, a first in the fairness literature. In doing so we have extended the measure of discrimination to these cases. We have shown our method produces state-of-the art results on three common benchmark datasets, while requiring no parameter tuning to use, and is able to uniformly reduce Discrimination across any feature in each corpus.

REFERENCES

[1] Yahav Bechavod and Katrina Ligett. 2017. Learning Fair Classifiers: A Regularization-Inspired Approach. In *FAT ML Workshop*. http://arxiv.org/abs/1707.00044

[2] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2017. A Convex Framework for Fair Regression. In *FAT ML Workshop*. http://arxiv.org/abs/1706.02409

[3] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[4] Leo Breiman. 2003. Manual on setting up, using, and understanding random forests v4.0. *Statistics Department University of California Berkeley, CA, USA* (2003).

[5] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. CRC press.

[6] Toon Calders, Asim Karim, Faisal Kamiran, Wasif Ali, and Xiangliang Zhang. 2013. Controlling Attribute Effect in Linear Regression. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 71–80. https://doi.org/10.1109/ICDM.2013.114

[7] Toon Calders and Sicco Verwer. 2010. Three Naive Bayes Approaches for Discrimination-free Classification. *Data Min. Knowl. Discov.* 21, 2 (9 2010), 277–292. https://doi.org/10.1007/s10618-010-0190-x

[8] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: Reliable Large-scale Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[9] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness Through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS '12)*. ACM, New York, NY, USA, 214–226. https://doi.org/10.1145/2090236.2090255

[10] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. 2017. Decoupled classifiers for fair and efficient machine learning. In *FAT ML Workshop*. https://doi.org/1707.06613

[11] Harrison Edwards and Amos Storkey. 2016. Censoring Representations with an Adversary. In *International Conference on Learning Representations (ICLR)*. http://arxiv.org/abs/1511.05897

[12] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. 2014. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research* 15 (2014), 3133–3181. http://jmlr.org/papers/v15/delgado14a.html

[13] Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38, 4 (2002), 367–378. http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://www.sciencedirect.com/science/article/pii/S0167947301000652

[14] Eva García-Martín and Niklas Lavesson. 2017. Is it ethical to avoid error analysis?. In *FAT ML Workshop*. http://arxiv.org/abs/1706.10237

[15] Patrick Hall and Navdeep Gill. 2017. Debugging the Black-Box COMPAS Risk Assessment Instrument to Diagnose and Remediate Bias. (2017). https://openreview.net/pdf?id=r1iWHVJ7Z

[16] Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*.

[17] Faisal Kamiran and Toon Calders. 2009. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*. IEEE, 1–6. https://doi.org/10.1109/IC4.2009.4909197

[18] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware Learning Through Regularization Approach. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW '11)*. IEEE Computer Society, Washington, DC, USA, 643–650. https://doi.org/10.1109/ICDMW.2011.83

[19] Virgile Landeiro and Aron Culotta. 2016. Robust Text Classification in the Presence of Confounding Bias. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, 186–193. http://dl.acm.org/citation.cfm?id=3015812.3015840

[20] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2016. The Variational Fair Autoencoder. In *International Conference on Learning Representations (ICLR)*. http://arxiv.org/abs/1511.00830

[21] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. In *Advances in Neural Information Processing Systems 26*, C.j.c. Burges, L Bottou, M Welling, Z Ghahramani, and K.q. Weinberger (Eds.). 431–439. http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/281.pdf

[22] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. 2011. k-NN As an Implementation of Situation Testing for Discrimination Discovery and Prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 502–510. https://doi.org/10.1145/2020408.2020488

[23] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware Data Mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, USA, 560–568. https://doi.org/10.1145/1401890.1401959

[24] J R Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann series in {M}achine {L}earning, Vol. 1. Morgan Kaufmann. 302 pages. http://portal.acm.org/citation.cfm?id=152181

[25] Edward Raff. 2017. JSAT: Java Statistical Analysis Tool, a Library for Machine Learning. *Journal of Machine Learning Research* 18, 23 (2017), 1–5. http://jmlr.org/papers/v18/16-131.html

[26] Michael Skirpan and Micha Gorelick. 2017. The Authority of "Fair" in Machine Learning. In *FAT ML Workshop*. http://arxiv.org/abs/1706.09976

[27] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning Fair Representations. In *Proceedings of the 30th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Sanjoy Dasgupta and David McAllester (Eds.), Vol. 28. PMLR, Atlanta, Georgia, USA, 325–333. http://proceedings.mlr.press/v28/zemel13.html

APPENDIX

A.1 Proof of k-way Discrimination

Here we prove that (6) and (7) are equivalent when $k = 2$.

To simplify, let the predictive mean for the points T be μ, and for each subset T_i be μ_i. Writing out for $k = 2$ for (7), we get

$$\text{Discrimination} = |\mu - \mu_1| + |\mu - \mu_2|$$

Setting (6) and (7) equal to each other we get

$$|\mu_1 - \mu_2| = |\mu - \mu_1| + |\mu - \mu_2|$$

Assume, without loss of generality, that $\mu_1 > \mu > \mu_2$. In this case we can use the absolute value to re-write as

$$|\mu_1 - \mu_2| = |\mu_1 - \mu| + |\mu - \mu_2|$$

and then equivalently simplify as

$$\mu_1 - \mu_2 = (\mu_1 - \mu) + (\mu - \mu_2)$$

$$\mu_1 - \mu_2 = \mu_1 - \mu_2 + \mu - \mu$$

$$\mu_1 - \mu_2 = \mu_1 - \mu_2$$

.

Thus we obtain the same solution given a fixed ordering of the μs. The absolute value operation allows us to re-order the contained terms to match any distinct order of μs. For the case that $\mu_1 = \mu_2$, then it must be that $\mu = \mu_1$, and all terms will zero out. Therefor, we prove that (6) = (7).

A.2 Why Define Gain on Mean over Variance?

We take a moment to further expound upon why we have re-written the gain of numeric attributes with respect to the difference in means, when the original CART approach uses a criteria based on a reduction in variance. This original CART splitting condition can be defined by

$$\arg\min_{t} \frac{1}{t}\sum_{i=1}^{t}\sum_{j=1}^{t}\frac{1}{2}(x_i - x_j)^2 + \frac{1}{n-t}\sum_{i=t+1}^{n}\sum_{j=t+1}^{n}\frac{1}{2}(x_i - x_j)^2 \quad (10)$$

Where t is the splitting point, and $x_i \leq x_{i+1}$. This measures the sum of weighted variances between the two sets of points. This could be converted into our normalized gain form. Using the same notation, this would be

$$G_v(T, b) = 1 - \sum_{T_i \in \text{split}} \frac{|T_i|}{|T|} \frac{\sigma^2_{b,T_i}}{\sigma^2_{b,T}} \quad (11)$$

Using (11) would have the desirable property of being equivalent to the solution found by (10), and thus producing the same trees when there is no protected attribute. The problem with using this approach is that it does not align with our fairness goal, and can fail to produce fair trees when protecting numeric attributes.

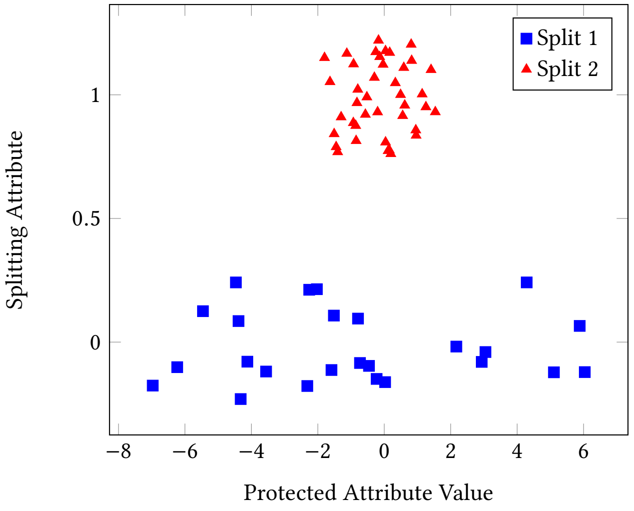

Figure 3. **Example of how splitting by variance fails to align with reducing bias. This split would receive a large penalty under a variance criteria, but both splits have the same mean — resulting in no discrimination.**

To demonstrate how this happens, consider the plot in Figure 3. Here we show the protected attribute's value on the x-axis, and the attribute we are splitting on on the y-axis. Choosing a split of $y \geq 0.5$, we see the data cleanly splits into two groups.

If we consider the variance-based gain defined by (11), this split would receive a large penalty. The split dramatically reduces the variance of the Split 2 group, which produces a large gain. Because it is the protected attribute, we subtract the gain — and thus a large penalty is applied.

Note though that the goal is to avoid discrimination in the predictions produced by the tree. Yet in this case, both splits have the same mean of zero, only their variances differ. If we were to think of the problem as predicting the protected attribute's value from the tree, we use the mean value of the attribute in the leaf nodes. The variance is forgotten anyway, and so we are penalizing a split which will not keep any significant information as it is.

Thus we prefer the intuition afforded by our new gain measure (5), which would produce no penalty for this split choice. The means would be the same, and so no predictive difference would occur with this split. Thus we find this split preferable for the protected attribute, as it would not aid in distinguishing the protected attribute at prediction time.