EURO Journal on Decision Processes 11 (2023) 100031

Contents lists available at ScienceDirect

EURO Journal on Decision Processes

journal homepage: www.elsevier.com/locate/ejdp

# Fairkit, fairkit, on the wall, who's the fairest of them all? Supporting fairness-related decision-making

Brittany Johnson[a], Jesse Bartola[b], Rico Angell[c], Sam Witty[c], Stephen Giguere[d], Yuriy Brun[c,*]

[a] Department of Computer Science, George Mason University, Virginia, United States
[b] Hubspot, Massachusetts, United States
[c] College of Information and Computer Sciences, University of Massachusetts Amherst, Massachusetts, United States
[d] Computer Science Department, University of Texas at Austin, Texas, United States

ARTICLE INFO

*Keywords:*
Software fairness
Algorithmic fairness
Tools for software engineers
Software engineering and artificial intelligence

ABSTRACT

Modern software relies heavily on data and machine learning, and affects decisions that shape our world. Unfortunately, recent studies have shown that because of biases in data, software systems frequently inject bias into their decisions, from producing more errors when transcribing women's than men's voices to overcharging people of color for financial loans. To address bias in software, data scientists and software engineers need tools that help them understand the trade-offs between model quality and fairness in their specific data domains. Toward that end, we present fairkit-learn, an interactive toolkit for helping engineers reason about and understand fairness. Fairkit-learn supports over 70 definition of fairness and works with state-of-the-art machine learning tools, using the same interfaces to ease adoption. It can evaluate thousands of models produced by multiple machine learning algorithms, hyperparameters, and data permutations, and compute and visualize a small Pareto-optimal set of models that describe the optimal trade-offs between fairness and quality. Engineers can then iterate, improving their models and evaluating them using fairkit-learn. We evaluate fairkit-learn via a user study with 54 students, showing that students using fairkit-learn produce models that provide a better balance between fairness and quality than students using scikit-learn and IBM AI Fairness 360 toolkits. With fairkit-learn, users can select models that are up to 67% more fair and 10% more accurate than the models they are likely to train with scikit-learn.

## 1. Introduction

Data-driven software is used increasingly to make automated decisions that shape our society. Software decides what products we are led to buy (Mattioli, 2012); who gets access to financial instruments (Olson, 2011) or gets hired (Raghavan et al., 2019); what a self-driving car does (Goodall, 2016), how medical patients are diagnosed and treated (Strickland, 2016), and when to grant bail (Angwin et al., 2016). Unfortunately, recent studies have shown that such software can inherit biases from data and the environment. For example, translation engines can inject societal biases into its translations (Caliskan et al., 2017). YouTube makes more mistakes when automatically generating closed captions female than male voices (Koenecke et al., 2020; Tatman, 2017). Racial bias affects the ads search engines display, e.g., showing ads for (nonexistent) arrest records when searching for African American names (Sweeney, 2013). Amazon's software has failed to offer same-day delivery to predominantly minority neighborhoods (Letzter, 2016), while, Staples offered online discounts to customers only in more affluent neighborhoods (Haweawar, 2012; Mikians et al., 2012). Language processing tools are more accurate on English written by white people than people of other races (Blodgett and O'Connor, 2017). Facial recognition software recognizes female and non-white faces less often and less accurately than those of white men (Buolamwini and Gebru, 2018; Klare et al., 2012). And the software US courts use to assess the risk of a criminal committing another crime exhibits racial bias (Angwin et al., 2016).

One fundamental cause of these biases is that modern software often applies machine learning to data generated from the real world. First, the real world is full of biases, often subconscious ones that the people who exhibit them do not recognize. In fact, humans often do not realize their biased behavior until they see an automated system reproduce it (Peng, 2019). Second, machine learning is notoriously opaque due to its probabilistic nature, sensitivity to small design decisions such as hyperparameter tuning, complex data preprocessing and model architecture, and nontransparent operation (Barocas, 2018; Doshi-Velez and Kim, 2017; Holstein et al., 2019). As a result, models learned from data can often encode discriminatory behavior from the data's bias, but that behavior is both hard to identify and eliminate (Galhotra et al., 2017).

* Corresponding author.
*E-mail address:* brun@cs.umass.edu (Y. Brun).

https://doi.org/10.1016/j.ejdp.2023.100031
Received 1 May 2022; Received in revised form 16 February 2023; Accepted 27 March 2023

Recently, public's demand for transparency in data and learned model use has increased (Albrecht, 2016), and governments have initiated efforts to increase regulation of decisions made by software systems to reduce bias and improve transparency (de Blasio, 2018; Executive Office of the President, 2016; Soper, 2016). As a result, it is increasingly important to provide support tools for those who apply machine learning to data, study data, and build software systems that use data to make decisions. These tools must support detecting and understanding biases in data and learned models, and the inherent trade-offs between mitigating bias and maximizing decision quality. Industry experts have called for tools that help data scientists and engineers understand bias in data and curate datasets, and to audit and debug fairness issues (Holstein et al., 2019), all of which our paper aims to address.

The challenges in helping engineers reason about fairness include: (1) Fairness (as well as quality, defined as, for example, precision, recall, accuracy, etc.) mean different things in different data domains, and no single definition of fairness is universally appropriate, with definitions often being mutually exclusive on datasets. (2) The trade-offs between fairness and quality are typically a function of the data and not of the tools applied to train models, and algorithms that produce fair models on some datasets may produce biased ones on others. (3) The space of possible models machine learning can produce is astronomically large due to the combinatorial explosion caused by a large number of learning algorithms, hyperparameters, and data permutations that affect the models. (4) Learning algorithms that attempt to account for fairness typically do not provide guarantees on the behavior of the models they produce (Zafar et al., 2015; 2017a), and can sometimes inject more bias than fairness-unaware algorithms (Galhotra et al., 2017); using fairness-aware algorithms to reduce one kind of bias can significantly increase other biases (Galhotra et al., 2017); and learning algorithms that do provide guarantees about their models' fairness can, under some conditions, break those guarantees (Agarwal et al., 2018) or fail to produce a model altogether, even if fair ones exist (Metevier et al., 2019; Thomas et al., 2019).

While some modern tools can measure various dimensions of fairness of a given model (Adebayo and Kagal, 2016; Bellamy et al., 2018; Galhotra et al., 2017; IBM, 2019; Tramer et al., 2017), and some machine learning algorithms can train models while enforcing fairness constraints (Agarwal et al., 2018; IBM, 2019; Metevier et al., 2019; Thomas et al., 2019; Zafar et al., 2015; 2017a), none of these tools provide support for understanding the trade-offs between fairness and quality of the models and for comparing and contrasting models along the combinations of fairness and quality measures they produce. For example, scikit-learn, the state-of-the-art go-to toolkit used ubiquitously by data scientists in industry (scikit-learn, 2019), provides tools for training many types of machine learning models, and evaluating them for quality, such as precision and recall, but not fairness metrics. IBM's open source toolkit, AI Fairness 360, adds support for computing fairness metrics on learned models and learning algorithms that account for some definition for fairness (Bellamy et al., 2018; IBM, 2019). Fairness-aware learning algorithms, such as fairlearn (Agarwal et al., 2018) or RobinHood (Metevier et al., 2019) and others designed using the Seldonian Framework (Thomas et al., 2019), can enforce fairness constraints, but without helping engineers understand how that enforcement affects model quality. The bottom line is, these tools still fail to provide support for understanding the trade-offs between fairness and quality, e.g., to help data scientists answer questions such as "Does finding a more fair model necessarily imply the model's quality will decrease, and by how much?"

Toward this end, we have developed fairkit-learn, a tool that builds on scikit-learn and IBM AI Fairness 360, to help data scientists and software engineers better understand the model fairness landscape. Fairkit-learn eases adoption by interfacing with scikit-learn, can support the over 70 definitions of fairness implemented in AI Fairness 360, and works with all of scikit-learn's and AI Fairness 360's algorithms, metrics, and datasets. There are also interfaces for easily including more definitions, metrics, and datasets. Fairkit-learn uses visualization to help engineers understand the fairness properties of specific models, which learning algorithms learn models that better satisfy competing requirements of fairness and quality in a particular domain, and demonstrate opportunities for selecting models that improve fairness or quality at the lowest expense of the other. For example, fairkit-learn can perform a grid search through tens of thousands of possible models learned using different machine learning algorithms with different combinations of hyperparameters, and select the Pareto-optimal set of models with respect to multiple data-scientist-selected fairness and quality definitions. Our goal is to complement, not replace, the existing landscape of tools that help data scientists make informed decisions about machine learning models by combining visualization and search features, and improving usability. We have previously published a short tool demonstration of fairkit-learn (Johnson and Brun, 2022), but the fairkit-learn design and evaluation contributions are unique to this paper.

Figure 1 shows a sample fairkit-learn visualization. Here, an engineer is comparing models learned by three learning algorithms — logistic regression, random forest classifier, and adversarial debiasing — on the COMPAS recidivism dataset. Fairkit-learn trains approximately 80 different models using these three algorithms by varying their hyperparameters in a grid-search, and computes the much smaller (here, seven) subset of the models that make up the Pareto-optimal set. Fairkit-learn visualizes the seven models with respect to two metrics selected by the user: disparate impact, a fairness metric, visualized on the x-axis, and model accuracy, a quality metric, visualized on the y-axis. The visualization elides multiple sub-optimal models to show only those for which improving fairness decreases accuracy, and vice versa (the Pareto-optimal model set). This visualization makes it easy to see that (1) in this data domain, model fairness and model accuracy are opposing forces (in other domains, they can be complementary), (2) a small reduction in quality (63% versus 68%) can produce a large increase in fairness (69% versus 45%), and (3) random forest classifier models (orange) tend to produce more-fair models at a slight cost in accuracy, adversarial debiasing (magenta), a machine learning algorithm intended to be fairness-aware, produces less fair but slightly more accurate models, and logistic regression (purple) models perform slightly better than adversarial debiasing.

We evaluate fairkit-learn in a controlled user study with 54 students studying data science and software engineering.[1] (Recent studies Höst et al., 2000; Naiakshina et al., 2019 have demonstrated that, in studies like ours, findings from student subjects generalize to findings from professional subjects.) Our within-subject study asked subjects to use scikit-learn, IBM AI Fairness 360, and fairkit-learn to explore the machine-learning-model landscape on three datasets, aiming to produce models that satisfy a combination of fairness and quality metrics. We found that subjects who used fairkit-learn produced more fair models than when using scikit-learn and that while IBM AI Fairness 360 may be better for engineers only interested in improving fairness, fairkit-learn supports finding models that are both fair and of high quality (more so than AI Fairness 360). With fairkit-learn, users can select models that are up to 67% more fair and 10% more accurate than the models they are likely to train with scikit-learn.

*Our work's key contributions:*

- Fairkit-learn, a novel open-source tool that uses familiar interfaces and visualization for exploring, evaluating, and visualizing the performance and fairness trade-offs in machine learning models. Unlike existing tools:
  - Fairkit-learn supports combining multiple fairness and quality considerations when evaluating and comparing models. This al-

[1] This human-subject study was approved by the UMass Amherst Institutional Review Board.

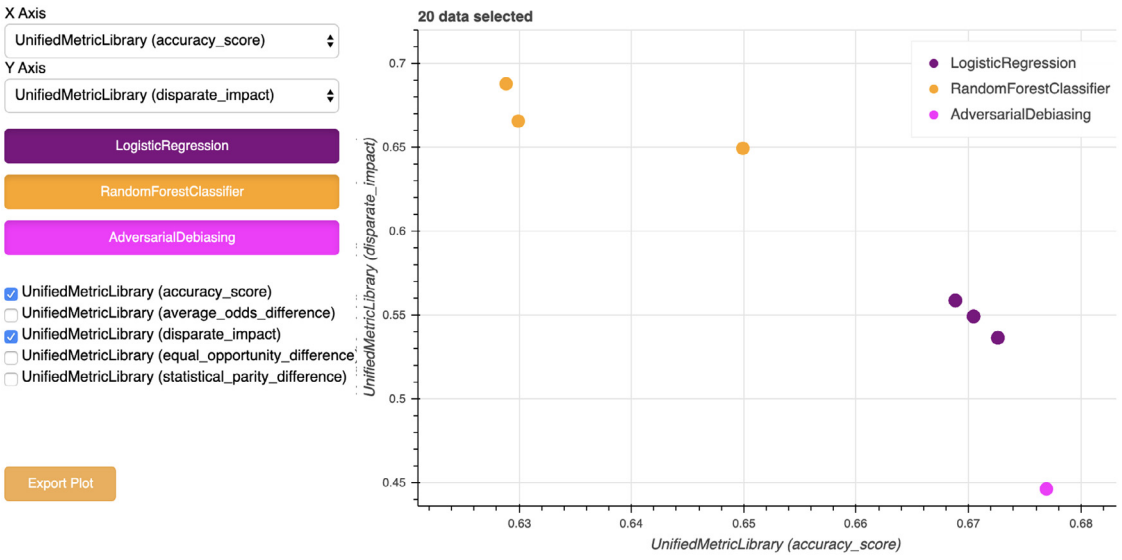


**Fig. 1.** Fairkit-learn trains and evaluates a large-number of machine learning models using multiple learning algorithms (here, logistic regression, random forrest, and adversarial debiasing) and an array of hyperparameters, finding the Pareto-optimal set of models that represent the best combination of quality metrics (here, accuracy, shown on the y-axis) and fairness metrics (here, disparate impact, shown on the x-axis). Fairkit-learn's visualization helps engineers understand the domain of their data (here, the COMPAS recidivism dataset Angwin et al., 2016), explaining relationships and trade-offs between quality and fairness metrics, and showing which algorithms achieve better combinations of multiple metrics.

lows data scientists to optimize their models with respect to multiple fairness and quality factors simultaneously.

- Fairkit-learn simplifies the process of exploring the space of possible models by automatically performing grid searches over multiple learning algorithms, model hyperparameters, and data permutations, lifting the burden of implementing such a search off the user.
- Instead of auditing learned models for fairness (as, e.g., Adebayo and Kagal, 2016; Galhotra et al., 2017; Tramer et al., 2017), fairkit-learn helps engineers understand the fairness-quality trade-off landscape *during* model development, allowing them to make design decisions about these trade-offs when those decisions can still affect overall system performance.
- Fairkit-learn works with more than 70 definitions of fairness and quality, allowing engineers to evaluate the applicability of different definitions to their data domains and select those that make most sense in their particular situations.
- Fairkit-learn uses an interactive, visualization-based approach that displays the Pareto-optimal set of solutions, to clearly communicate trade-offs to the engineers, helping them make informed decisions.

- A user study, evaluating fairkit-learn against scikit-learn and IBM AI Fairness 360, showing that subjects using fairkit-learn train models that better balance fairness and accuracy. While, in the real world, often there is no single best model, fairkit-learn helped subjects select Pareto-optimal models. Our study also provides insights into how engineers reason about fairness when using traditional machine learning tools, e.g., scikit-learn, to train and evaluate models.

It is important to observe that enacting model fairness is a complex task, and that fairkit-learn is not a silver bullet that will, alone, solve this problem — it helps engineers evaluate fairness of models to be used in software, and can help them understand fairness-quality trade-offs, but our controlled study observed that engineers may still make poor decisions even when informed, at times failing to select the fairest models. Our study shows that fairkit-learn forms a useful toolkit for understanding and building fair models to be used in software systems, but that it does not represent the end-all solution and that further work on improving interfaces and tools engineers can use is necessary and warranted.

A recent study has identified gaps in existing fairness-supporting tools (Lee and Singh, 2021), and fairkit-learn explicitly addresses these gaps. Fairkit-learn is a Python library and uses the same interface as scikit-learn, making it easy to integrate into engieers' workflows and significantly reducing the learning curve. Fairkit-learn avoids information overload by providing an interactive visualization of only Pareto-optimal models; it avoids oversimplification by providing methods for digging deeper and revealing additional model information when needed, and in different formats. Fairkit-learn supports end-to-end model development considerations by putting data cleaning algorithms, learning algorithms, analysis, and visualization tools all into one toolkit with one common interface, and supporting interactive model and fairness definition selection and refinement.

The rest of the paper is structured as follows: Section 2 describes fairkit-learn. Section 3 outlines our evaluation methodology. Section 4 presents and discusses the evaluation results and the threats to their validity. Section 5 places our research in the context of related work and Section 6 summarizes our contributions.

## 2. The fairkit-learn toolkit

Fairkit-learn is an open-source, publicly available Python toolkit designed to help engineers evaluate and explore machine learning models with respect to quality and fairness metrics simultaneously.[2]

Fairkit-learn builds on top of scikit-learn, the state-of-the-art tool suite for data mining and data analysis, and AI Fairness 360, the state-of-the-art Python toolkit for examining, reporting, and mitigating machine learning bias in individual models (IBM, 2019; scikit-learn, 2019). Fairkit-learn supports all metrics and learning algorithms available in scikit-learn and AI Fairness 360, and all of the bias mitigating pre- and post-processing algorithms available in AI Fairness 360, and provides extension points to add more metrics and algorithms.

This section describes the complexity of model fairness and the space of fairness definitions fairkit-learn handles, fairkit-learn's search capabilities for helping engineers explore and understand the space of possible models and data permutations, and fairkit-learn's analysis of

[2] http://go.gmu.edu/fairkit-learn

Pareto-optimal sets of models and visualization capabilities for illustrating trade-offs between model fairness and quality.

### 2.1. Integrated machine learning tools

We selected two existing machine learning toolkits as the foundation for fairkit-learn: scikit-learn (scikit-learn, 2019) and IBM's AI Fairness 360 (IBM, 2019). We discuss each of these tools separately below.

#### 2.1.1. Scikit-learn

Scikit-learn is a commonly used and integrated machine learning toolkit, therefore we wanted to ensure that fairkit-learn works with its models and functionality. While scikit-learn provides a number of algorithms and metrics for training and evaluating machine learning models, it does not support training or evaluating models for fairness. It also does not have built-in support for exploring the space of machine learning model configurations; if an engineer wants to find an optimal model for a given metric, she must implement the code to do so herself. Scikit-learn also only supports evaluating machine learning models by one metric at a time — any trade-off analysis has to be written by the user.

#### 2.1.2. AI fairness 360

IBM AI Fairness 360 provides an exhaustive set of datasets, models, algorithms, and metrics that pertain to machine learning model fairness, so we used this toolkit as the foundation for fairkit-learn's fairness components. Along with this large set of functionalities, the website provides detailed documentation and examples for using the various components of the toolkit.[3] And like fairkit-learn, AI Fairness 360 is built using scikit-learn. However, AI Fairness 360, like scikit-learn, does not provide built-in support for exploring the space of models and configurations nor does it provide support for evaluating trade-offs between multiple metrics. Any trade-off evaluations, along with model configuration exploration, would have to be implemented by the user.

### 2.2. Fairness metrics

Fairness is a broad notion that can be partially represented by many formal definitions (Narayanan, 2018). Unfortunately, users, data scientists, and regulators rarely agree on a single definition (Grgic-Hlaca et al., 2018), though they often agree that fairness, in some form, is important (Woodruff et al., 2018). In fact, while each definition of model fairness is appropriate in some context (Makhlouf et al., 2021), many are impossible to satisfy simultaneously (Friedler et al., 2016; Kleinberg et al., 2017). To effectively support engineers across many domains, **fairkit-learn supports many fairness definitions, including all 70+ supported by IBM AI Fairness 360** (Bellamy et al., 2018; IBM, 2019), and provides extension points to add more.

Here, we describe several representative definitions of fairness fairkit-learn handles to give the reader as sense of their diversity. More complete lists exist, e.g., Narayanan (2018), and active research in the area of fairness definitions is continually expanding that list at this time.

Together with intuitive descriptions of each definition, we include a formal mathematical defintion, similar to what prior work Thomas et al. (2019) has done. For these definitions, we consider three random variables, $X$, $T$, and $Y$, where $X$ is the set of non-sensitive attributes, $T$ is the set of sensitive attributes, and when applied to classification, $Y$ is restricted to being in a (typically small) discrete set (here, we will focus on *binary classification*, where $Y \in \{-1, +1\}$). We refer to $X \times T$ as the *feature vector* and we refer to $Y$ as the *label*. A classifier $\theta$ is a function that consumes an element of $X \times T$ and produces an element of $Y$ ($\theta : X \times T \to Y$). In other words, for a datapoint $d = (x, t) \in X \times T$, $\theta(d) = \hat{y} \in Y$. Note that we make a distinction between the true label $y$ and the classifier-produced label $\hat{y}$. While the classifier's goal is to determine the correct $y$, it may not always succeed. The classifier $\theta$ is typically learned from a dataset $D \subseteq X \times T \times Y$ of labeled data, though most definitions (all below except disparate treatment) of fairness are agnostic to how $\theta$ is created. For simplicity of exposition, we assume below that $T = \{0, 1\}$, that is, there is a single protected attribute with two possible values; however, all definitions apply to scenarios with multiple protected attributes, with multiple possible values each. We refer to the set of all feature vectors with the same value for the protected attribute as a *group*. We use the notation $\Pr(\hat{y}(X, \theta){=}{+}1 | T{=}\tau)$ to mean the fraction of feature vectors in a group that the model classifies with label $+1$ (members of the positive class).

- **Disparate treatment** is a concept originally of legal origins. The computer science formalization of this definition says that for a model to satisfy disparate treatment with respect to a set of attributes, it must have been learned without access to those attributes (Zafar et al., 2017a). Formally, $\theta : X \to Y$ (as opposed to the general case of $\theta : X \times T \to Y$). However, this definition often fails to ensure meaningful fairness in practice, because data attributes $X$ and $T$ are often correlated, e.g., age correlates with savings, race correlates with name, and, in the United States, race correlates with zip code, models trained without access to a set of attributes $T$ can still effectively act unfairly with respect to those attributes (Ingold and Soper, 2016; Sweeney, 2013).
- **Disparate impact** captures the notion that a model may have adverse effects on protected groups (Chouldechova, 2017; Griggs v. Duke Power Co., 1971; Zafar et al., 2017a). To satisfy the disparate treatment definition, a model must treat similarly the same fraction of individuals of each group. For example, if an employer hires $\frac{1}{2}$ of its male applicants, then that employer must hire at least $\frac{1}{2}$ of its female applicants (Griggs v. Duke Power Co., 1971). If the fractions are different, the ratio $p$ between the fractions is a measure of bias. Formally, for a classifier $\theta$

$$p = \min\left(\frac{\Pr(\hat{y}(X,\theta){=}{+}1|T{=}0)}{\Pr(\hat{y}(X,\theta){=}{+}1|T{=}1)}, \frac{\Pr(\hat{y}(X,\theta){=}{+}1|T{=}1)}{\Pr(\hat{y}(X,\theta){=}{+}1|T{=}0)}\right).$$

- **Demographic parity**, also called **statistical parity** and **group fairness**, is closely related to disparate impact, and requires that the model's predictions are statistically independent of the attribute with respect to which the model is fair (Calders and Verwer, 2010; Dwork et al., 2012). The measure of bias is, unlike for disparate impact, the difference between the fractions. Formally, the measure of demographic parity $p$ of a classifier $\theta$ is

$$p = |\Pr(\hat{y}(X,\theta){=}{+}1|T{=}0) - \Pr(\hat{y}(X,\theta){=}{+}1|T{=}1)|.$$

- **Delayed impact** is concerned with the fact that making seemingly fair decisions can, in the long term, produce unfair consequences (Liu et al., 2018). For example, to make up for a disparity in recidivism predictions by race, a model may, at random, decrease its predictions for one race. While on its face, this may improve the situation for members of that race, if this results in more visibility for repeat offenders of that race, the public's perception may have a more negative effect toward that race, producing delayed negative impact. Measuring delayed impact requires temporal indicator data, of, for example, long-term improvement, stagnation, and decline in variables of interest (Liu et al., 2018).
- **Predictive equality** requires that false positive rates are equal among groups (Chouldechova, 2017; Corbett-Davies et al., 2017). Formally, the measure of predictive equality $p$ of a classifier $\theta$ is

$$p = |\Pr(\hat{y}(X,\theta){=}{+}1|T{=}0, Y{=}-1) - \Pr(\hat{y}(X,\theta){=}{+}1|T{=}1, Y{=}-1)|.$$

Note that this definition only considers feature vectors whose true label is $y = -1$.

[3] https://aif360.mybluemix.net

- **Equal opportunity** requires that false negative rates are equal among groups (Chouldechova, 2017; Hardt et al., 2016). Formally, the measure of equal opportunity $p$ of a classifier $\theta$ is

$$p = |\Pr(\hat{y}(X,\theta) = -1 | T=0, Y=+1) - \Pr(\hat{y}(X,\theta) = -1 | T=1, Y=+1)|.$$

  Note that this definition only considers feature vectors whose true label is $y = +1$.
- **Equalized odds**, a combination of predictive equality and equal opportunity, requires that both false positive and false negative rates are equal among groups (Hardt et al., 2016). Consequently, the equalized odds criterion can be viewed as the conjunction of the predictive equality and equal opportunity criteria. Formally, the measure of equalized odds $p$ of a classifier $\theta$ is the mean of predictive equality and equal opportunity.
- **Treatment equality** requires that the ratio of the false-positive rate to the false-negative rate is the same for each group (Berk et al., 2018). Formally, the measure of treatment equality $p$ of a classifier $\theta$ is

$$p = \left| \frac{\Pr(\hat{y}(X,\theta) = -1 | T=0, Y=+1)}{\Pr(\hat{y}(X,\theta) = +1 | T=0, Y=-1)} - \frac{\Pr(\hat{y}(X,\theta) = -1 | T=1, Y=+1)}{\Pr(\hat{y}(X,\theta) = +1 | T=1, Y=-1)} \right|.$$

- **Causal fairness** is based on the counterfactual causal relationship between variables. To be causally fair, a classifier must predict the same label for all feature vectors that are the same except for those attributes. In other words, if two individuals differ only in protected attributes, and are otherwise identical, this definition requires classifiers to predict the same outcome for both individuals (Galhotra et al., 2017). For example, a recidivism model is causally fair with respect to race only if it predicts identical labels for all pairs of individuals identical in every way except race. Formally, the measure of causal fairness $p$ is the fraction of the feature vectors whose only differences are in the $T$ attributes for which the $\hat{y}$ labels the classifier $\theta$ assignes differ.

$$p = \Pr(\hat{y}(X,\theta) | T=0 \neq \hat{y}(X,\theta) | T=1).$$

  The term causal fairness has also been used to describe a broader set of definitions based on Pearl's causal framework (Kusner et al., 2017; Pearl, 2009).
- **Counterfactual fairness** similarly attempts to measure the causal impact of changing a sensitive attribute of an individual, but, unlike the above definition of causal fairness, models the relationship between sensitive and other attributes (Kusner et al., 2017).
- **Individual fairness**, also referred to as **metric fairness**, requires that, given a distance metric to compare two feature vectors, the model should predict similar labels for similar feature vectors, on average (Dwork et al., 2012). **Approximate metric fairness** extends this definition by incorporating a tolerance parameter to obtain generalization bounds (Rothblum and Yona, 2018).
- **Representation disparity** limits the error for all subgroups (Hashimoto et al., 2018). The amount of representation disparity is the maximum loss for any particular group. Formally, the measure of representation disparity $p$ for the classifier $\theta$ is the maximum loss for any particular group:

$$\max_{\tau \in \{0,1\}} \mathbf{E}[\ell(X,\theta) | T=\tau],$$

  where $\ell(X,\theta)$ is the loss associated with the parameter vector, $\theta$.
- **Conditional use accuracy equality** requires that precision (the probability that the model is correct when it predicts a label) is the same for all groups (Berk et al., 2018). Formally, the measure of conditional use accuracy equality $(p_+, p_-)$ of a classifier $\theta$ is

$$p_+ = |\Pr(Y=+1 | T=0, \hat{y}(X,\theta)=+1) - \Pr(Y=+1 | T=1, \hat{y}(X,\theta)=+1)|$$

  and

$$p_- = |\Pr(Y=-1 | T=0, \hat{y}(X,\theta)=-1) - \Pr(Y=-1 | T=1, \hat{y}(X,\theta)=-1)|.$$

- **Overall accuracy equality** requires that the accuracy of the classifier (fraction of the feature vectors that the model correctly classifies) is equal for each group (Berk et al., 2018). Formally, the measure of overall accuracy equality $p$ is

$$p = |\Pr(Y = \hat{y}(X,\theta) | T=0) - \Pr(Y = \hat{y}(X,\theta) | T=1)|. \quad (1)$$

### 2.3. Model search

Unlike existing tools, which require engineers to write their own code to evaluate more than one model configuration, fairkit-learn provides functionality that allows engineers to search over any number of model configurations (given enough memory and power) for Pareto-optimal solutions (that best balance quality metrics of concern and fairness). Figure 2 shows code that initializes each parameter required for the model search: *models, metrics, hyperparameters, thresholds*, and *pre-/post-processing algorithms*.

#### 2.3.1. Models

To run the grid search, you need to specify at least one model to include (`models` in Fig. 2). You can specify as many models as available computational resources will allow. Fairkit-learn is currently compatible with scikit-learn and AI Fairness 360, but can be extended to work with others via the model wrapper class provided.

#### 2.3.2. Metrics

Also required for the grid search are metrics to evaluate each model configuration (`metrics` in Fig. 2). Fairkit-learn is currently compatible with metrics from scikit-learn and AI Fairness 360, but uses a wrapper metric class that can be extended with other metrics.

#### 2.3.3. Hyperparameters

Fairkit-learn can run using default hyperparameters, or users can provide different values for each hyperparameter to evaluate in the grid search. The example in Fig. 2 runs the `AdversarialDebiasing` and `RandomForestClassifier` models with default parameters and provides options for two of the `LogisticRegression` model hyperparameters.

#### 2.3.4. Thresholds

The threshold parameter denotes the probabilistic threshold required to be considered a positive classification (in a binary classification). For example, if the threshold is 0.7, then any prediction with $\geq 0.7$ probability will be considered favorable.

#### 2.3.5. Pre- and post-processing algorithms

Finally, users have the option of specifying any data preprocessing or model post-processing algorithms to include in the search (`preprocessors` and `postprocessors` in Fig. 2). Fairkit-learn currently works with pre- and post-processing algorithms provided by AI Fairness 360.

Once the search is done, results are written to a.csv file. The.csv file is used to render a visualization of the results of the grid search.

### 2.4. Search result visualization

To help engineers process the results of the grid search, fairkit-learn provides functionality that allows users to visualize the results. The visualization shown in Fig. 1 is showing some results from the search shown in Fig. 2. More specifically, the visualization is showing the *Pareto frontier* of the `LogisticRegression`, `RandomForestClassifier`, and `AdversarialDebiasing` models with respect to accuracy and (`accuracy_score`) and fairness (`disparate_impact`).

When using the fairkit-learn visualization, one can view the Pareto frontier of any two metrics by selecting those metrics in the checklist

```
models = {'LogisticRegression': LogisticRegression,
          'RandomForestClassifier': RandomForestClassifier,
          'AdversarialDebiasing': AdversarialDebiasing}

metrics = {'UnifiedMetricLibrary': [UnifiedMetricLibrary,
                                    'accuracy_score',
                                    'average_odds_difference',
                                    'statistical_parity_difference',
                                    'equal_opportunity_difference',
                                    'disparate_impact'
                                   ]

processor_args = {'unprivileged_groups': unprivileged, 'privileged_groups': privileged}
hyperparameters = {'LogisticRegression':{'penalty': ['l1', 'l2'], 'C': [0.1, 0.5, 1]},
                   'RandomForestClassifier':{},
                  'AdversarialDebiasing': DEFAULT_ADB_PARAMS(**processor_args)
                  }

thresholds = [i * 10.0/100 for i in range(5)]

preprocessors=[DisparateImpactRemover(), Reweighing(**processor_args)]
postprocessors=[CalibratedEqOddsPostprocessing(**processor_args)]
```

**Fig. 2.** Example parameters for model search in fairkit-learn.

and for the X and Y axes (as shown in Fig. 1). To access all search results (including not Pareto optimal), select all metrics and choose the X and Y axis you want to view. Engineers can also toggle which models to show in the plot by clicking the model button (e.g., the magenta `AdversarialDebiasing` button) hover over the data points in a given plot to get more information on the model configuration at that point (e.g., hyperparameter values). The visualization can also be exported for later viewing and comparison, along with a JSON file that describes the exported plot.

## 3. Evaluation methodology

To evaluate fairkit-learn and explore how engineers train fair models, we conducted a user study to validate the following hypotheses:

1. $H_1$ Compared to out-of-the-box scikit-learn models, fairkit-learn supports training fairer models.
2. $H_2$ When asked to find the most fair model, individuals who use fairkit-learn are able to train models that are more fair.
3. $H_3$ When asked to find a model that best balances fairness and accuracy, individuals who use fairkit-learn are able to train models that are more fair and comparably accurate.

We also collect data to answer the question *how do engineers reason about model fairness when not using fairness tooling?* and we qualitatively explore some of the reasons why fairkit-learn's visual interface enables reasoning about fairness and quality in ways scikit-learn and AI Fairness 360 do not.

All experimental artifacts, including the Jupyter Notebooks used in our study, are available online for future reference and replication.[4] The publicly available fairkit-learn implementation also contains Jupyter-Notebook-based tutorials for using fairkit-learn.[5]

### 3.1. Datasets

We used three real-world datasets to evaluate fairkit-learn. Each dataset has its own definition of which groups are privileged and can be used for binary classification tasks.

#### 3.1.1. Task 1: ProPublica COMPAS dataset

The COMPAS dataset is publicly available and contains recidivism data for defendants in Browards County between 2013 and 2014 (ProPublica, 2019). For each individual in the dataset, the dataset includes their criminal history both before and after arrest, and the risk assessment score, as calculated by the COMPAS system (Angwin et al., 2016). In 2016, ProPublica found significant differences between predictions the COMPAS system made based on race, finding that the system more often predicted African American defendants would commit a crime again, when, in reality, they did not, while predicting that white defendants would not a commit a crime again, when, in reality, they did. Data scientists can use this dataset to train models to predict recidivism and to ensure fairness. For our analyses, we treated Caucasian females as the the privileged group, treating race and sex as protected attributes.

#### 3.1.2. Task 2: German credit dataset

The German credit dataset is publicly available and contains financial data of 1,000, some of whom are classified as potential credit risks (Statlog, 1994b). The dataset consists of attributes ranging from credit history to personal status and sex. Engineers can use the German credit dataset when, for example, training models for use in banking or loan approval software. For our analyses, we treated the men 25 years of age or older as the privileged group, treating age and sex as protected attributes.

#### 3.1.3. Task 3: Adult census income dataset

The Adult census income dataset is publicly available and contains Census data, such as race, occupation, and salary, for 48,842 individuals from 1994 (Statlog, 1994a). Engineers can use this dataset to train models that make income predictions (e.g., whether a person make more than US$50K per year). For our analyses, we treated Caucasian men as the privileged group, treating sex and race as the protected attributes.

### 3.2. State-of-the-art comparison

When we first developed and publicly released the code for fairkit-learn, no other visualization-based fairness-aware toolkit existed. Soon after, IBM released AI Fairness 360 (IBM, 2019). Thus, for our evaluation, we compare to AI Fairness 360, and also scikit-learn, the state-of-the-art relevant tools that existed at the time of our study.

Since then, several other relevant tools have emerged. For example, Microsoft's fairlearn (Agarwal et al., 2018), which was only a machine learning algorithm that provided no visualization capabilities at the time of our study, has added visualization features after our study. In general, in this fast-moving field, many relevant industrially created frameworks have emerged. What allows industry to produce these frameworks so rapidly is that it follows a different set of standards for release. For example, neither IBM's AI Fairness 360 nor Microsoft's fairlearn visualization toolkit published a scientific, peer-reviewed paper evaluating

[4] https://go.gmu.edu/fkl-study-materials
[5] http://go.gmu.edu/fairkit-learn

their tools against the state of the art. Instead, IBM released an non-peer-reviewed blog post (Varshney, 2018) and Microsoft a non-peer-reviewed technical report (Bird et al., 2020) that describe their tools but do not evaluate them. Similarly, FAT Forensics (Sokol et al., 2020) is published as open-source software, but has no peer-reviewed evaluation against other tools. And Google's ML-fairness-gym (Ml-fairness-gym) is peer-reviewed (D'Amour et al., 2020), but it too was not evaluated against other frameworks.

As we strive for a higher bar, we evaluated our fairkit-learn against the state-of-the-art techniques available at the time of our study, and pursued peer review. Because industrial tools can be released quickly without those requirements, and user-based evaluations take significant time and resources, they cannot be simply repeated every time a new industrial toolkit emerges. In the time that evaluation takes place, a new toolkit would already be released.

### 3.3. User study design

To validate our hypotheses and answer our research question, we designed a user study to explore the effects of various tooling on the machine learning models engineers train. The state-of-the-art in training and evaluating machine learning models, and at the core of both fairkit-learn and AI Fairness 360, is scikit-learn. Therefore, we designed our experiment with two control groups: one that only uses functionality provided by scikit-learn and the other only using AI Fairness 360.

To make our study design more realistic, we created a Jupyter Notebook[6] for each experimental group. We presented the notebooks to participants as a homework assignment with three tasks.

Each notebook provided information on the tasks, relevant details, and links to external documentation. For each task, we provided participants with a real-world dataset and a tutorial on how to use one of the tools. Following each tutorial, we asked them to complete the following subtasks:

1. Find a machine learning model you believe will be the most accurate.
2. Find a machine learning model you believe will be the most fair.
3. Find a machine learning model you believe will best balance both accuracy and fairness.

For our evaluation, we selected a subset of the metrics available for use in fairkit-learn. However, as previously mentioned, users of fairkit-learn can incorporate and use any fairness metric of their choice. To complete the tasks, we gave each participant all the necessary study materials and instructions for participation. The first task notebook provided participants with background information on what they would be doing, the tools they will be using, and where to submit their responses.

Our design consisted of 6 experimental groups. To reduce the effects of learning bias on the validity of our findings, each experimental group used the tools in a different order. We used the same three datasets from AI Fairness 360 for all 6 experimental groups. This allowed us to increase confidence that our findings generalize.

After the exercise was complete, we collected participant notebooks and other relevant data. Next, we outline what data we collected and how.

### 3.4. Data collection

We collected data from participants as they completed the exercise in an online response form. The form consisted of 5 pages. The first page asked participants demographic questions, including questions about their experience with Python and various Python machine learning tools. The next three pages corresponded to each task where we asked participants variations of the following questions, depending on the subtask:

1. *Describe the best model and report its metric(s) scores.*
2. *Why did you select this model?*

We also collected notebook changes and snapshots using nbcomet,[7] an open source tool for tracking Jupyter Notebook changes. We used this data to triangulate with form responses when possible and necessary.

### 3.5. Data analysis

After data collection, we had to clean, prepare, and analyze the data. The first step in our data analysis was to extract and clean responses from the response form. We first had to extract, organize, and make each participants' responses anonymous. We organized participant responses by task, and then by tool within each task, since that is how we planned to analyze the data.

Research has shown that one of the challenges engineers have is dealing with the machine learning aspects of working with data, such as feature engineering and hyperparameter tuning (Kim et al., 2017; Sanders and Giraud-Carrier, 2017). To evaluate $H_1$, we compared our study's base scikit-learn models with default parameter settings to the models participants selected using fairkit-learn. For each default model, we calculated accuracy and fairness scores for each fairness metric used in the study. We then compared the averages of each for the default models to the averages for participant models selected when using fairkit-learn for fairness related subtasks.

To evaluate $H_2$ and $H_3$, we calculated fairness scores and accuracy for each of participants' model selections in the "find the most fair model" and "find the model that best balances both" subtasks. We averaged scores for each metric and measured the difference between those averages using a two-sample t-test ($\alpha = 0.05$).

We further analyzed responses from the tasks where participants used scikit-learn to find the most fair model. We extracted model selections and the qualitative and quantitative rationales for fair model selections in the response form. We then categorized the methods used by participants into the following categories: (1) *did not try to evaluate for fairness*, (2) *evaluated with a metric*, (3) *evaluated with something other than a metric*, and finally (4) *implemented one or more fairness metrics for evaluation*. We kept track of metrics, and other information, used when participant did try to evaluate fairness.

### 3.6. Participants

We recruited 54 participants from an advanced software engineering course: 30 undergraduates and 24 graduate students. One participant reported having industry experience as a data scientist.

Twenty-six participants had experience with using scikit-learn prior to participating. One participant had prior experience using AI Fairness 360 and no participants had prior experience with fairkit-learn. Fifty-one participants had experience with other Python data science and machine learning tools, such as numpy, scipy, and tensorflow. Fourty participants had prior experience using Jupyter Notebooks. On average, participants had approximately 2 years of Python programming experience; this excludes two participants who did not report any years of experience with Python despite reporting having experience with various Python tools.

## 4. Evaluation results and discussion

We used data collected from our user study to validate and explore how engineers train models and evaluate them for fairness and accuracy. In comparing fairkit-learn models to scikit-learn default models, we found that even with concerns split between fairness and accuracy, participants selected fairer, more accurate models when using fairkit-learn (Fig. 3). When fairness is the only concern, our study found that

[6] https://jupyter.org/

[7] https://github.com/activityhistory/nbcomet

| Tool | Average Odds (lower is better) | Statistical Parity (lower is better) | Equal Opportunity (lower is better) | Disparate Impact (higher is better) | Accuracy (higher is better) |
|---|---|---|---|---|---|
| scikit-learn (default) | $0.173 \pm 0.214$ | $0.221 \pm 0.235$ | $0.150 \pm 0.202$ | $0.555 \pm 0.281$ | $0.741 \pm 0.248$ |
| fairkit-learn (fairness) | $0.116 \pm 0.083$ | $0.225 \pm 0.108$ | $0.093 \pm 0.075$ | $0.725 \pm 0.116$ | $0.788 \pm 0.106$ |
| fairkit-learn (fairness+accuracy) | $0.086 \pm 0.073$ | $0.229 \pm 0.109$ | $0.070 \pm 0.066$ | $0.829 \pm 0.098$ | $0.815 \pm 0.101$ |

**Fig. 3.** Mean fairness scores (across all three tasks) of default scikit-learn models and fairkit-learn models selected by participants for fairness related subtasks. Each mean is annotated with the 95% two-sided confidence interval.

| Tool | Average Odds (lower is better) | Statistical Parity (lower is better) | Equal Opportunity (lower is better) | Disparate Impact (higher is better) | Accuracy (higher is better) |
|---|---|---|---|---|---|
| scikit-learn | $0.163 \pm 0.096$ | $0.213 \pm 0.110$ | $0.154 \pm 0.094$ | $0.570 \pm 0.129$ | $0.734 \pm 0.115$ |
| AI Fairness 360 | $0.079 \pm 0.070$ | $0.200 \pm 0.104$ | $0.061 \pm 0.062$ | $0.814 \pm 0.101$ | $0.816 \pm 0.101$ |
| fairkit-learn | $0.116 \pm 0.083$ | $0.225 \pm 0.108$ | $0.093 \pm 0.075$ | $0.725 \pm 0.116$ | $0.788 \pm 0.106$ |

**Fig. 4.** Mean fairness scores (across all three tasks) of models selected by participants for fairness subtasks. Each mean is annotated with the 95% two-sided confidence interval.

| Tool | Average Odds (lower is better) | Statistical Parity (lower is better) | Equal Opportunity (lower is better) | Disparate Impact (higher is better) | Accuracy (higher is better) |
|---|---|---|---|---|---|
| scikit-learn | $0.190 \pm 0.102$ | $0.242 \pm 0.111$ | $0.171 \pm 0.098$ | $0.533 \pm 0.130$ | $0.747 \pm 0.113$ |
| AI Fairness 360 | $0.097 \pm 0.077$ | $0.258 \pm 0.114$ | $0.076 \pm 0.069$ | $0.820 \pm 0.10$ | $0.822 \pm 0.100$ |
| fairkit-learn | $0.086 \pm 0.073$ | $0.229 \pm 0.109$ | $0.070 \pm 0.066$ | $0.829 \pm 0.098$ | $0.815 \pm 0.101$ |

**Fig. 5.** Mean fairness scores (across all three tasks) of models selected by participants for balancing fairness and accuracy subtasks. Each mean is annotated with the 95% two-sided confidence interval.

AI Fairness 360 can generally find more fair models than fairkit-learn and scikit-learn (Fig. 4). When trying to balance fairness and accuracy, fairkit-learn is capable of finding models that are high performing and generally more fair than models found by AI Fairness 360 and scikit-learn (Fig. 5).

When evaluating fairness without using tools designed to do so, our study found that engineers have different ways of reasoning about model fairness ranging from "educated guesses" to implementing their own fairness metrics. Most often, participants use accuracy as some proxy for fairness. However our data suggests engineers may have different ideas of the relationship between accuracy and fairness.

### 4.1. $H_1$: Out-of-the-box model fairness

One of the most commonly used toolkits for machine learning is scikit-learn, and it is not clear how often engineers modify default parameters (Kim et al., 2017; Sanders and Giraud-Carrier, 2017). One of the main advantages to using fairkit-learn is that it provides easy-to-use support searching different hyperparameter configurations, and exploring those configurations' effects on model quality and fairness. As shown in Fig. 3, participant models selected for fairness related subtasks while using fairkit-learn outperformed all scikit-learn default models with respect to all metrics considered. We see even more improvement when participants used fairkit-learn to find models that best balance between fairness and accuracy, with the differences as high as 0.30 between fairkit-learn and scikit-learn model fairness scores. These fairness improvements came with no sacrifice to accuracy; in fact, we see an increase of 0.04–0.07 in accuracy score.

> Our findings suggest that when using fairkit-learn, engineers can find models that are more fair and accurate than out-of-the-box, default scikit-learn models.

### 4.2. $H_2$: More fair models with fairkit-learn

One of the goals of fairkit-learn is to help engineers find the fairest models possible. When asked to find a model that will be the most fair, overall the models selected by participants using fairkit-learn were more fair than models selected using scikit-learn. While comparable, the models selected when using AI Fairness 360 were generally more fair than models selected when using fairkit-learn. Figure 4 shows the average fairness scores by metric across tasks by tool. While one of the goals of fairkit-learn is to support finding fair models, the primary goal is to support balancing fairness with other quality concerns. Given the primary goal of AI Fairness 360 is to support training fair models, these findings are not surprising.

> Our findings suggest that when using fairkit-learn, engineers can find models that are comparable in fairness to those found using AI Fairness 360 and more fair than models found using scikit-learn.

### 4.3. $H_3$: Fair and accurate models with fairkit-learn

While fairkit-learn wants to help engineers improve the fairness of their models, the primary goal of our toolkit is to support engineers when attempting to balance fairness with other quality metrics, such as accuracy. Figure 5 shows the average fairness by metric and accuracy of models selected as the most balanced across tasks by tool. When asked to find a model that will best balance fairness and accuracy, participants selected models that were more fair with almost no loss with respect to accuracy across all three tasks when using fairkit-learn.

Of course, the fairness-quality trade-off depends on the dataset used to train the models (see Section ). Thus, we expect that for some datasets, more improvement in fairness is possible, and fairkit-learn should be able to enable that improvement. We observed that for Task 3, which

uses the adult census income dataset, the users demonstrated a greater improvement with respect to fairness when using fairkit-learn than when using scikit-learn. For that dataset, fairkit-learn models were more fair as well as more accurate than AI Fairness 360 and scikit-learn models.

> Our findings suggest that engineers find more fair models of high quality when using fairkit-learn than when using AI Fairness 360 or scikit-learn. When using fairkit-learn, the balanced models are comparably accurate, and sometimes more accurate, compared to using AI Fairness 360 or scikit-learn.

### 4.4. Evaluating fairness without fairness tools

We expect that engineers would be able to at least reason about fairness when using tools like fairkit-learn and AI Fairness 360, given they provide functionality for doing so. It is less obvious how engineers handle fairness when using tools like scikit-learn that do not provide functionality for training or evaluating fair models.

For four participants, lack of immediate or easy access to fairness tooling rendered them either unable to find a model they felt would be fair or unable to reason about why a given model should be considered fair. When asked to select a model they felt would be most fair while using scikit-learn, all participants selected a model. However, the ability to explain why they selected that model over others varied. For Task 1, two participants could not figure out how to reason about the fairness of the model they selected. P47, rather than using some metric or resource to reason about the fairness of their model, put "not applicable."

For those who did try to reason about the fairness of their models, participants used various (and sometimes contradictory) ways of evaluating the fairness of a given model. Only four participants used what would be considered fairness metrics to evaluate their models. Two participants used another fairness tool, FairML (Adebayo, 2016), to evaluate model fairness. Seven participants created their own metric to evaluate model fairness.

The majority of participants that used a metric (21 out of 25) used model accuracy, or some related metric, to evaluate fairness. However, participants were split on whether higher accuracy was an indicator of being more fair or if lower accuracy was a better indicator. Eight out of 21 participants that used accuracy reported selecting a given model because it has the highest accuracy. But three out of the 21 participants felt that *lower* accuracy meant a model was more likely to be fair. Those who opted for higher accuracy noted that they felt higher accuracy meant a model would handle bias better, while those who took the lower accuracy route noted they felt accuracy had to be sacrificed to help guarantee fairness.

Fifteen participants cited making an "educated guess" regarding model fairness. Participants backed their educated guesses with the accuracy score, outside resources, background knowledge of machine learning models and how they work, or some combination of the three. Often accuracy was coupled with some other metric or explanation for model selection but some participants made decisions on the fairness of their models without using any metrics. For two of 15 educated guesses, the decision was based on the models they evaluated using fairkit-learn or AI Fairness 360 in previous tasks. For another two participants, the model that was able to achieve the highest accuracy in the the shortest amount of time was considered to be the most fair. Participants also made assumptions about the dataset, such as how well distributed it is, to determine model fairness.

> Our findings suggest that engineers sometimes struggle with reasoning about fairness without the proper tooling. Engineers have different, sometimes contradictory, ways of reasoning about fairness when asked to do so, often using accuracy as a proxy for fairness, despite clear evidence that those metrics are often at odds.

### 4.5. Threats to validity

*External validity* Our study compared fairkit-learn to a subset of the tools available for training and evaluating machine learning models. To increase the generalizability of our findings, we selected tools that are considered to be state-of-the-art and supported by industry practices.

We derived tasks to evaluate our tool, however, the tasks and results may not be representative of what real engineers do and the models they would build. To mitigate the effects of this threat, we provided participants with real datasets that have been used in research and practice. We also asked participants to evaluate bias against attributes that practitioners would care about (e.g., race).

While it may be seen as a threat to validity that we used students as participants, previous research suggests student behavior when completing programming tasks is often not far from that of professionals (Höst et al., 2000; Naiakshina et al., 2019). Given the students in our sample were studying related topics, this heightens the potential for our findings to generalize to practicing engineers.

*Internal validity* Our user study participants were students completing the study as a homework assignment. This leads to the potential for selection bias. The course included a diverse set of participants and we randomly assigned the task and tool ordering across participants, somewhat mitigating this threat.

The design of our study had participants complete the same tasks for each tool, which introduces the potential for testing bias. To minimize this threat, we had participants use a different dataset for each task which meant different approaches would be needed to meet the goals for each task.

*Construct validity* Given the technical and time requirements for completing the study, one issue we encountered was the effect of technical difficulties and time management of participants on data collection. We included various safeguards for keeping track of participants' contributions and minimizing the effects of technical difficulties on study completion and data collection.

### 4.6. Discussion

Our findings suggest that tools like fairkit-learn and AI Fairness 360 can help engineers and software engineers find fair models over tools such as scikit-learn (that are not designed for this purpose). Further, we find that while AI Fairness 360 may be better for focusing on fairness alone, fairkit-learn is able to help engineers find the best balance between fairness with other quality concerns, such as accuracy. Still, while fairkit-learn was helpful in many situations, that was not always the case. Our conclusions observe that fairkit-learn can help engineers understand the fairness landscape and select more fair models in many situations, but that it is not an end-all solution to this complex problem, and further research into the situations where our study found fairkit-learn to be less effective is warranted. This section discusses the implications of our findings.

#### 4.6.1. Using Pareto optimality for balanced models

One of the primary contributions of fairkit-learn is the ability to search a large space of models and return results from only the best or optimal models with respect to the relevant metrics. This is done by calculating the Pareto optimal set of models from a given set of evaluated models and metrics. Rather than engineers having to do their own coding and math to compute a large number of models to get a sense for where the best balance lies, data scientists can use the grid search provided by fairkit-learn to find fair models and understand the fairness-quality trade-offs.

While the notion of a Pareto-optimal set of models can be useful for finding the most fair models, our study suggests it is most useful for trying to find models that balance more than one metric. In the case of our study, we wanted to balance accuracy and fairness. But in the real world, data scientists may have other metrics they want to balance, including

multiple quality metrics. Fairkit-learn can help engineers find models that are Pareto optimal for a given set of metrics, thereby increasing the possibility of improving the models used for a given dataset or task.

#### 4.6.2. Understanding relationships between model fairness and quality

Intuitively, adding extra constraints on machine learning algorithms should only constrain the solutions they return. Thus, asking an algorithm to learn a model that maximizes accuracy should always produce a more-accurate model than when asking an algorithm to maximize accuracy while also adhering to a fairness constraint. The same intuition, theoretically, holds for a user attempting to select a model — when faced with extra constraints, the user should only find less-accurate models. However, in our experiments, we observed that when given the tools to measure and enforce fairness, users often returned not only more fair but also more accurate models. While observing accuracy improvements when enforcing fairness may not be the most common outcome, because we observed it in our evaluation, we suggest three possible reasons this could happen. This list is not intended to be exhaustive.

First, users do not typically report optimal solutions. The space of possible models is enormous, and users aim to produce a high-accuracy model, but, typically, finding the absolute most accurate model is not feasible. As a result, it is theoretically possible to find a more accurate model if one looks harder. But that's precisely what having an extra constraint forces the user to do — look harder. When a user is prompted to consider fairness and given the tools to do so, they are likely to explore more models, more different learning algorithms, and more hyperparameter values. As a result, they are likely to come across more accurate models during their search, and if some of those happen to also be more fair, the models the users chose would then improve both accuracy and fairness.

Second, the above reasoning is not only true for users, but also for machine learning algorithms. Such algorithms are not optimal-solution seekers. Again, the search space of possible solution models is typically too large, and machine learning algorithms simply attempt to find a model that *tries* to minimize a particular loss function (e.g., inaccuracy), but provide no guarantees that the trained model is optimal. In real-world datasets, adding an extra constraint, such as optimizing for fairness, can force the algorithm to consider more possible solutions. Considering more possible solutions, can, in turn, result in finding a more accurate model. Thus, even if the user does not explicitly consider more solutions when asked to consider fairness in training models, the underlying algorithms are likely to still do so, thus potentially finding solutions that improve both fairness and accuracy.

Third, the assumption that fairness and quality are at odds with one another is not always correct. It is certainly possible to design a dataset for which learning a more fair model necessitates that the model is less accurate. Imagine, for example, a dataset of loan applicants and the bank's loan decisions in which all men get loans and no women get loans. When training on such a biased dataset, accuracy and fairness are at odds. However, in real-world datasets, fairness and quality often complement each other (Thomas et al., 2019) because the bias in data can be an artifact of real-world discrimination. For example, a bank not giving loans to women or Black applicants because of their gender or race can result in not giving loans to people who can actually repay them. Thus, depending on how accuracy is measured, training more fair models on such datasets can improve real-world accuracy by removing the bias that led to not only unfair but also inaccurate decisions.

#### 4.6.3. The importance of fairness tools

Findings from our study suggest that even without fairness machine learning tools, engineers will sometimes try to reason about the fairness of machine learning models, often making improper assumptions, leading to poor reasoning. On the one hand, the first step to training fair models is thinking about model fairness while training it. On the other hand, one can argue that ad hoc rationale for model fairness is no better (if not worse than) not evaluating for fairness at all.

Although some participants found reasonably fair models when using scikit-learn, more often than not, larger sacrifices had to be made (either in terms of fairness, or accuracy, or both) when trying to find a well-balanced model using scikit-learn than when using fairkit-learn and AI Fairness 360. There was also much more inconsistency behind the rationale for selecting fair or well-balanced models, which can lead to uncertainty regarding how fair or unfair a model really is. Our results shed a light on the importance of using tools that support fair model training and evaluation.

#### 4.6.4. Misconceptions regarding model fairness

When tasked with evaluating model fairness and not equipped with fairness machine learning tools, participants in our study made various assumptions to reason about the fairness of a given model. Some of the assumptions participants made contradicted others, such as the relationship between accuracy and fairness. Many participants used accuracy as a proxy for fairness, even though in the datasets available to them, accuracy and fairness metrics are opposing forces: increasing one typically decreases the other. Our data suggest that there may be misconceptions engineers have regarding what it means for a model to be fair, and, when not armed with proper tools, make incorrect assumptions and use those assumptions in evaluating fairness. One reason this discrepancy may exist is due to the large (and growing) number of ways one can mitigate and measure model fairness. An important step to training fair models is understanding what it means for a model to be fair in a given context and what factors may affect overall fairness or quality of a given model. Typically, domain experts are the ones who understand the fairness requirements of their domain, and adequately communicating these requirements to the engineers building the system is critical.

#### 4.6.5. Qualitative comparisons

While AI Fairness 360 can be used to produce visualizations, the built-in visualizations significantly limit their support for engineers reasoning about the interactions between model fairness and quality. As an example, Fig. 6 shows a screenshot from an online tutorial of AI Fairness 360 applied to the COMPAS recidivism dataset. Here, AI Fairness 360 first trains models using fairness-unaware algorithms and computes five fairness metrics of the resulting models, and then applies a data-reweighing algorithm for bias-mitigation and recomputes the five metrics on the resulting models. The five graphs, one per definition, shows the resulting fairness metric scores for each of the definitions for the two models, showing that for all five metrics, the mitigation brought them within acceptable levels.

While such visualizations allow the user to understand the impact of an approach on metrics, they, unlike fairkit-learn's visualizations (recall Fig. 1), fail to support informing decisions reasoning about the overall landscape of the quality versus fairness trade-off in four important ways:

First, these AI Fairness 360 visualizations do not display model fairness and quality metrics on the same graph (although it is possible to place the two separate graphs onto the same plot), failing to support understanding how improvement in each each fairness metric affects accuracy, or other quality metrics. For example, in Fig. 6, the improvements in fairness caused by the data-reweighing mitigation strategy might significantly reduce the model's accuracy (or it might not), but the visualization fails to convey that information. By contrast, fairkit-learn visualizes the relationship between any two user-selected metrics, including one fairness and one quality metric at a time (see Fig. 1), supporting such understanding and reasoning about the effects of mitigation strategies, data and model procesessing procedures, and learning algorithms have on these metrics.

Second, fairkit-learn visualizations allow for comparing multiple fairness metrics directly, on a single graph, while AI Fairness 360 relies on side-by-side, separate graphs, one per metric. Fairkit-learn's visualizations, thus, support understanding the relationships between fairness metrics, which can be quite complex, including situations where improving some metrics necessarily hurts the others, and situations where


**Fig. 6.** A screenshot of AI Fairness 360 visualization from an online tutorial (https://aif360.mybluemix.net/data/) of a reweighing algorithm bias-mitigation strategy applied to the COMPAS recidivism dataset.

improving some metrics helps the others (Friedler et al., 2016; Kleinberg et al., 2017).

Third, a single fairkit-learn visualization can compare multiple bias-mitigation strategies, demonstrating, for example, whether some strategies are completely dominated by others with respect with user-selected fairness and quality metrics, or whether some strategies perform better when model quality is the top concern, but others when model fairness is. AI Fairness 360 visualizes one mitigation strategy (and one fairness definition) per graph, failing to support the needed comparisons.

Forth, by computing the Patero-optimal frontier of solutions with respect to an arbitrary number of metrics (recall that while the two-dimensional visualization only displays the comparison across two metrics at once, the user can select more metrics for computing the Pareto-optimal solutions that are displayed), fairkit-learn visualizations (Fig. 1) support understanding the fundamental trade-offs between quality and fairness metrics within the dataset that persevere despite applying multiple mitigation strategies, data and model procesessing procedures, and learning algorithms.

The scikit-learn toolkit also allows for some basic visualizations that are similar to the AI Fairness 360 ones, but without explicitly encoding fairness metrics, further complicating using scikit-learn for understanding models' effects on fairness. Importantly, AI Fairness 360, scikit-learn, and fairkit-learn are all extendable frameworks. That is, developers can create new visualizations on top of ones that already exist. In fact, fairkit-learn is simply code written on top of scikit-learn, so, in the extreme, a developer could reimplement fairkit-learn's functionality and enable all the visualization comparisons with scikit-learn. Our comparisons here appropriately focus on the functionality already included in the three toolkits.

## 5. Related work

We now place fairkit-learn in the context of related contributions on algorithmic fairness and support for evaluating system fairness and performance.

### 5.1. Perceptions on algorithmic fairness

While it is important to explore algorithmic fairness when considering the use of machine learning models, some research has explored how end-users understand and perceive the notion of algorithmic fairness (Grgic-Hlaca et al., 2018; Warshaw et al., 2016; Woodruff et al., 2018).

Warsaw and colleagues interviewed 21 high school educations individuals on their beliefs and misconceptions regarding how companies collect data and make inferences about them using that data (Warshaw et al., 2016). They found that most participants believed companies make decisions either based largely on stereotypes or based on online behaviors and intuition.

Similarly, Woodruff and colleagues conducted an interview study with 44 traditionally marginalized individuals on how they feel about algorithmic fairness (Woodruff et al., 2018). When provided with a description of what it meant for algorithms to be unfair, participants expressed concerns regarding the implications of algorithmic (un)fairness. They also found that participants expected companies to address these sorts of things, regardless of the cause.

Grgió-Hlača et al. propose an explanatory framework to understand the features people consider fair or unfair to use in decision-making and why (Grgic-Hlaca et al., 2018). They deployed a series of scenario-based surveys, developed based on their framework, and found that they can accurately predict features that would be deemed fair to use.

While these studies help us understand perceptions of algorithmic fairness and what features may affect the public's perception of a given software's fairness, it does not help engineers make informed decisions that can ensure, for example, that the proper features are being considered while also providing a high performing system. In contrast, fairkit-learn empowers engineers to explore the space of possible models, with regard to the features and metrics they care about, such that they can better ensure algorithmic fairness.

Finally, transferring academic technology to industry always poses some challenges, but initial attempts for transitioning machine learning fairness evaluation technology in the banking industry shows promise (Castelnovo et al., 2020). This suggests some hope for the use of fairkit-learn, and other similar tools, in industry.

### 5.2. Evaluating model fairness & performance

Typically, machine learning model performance is evaluated using metrics pertaining to the accuracy of that model. scikit-learn (Pedregosa et al., 2011) is one of the most common tools used for training and evaluating machine learning models. scikit-learn is an open source Python module that provides engineers with a variety of machine learning algorithms and various metrics for evaluating models for performance, though no fairness metrics. It is designed to be easy to use and accessible to non-specialists. While scikit-learn is useful for training and evaluating models based on their performance, there is no built in functionality for measuring model fairness or mitigating bias.

There exist tools designed to help engineers reason about fairness in their machine learning models (Adebayo, 2016; Bellamy et al., 2018; Wexler, 2018). FairML helps engineers avoid unintentional discrimination by automatically determining relative significance of model inputs to that model's predictions, allowing engineers to more easily audit predictive models. Meanwhile, Fairway combines pre-processing and in-processing methods to remove bias from training data and models (Chakraborty et al., 2020). Meanwhile other tools can mitigate (Hort et al., 2021) or repair (Sun et al., 2022) bias in models by altering their behavior.

FairPrep helps data scientists follow best practices in software engineering and machine learning to develop models according to the scientists' needs (Schelter et al., 2020). FairRover helps scientists explore the trade-offs that result from use of machine learning models, focusing on responsible and ethical uses (Zhang et al., 2021a).

Google developed the What-If Tool to help programmers and non-programmers analyze and understand machine learning models without writing code (Wexler, 2018). Provided a TensorFlow model and a dataset, the What-If Tool allows you to visualize the dataset, edit individuals in the dataset and see the effects, perform counterfactual analysis, and evaluate models based on performance and fairness.

Similar to the What-If Tool is AI Fairness 360, a Python tool suite for mitigating bias and evaluating models for fairness and performance (Bellamy et al., 2018). The package includes fairness metrics, metric explanations, and bias mitigation algorithms for datasets and models. AI Fairness 360 is designed to be extensible and accessible to data scientists and practitioners.

FairVis is a visual analytics system that supports exploring fairness and performance with respect to certain subgroups in a dataset (Cabrera et al., 2019). Its focus is auditing trained model performance on data from these subgroups. FairVis supports directly exploring a dataset and the labels produced by an externally trained model, whereas fairkit-learn focuses on training models and visualizing learned model behavior. Users could make inferences about the dataset using fairkit-learn, but that process would be less direct, for example, generalizing from common behavior of multiple learned models. FairVis allows users to specify custom subgroups and explore a set of ten metrics, one metric per plot. Unlike FairVis, fairkit-learn supports training models using fairness-aware and fairness-unaware methods, supports more metrics, and allows the user to specify custom metrics. To compare model behavior, FairVis requires training models and applying those models to data externally. The features of these two tools are complementary and can likely both help users understand the fairness and accuracy properties of data and of learned models. While FairVis had not been evaluated directly with users via a controlled study (Cabrera et al., 2019), and our controlled user study does not include FairVis, future work on combining features of the two tools and evaluating their effect on users is likely to produce fruitful results.

While there exists tools that can help engineers evaluate model fairness and performance, fairkit-learn works with existing tools to help engineers find Pareto-optimal models that balance fairness and performance and a visualization that makes it quicker and easier to explore the effects of different model configurations.

### 5.3. Training fair models

Machine learning approaches that aim to train fair models even when using biased training data fall into three primary categories: (1) data transformation (perturbing input data to quantify bias in data) (2) algorithm manipulation (modifying the machine learning cost function typically by adding fairness constraints or regularization), and (3) outcome manipulation (balancing the outcome across multiple groups). Dwork et al. formulate fairness as an optimization problem that can be solved by a linear program (Dwork et al., 2012). They minimize a loss function while achieving a Lipschitz property for a defined similarity metric between two individuals and then they analyze when this local fairness constraint implies statistical parity. Corbett-Davies et al. reformulate algorithmic fairness as constrained optimization with their fairness definitions as constraints (Corbett-Davies et al., 2017). Meanwhile Zhang et al. use adversarial learning as a means for finding fair models (Zhang et al., 2018). Zafar et al. define a measure of decision boundary fairness: the covariance between sensitive (protected) attributes and the signed distance between the subjects' feature vectors and the decision boundary of a classifier (Zafar et al., 2017b). They take two different constrained optimization approaches: (1) maximizing accuracy subject to fairness constraints and (2) maximizing fairness subject to accuracy constraints. Kamishima et al. express fairness regularization as a function of the data and logistic regression model weights and then they optimize the set of weights using standard conjugate gradient methods (Kamishima et al., 2012). Their proposed fairness regularization is differentiable and smooth, thus enabling gradient descent or second order optimization methods. Thomas et al. introduce the Seldonian Framework for designing machine learning algorithms that perform a one-time safety-check to produce models that are probabilistically guaranteed to satisfy fairness and safety constraints, even when applied to unseen data (Thomas et al., 2019). Users of algorithms within the Seldonian Framework can apply a large number of fairness and safety constraints, including multiple simultaneously. Metevier et al. then demonstrate contextual bandit algorithms within the Seldonian Framework (Metevier et al., 2019), which can satisfy delayed impact constraints (Liu et al., 2018). While designing novel classification techniques that explicitly optimize for fairness has shown great promise, fairkit-learn tackles a related by different problem of helping data scientists understand the quality-fairness trade-offs and make decisions about which fairness definitions to use and which models to select in their specific domains. Models can be trained to respect fairness definitions even when the data to which the models are applied come from a different distribution than the training data (Giguere et al., 2022).

Model cards can accompany trained machine learning models to inform users of benchmark evaluations in certain conditions, which can both disclose the intended use context and warn users of possible misuses of models (Mitchell et al., 2018). By contrast, fairkit-learn can produce benchmark results when model cards are not available, and help uses see fairness metrics and other parameters relevant to their application domain, which the algorithm's designers may not have considered a priori.

### 5.4. Testing for fairness

In contrast to correcting for fairness explicitly, there exist a number of open-source software systems that *test* for fairness. Galhotra et al. present Themis, a system that automatically generates test suites to measure a (1) a causal definition of fairness (if two individuals differ in only a single protected attribute then the system recommendation is the same) and (2) group fairness (Galhotra et al., 2017). By biasing the search mechanisms used in such testing, it is possible to find examples of bias more efficiently (Soremekun et al., 2022; Udeshi et al., 2018; Zhang et al., 2021b; 2020), potentially useful for debugging or bias repair, though not for measuring bias frequency. FairTest discovers bias bugs, tests systems for discrimination, and conducts error profiling of machine learning algorithms (Tramer et al., 2017), but does not help remove bias. FairML, an iterative orthogonal transformation process, aims to remove the effect of a given attribute from a dataset (Adebayo and Kagal, 2016), which creates variants of datasets, which then need to be explored by tools such as fairkit-learn, which, in turn, helps explore the entire space of model configurations and find the ones that satisfy fairness conditions.

## 6. Contributions

We have presented fairkit-learn, a novel open-source toolkit designed to support engineers in training fair machine learning models. A

controlled user study showed that students using fairkit-learn produced models that provided a better balance between fairness and accuracy than students using state-of-the-art tools scikit-learn and IBM AI Fairness 360. Exploring how engineers approach evaluating fairness when fairness tools are not available, we found that they struggle, and often default to using quality metrics, such as accuracy, as a proxy for fairness (despite the fact that these metrics are often at odds with fairness). Overall, fairkit-learn is an effective tool for helping engineers understand the fairness-quality landscape, and our user study shows promising results, suggesting that further work improving and evaluating fairkit-learn with industrial engineers is worthwhile.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work is supported by the National Science Foundation under grant no. CCF-1763423, and by Google.

## References

Adebayo, J., Kagal, L., 2016. Iterative orthogonal feature projection for diagnosing bias in black-box models. CoRR. http://arxiv.org/abs/1611.04967

Adebayo, J.A., 2016. FairML: ToolBox for diagnosing bias in predictive modeling. Massachusetts Institute of Technology Ph.D. thesis.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., Wallach, H., 2018. A reductions approach to fair classification. In: International Conference on Machine Learning (ICML). Stockholm, Sweden

Albrecht, J.P., 2016. How the GDPR will change the world. Eur. Data Prot. Law Rev. 2, 287.

Angwin, J., Larson, J., Mattu, S., Kirchner, L., 2016. Machine Bias. ProPublica. May 23, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Barocas, S., 2018. Accounting for artificial intelligence: rules, reasons, rationales. Human Rights, Ethics, and Artificial Intelligence.

Bellamy, R.K.E., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilovic, A., Nagar, S., Ramamurthy, K.N., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K.R., Zhang, Y., 2018. AI Fairness 360: an extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. CoRR. https://arxiv.org/abs/1810.01943

Berk, R., Heidari, H., Jabbari, S., Kearns, M., Roth, A., 2018. Fairness in criminal justice risk assessments: the state of the art. Sociol. Methods Res. doi:10.1177/0049124118782533.

Bird, S., Dudík, M., Edgar, R., Horn, B., Lutz, R., Milan, V., Sameki, M., Wallach, H., Walker, K., 2020. Fairlearn: A Toolkit for Assessing and Improving Fairness in AI. Technical Report MSR-TR-2020-32. Microsoft. https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/

Blodgett, S.L., O'Connor, B., 2017. Racial disparity in natural language processing: a case study of social media african-american english. Fairness, Accountability, and Transparency in Machine Learning (FAT/ML). Halifax, NS, Canada

Buolamwini, J., Gebru, T., 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In: Proceedings of the 1st Conference on Fairness, Accountability and Transparency. PMLR, New York, NY, USA, pp. 77–91. http://proceedings.mlr.press/v81/buolamwini18a.html

Cabrera, Á.A., Epperson, W., Hohman, F., Kahng, M., Morgenstern, J., Chau, D.H., 2019. FairVis: visual analytics for discovering intersectional bias in machine learning. In: IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 46–56.

Calders, T., Verwer, S., 2010. Three naive Bayes approaches for discrimination-free classification. Data Min. Knowl. Discov. 21 (2), 277–292. doi:10.1007/s10618-010-0190-x.

Caliskan, A., Bryson, J.J., Narayanan, A., 2017. Semantics derived automatically from language corpora contain human-like biases. Science 356 (6334), 183–186. doi:10.1126/science.aal4230.

Castelnovo, A., Crupi, R., Gamba, G.D., Greco, G., Naseer, A., Regoli, D., Gonzalez, B.S.M., 2020. BeFair: addressing fairness in the banking sector. In: International Conference on Big Data (Big Data) doi:10.1109/BigData50022.2020.9377894.

Chakraborty, J., Majumder, S., Yu, Z., Menzies, T., 2020. Fairway: a way to build fair ML software. In: ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE).

Chouldechova, A., 2017. Fair prediction with disparate impact: a study of bias in recidivism prediction instruments. Big Data 5 (2), 153–163.

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

D'Amour, A., Srinivasan, H., Atwood, J., Baljekar, P., Sculley, D., Halpern, Y., 2020. Fairness is not static: deeper understanding of long term fairness via simulation studies. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT), pp. 525–534. doi:10.1145/3351095.3372878. Barcelona, Spain

de Blasio, B., 2018. Mayor de Blasio announces first-in-nation task force to examine automated decision systems used by the city. https://tinyurl.com/y4s2623o/.

Doshi-Velez, F., Kim, B., 2017. Towards a rigorous science of interpretable machine learning. CoRR. https://arxiv.org/abs/1702.08608

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

Executive Office of the President, 2016. Big data: A report on algorithmic systems, opportunity, and civil rights. May, https://www.whitehouse.gov/sites/default/files/microsites/ostp/2016_0504_data_discrimination.pdf.

Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., 2016. On the (im)possibility of fairness. CoRR. http://arxiv.org/abs/1609.07236

Galhotra, S., Brun, Y., Meliou, A., 2017. Fairness testing: testing software for discrimination. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, pp. 498–510.

Giguere, S., Metevier, B., Brun, Y., da Silva, B.C., Thomas, P.S., Niekum, S., 2022. Fairness guarantees under demographic shift. In: International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=wbPObLm6ueA

Goodall, N.J., 2016. Can you program ethics into a self-driving car? IEEE Spectr. 53 (6), 28–58. doi:10.1109/MSPEC.2016.7473149.

Grgic-Hlaca, N., Redmiles, E.M., Gummadi, K.P., Weller, A., 2018. Human perceptions of fairness in algorithmic decision making: a case study of criminal risk prediction. In: Proceedings of the 2018 World Wide Web Conference. International World Wide Web Conferences Steering Committee, pp. 903–912.

Griggs v. Duke Power Co., 1971. 401 U.S. 424. https://supreme.justia.com/cases/federal/us/401/424/.

Hardt, M., Price, E., Srebro, N., 2016. Equality of opportunity in supervised learning. In: Annual Conference on Neural Information Processing Systems (NIPS). Barcelona, Spain

Hashimoto, T.B., Srivastava, M., Namkoong, H., Liang, P., 2018. Fairness without demographics in repeated loss minimization. In: International Conference on Machine Learning (ICML). Stockholm, Sweden

Haweawar, D., 2012. Staples, Home Depot, and Other Online Stores Change Prices Based on Your Location. VentureBeat. December 24, https://venturebeat.com/2012/12/24/staples-online-stores-price-changes

Holstein, K., Vaughan, J.W., III, H.D., Dudik, M., Wallach, H., 2019. Improving fairness in machine learning systems: what do industry practitioners need? In: SIGCHI Conference on Human Factors in Computing Systems (CHI), pp. 600:1–600:16. doi:10.1145/3290605.3300830. Glasgow, Scotland, UK

Hort, M., Zhang, J.M., Sarro, F., Harman, M., 2021. Fairea: a model behaviour mutation approach to benchmarking bias mitigation methods. In: European Software Engineering Conference and ACM SIGSOFT International Symposium on Foundations of Software Engineering (ESEC/FSE), pp. 994–1006. doi:10.1145/3468264.3468565. Athens, Greece

Höst, M., Regnell, B., Wohlin, C., 2000. Using students as subjects–a comparative study of students and professionals in lead-time impact assessment. Empir. Softw. Eng. 5 (3), 201–214.

IBM, 2019. AI Fairness 360 Open Source Toolkit. https://aif360.mybluemix.net.

Ingold, D., Soper, S., 2016. Amazon Doesn't Consider the Race of Its Customers. Should It?. Bloomberg. April 21, http://www.bloomberg.com/graphics/2016-amazon-same-day

Johnson, B., Brun, Y., 2022. Fairkit-learn: a fairness evaluation and comparison toolkit. In: International Conference on Software Engineering (ICSE) Demo track doi:10.1145/3510454.3516830. Pittsburgh, PA, USA

Kamishima, T., Akaho, S., Asoh, H., Sakuma, J., 2012. Fairness-aware classifier with prejudice remover regularizer. In: Flach, P.A., Bie, T.D., Cristianini, N. (Eds.), ECML/PKDD (2). Springer, pp. 35–50.

Kim, M., Zimmermann, T., DeLine, R., Begel, A., 2017. Data scientists in software teams: State of the art and challenges. IEEE Trans. Softw. Eng. 44 (11), 1024–1038.

Klare, B.F., Burge, M.J., Klontz, J.C., Bruegge, R.W.V., Jain, A.K., 2012. Face recognition performance: Role of demographic information. IEEE Trans. Inf. ForensicsSecur. (TIFS) 7 (6), 1789–1801. doi:10.1109/TIFS.2012.2214212.

Kleinberg, J.M., Mullainathan, S., Raghavan, M., 2017. Inherent trade-offs in the fair determination of risk scores. In: Innovations in Theoretical Computer Science Conference (ITCS), Vol. 67, pp. 43:1–43:23. doi:10.4230/LIPIcs.ITCS.2017.43. Berkeley, CA, USA

Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., Toups, C., Rickford, J.R., Jurafsky, D., Goel, S., 2020. Racial disparities in automated speech recognition. Proc. Natl. Acad. Sci. 117 (14), 7684–7689. doi:10.1073/pnas.1915768117.

Kusner, M.J., Loftus, J.R., Russell, C., Silva, R., 2017. Counterfactual fairness. In: Annual Conference on Neural Information Processing Systems (NIPS). Long Beach, CA, USA

Lee, M.S.A., Singh, J., 2021. The landscape and gaps in open source fairness toolkits. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–13.

Letzter, R., 2016. Amazon Just Showed US That 'Unbiased' Algorithms Can Be Inadvertently Racist. TECH Insider. April 21, http://www.techinsider.io/how-algorithms-can-be-racist-2016-4

Liu, L.T., Dean, S., Rolf, E., Simchowitz, M., Hardt, M., 2018. Delayed impact of fair machine learning. In: International Conference on Machine Learning (ICML), Vol. 80. PMLR, pp. 3150–3158.

Makhlouf, K., Zhioua, S., Palamidessi, C., 2021. On the applicability of machine learning fairness notions. ACM SIGKDD Explor. Newsl. 23 (1), 14–23. doi:10.1145/3468507.3468511.

Mattioli, D., 2012. On Orbitz, Mac users steered to pricier hotels. Wall Street J.. August 23, http://www.wsj.com/articles/SB10001424052702304458604577488822667325882

Metevier, B., Giguere, S., Brockman, S., Kobren, A., Brun, Y., Brunskill, E., Thomas, P., 2019. Offline contextual bandits with high probability fairness guarantees. In: 33rd Annual Conference on Neural Information Processing Systems (NeurIPS), Advances in Neural Information Processing Systems 32, pp. 14893–14904. Vancouver, BC, Canada

Mikians, J., Gyarmati, L., Erramilli, V., Laoutaris, N., 2012. Detecting price and search discrimination on the Internet. In: ACM Workshop on Hot Topics in Networks (HotNets), pp. 79–84. doi:10.1145/2390231.2390245. Redmond, Washington

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I.D., Gebru, T., 2018. Model cards for model reporting. In: ACM Conference on Fairness, Accountability, and Transparency (FAT*).

Ml-fairness-gym. https://github.com/google/ml-fairness-gym.

Naiakshina, A., Danilova, A., Gerlitz, E., Von Zezschwitz, E., Smith, M., 2019. "If you want, I can store the encrypted password" a password-storage field study with freelance developers. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–12.

Narayanan, A., 2018. 21 Fairness definitions and their politics. Tutorial at the Conference on Fairness, Accountability, and Transparency.

Olson, P., 2011. The Algorithm That Beats Your Bank Manager. CNN Money. March 15, https://tinyurl.com/h968txt

Pearl, J., 2009. Causal inference in statistics: an overview. Stat. Surv. 3, 96–146. doi:10.1214/09-SS057.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Peng, T., 2019. Humans don't realize how biased they are until AI reproduces the same bias, says UNESCO AI chair. https://tinyurl.com/y5jxadg6/.

ProPublica, 2019. COMPAS recidivism risk score data and analysis. https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis/.

Raghavan, M., Barocas, S., Kleinberg, J., Levy, K., 2019. Mitigating bias in algorithmic hiring: Evaluating claims and practices. CoRR. https://arxiv.org/abs/1906.09208

Rothblum, G.N., Yona, G., 2018. Probably approximately metric-fair learning. In: International Conference on Machine Learning (ICML).

Sanders, S., Giraud-Carrier, C., 2017. Informing the use of hyperparameter optimization through metalearning. In: 2017 IEEE International Conference on Data Mining (ICDM). IEEE, pp. 1051–1056.

Schelter, S., He, Y., Khilnani, J., Stoyanovich, J., 2020. FairPrep: promoting data to a first–class citizen in studies on fairness-enhancing interventions. In: International Conference on Extending Database Technology EDBT.

Scikit-learn, 2019. Scikit-learn: machine learning in Python. https://scikit-learn.org/stable/.

Sokol, K., Hepburn, A., Poyiadzi, R., Clifford, M., Santos-Rodriguez, R., Flach, P., 2020. FAT forensics: a Python toolbox for implementing and deploying fairness, accountability and transparency algorithms in predictive systems. J. Open Source Softw. 5 (49), 1904. doi:10.21105/joss.01904.

Soper, S., 2016. Amazon to Bring Same-Day Delivery to Bronx, Chicago After Outcry. Bloomberg. May 1, https://preview.tinyurl.com/hj7c3el

Soremekun, E., Udeshi, S., Chattopadhyay, S., 2022. ASTRAEA: grammar-based fairness testing. IEEE Trans. Softw. Engineering (TSE) 24. doi:10.1109/TSE.2022.3141758.

Statlog, 1994a. Census income data set. https://archive.ics.uci.edu/ml/datasets/census+income.

Statlog, 1994b. Statlog (German credit data) data set. https://tinyurl.com/4tp93njx.

Strickland, E., 2016. Doc bot preps for the O.R. IEEE Spectr. 53 (6), 32–60. doi:10.1109/MSPEC.2016.7473150.

Sun, B., Sun, J., Pham, L.H., Shi, J., 2022. Causality-based neural network repair. In: ACM/IEEE International Conference on Software Engineering (ICSE). Pittsburgh, PA, USA

Sweeney, L., 2013. Discrimination in online ad delivery. Commun. ACM (CACM) 56 (5), 44–54. doi:10.1145/2447976.2447990.

Tatman, R., 2017. Gender and dialect bias in YouTube's automatic captions. Workshop on Ethics in Natural Language Processing doi:10.18653/v1/W17-1606. Valencia, Spain

Thomas, P.S., da Silva, B.C., Barto, A.G., Giguere, S., Brun, Y., Brunskill, E., 2019. Preventing undesirable behavior of intelligent machines. Science 366 (6468), 999–1004. doi:10.1126/science.aag3311.

Tramer, F., Atlidakis, V., Geambasu, R., Hsu, D., Hubaux, J.-P., Humbert, M., Juels, A., Lin, H., 2017. FairTest: discovering unwarranted associations in data-driven applications. In: Security and Privacy (EuroS&P), 2017 IEEE European Symposium on. IEEE, pp. 401–416.

Udeshi, S., Arora, P., Chattopadhyay, S., 2018. Automated directed fairness testing, pp. 98–108. Montpellier, France

Varshney, K. R., 2018. Introducing AI fairness 360. https://www.ibm.com/blogs/research/2018/09/ai-fairness-360/.

Warshaw, J., Taft, N., Woodruff, A., 2016. Intuitions, analytics, and killing ants: inference literacy of high school-educated adults in the {US}. In: Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016), pp. 271–285.

Wexler, J., 2018. The what-if tool: code-free probing of machine learning models. https://ai.googleblog.com/2018/09/the-what-if-tool-code-free-probing-of.html.

Woodruff, A., Fox, S.E., Rousso-Schindler, S., Warshaw, J., 2018. A qualitative exploration of perceptions of algorithmic fairness. In: Conference on Human Factors in Computing Systems (CHI), p. 656.

Zafar, M.B., Valera, I., Rodriguez, M.G., Gummadi, K.P., 2015. Fairness constraints: mechanisms for fair classification. Fairness, Accountability, and Transparency in Machine Learning (FAT ML). Lille, France

Zafar, M.B., Valera, I., Rodriguez, M.G., Gummadi, K.P., 2017. Fairness beyond disparate treatment & disparate impact: learning classification without disparate mistreatment. Fairness, Accountability, and Transparency in Machine Learning (FAT ML). Perth, Australia

Zafar, M.B., Valera, I., Rogriguez, M.G., Gummadi, K.P., 2017. Fairness constraints: mechanisms for fair classification. In: International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 962–970.

Zhang, B.H., Lemoine, B., Mitchell, M., 2018. Mitigating unwanted biases with adversarial learning. In: AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society.

Zhang, H., Shahbazi, N., Chu, X., Asudeh, A., 2021. FairRover: explorative model building for fair and responsible machine learning. Workshop on Data Management for End-To-End Machine Learning doi:10.1145/3462462.3468882.

Zhang, L., Zhang, Y., Zhang, M., 2021. Efficient white-box fairness testing through gradient search. In: International Symposium on Software Testing and Analysis (ISSTA). Association for Computing Machinery, pp. 103–114. doi:10.1145/3460319.3464820. Denmark

Zhang, P., Wang, J., Sun, J., Dong, G., Wang, X., Wang, X., Dong, J.S., Dai, T., 2020. White-box fairness testing through adversarial sampling. In: ACM/IEEE International Conference on Software Engineering (ICSE), pp. 949–960. doi:10.1145/3377811.3380331. Seoul, South Korea