Journal of Machine Learning Research 24 (2023) 1-8 Submitted 3/23; Published 7/23

Fairlearn: Assessing and Improving Fairness of AI Systems

Hilde Weerts[1] H.J.P.WEERTS@TUE.NL
Miroslav Dudík[2] MDUDIK@MICROSOFT.COM
Richard Edgar[2] RIEDGAR@MICROSOFT.COM
Adrin Jalali ADRIN.JALALI@GMAIL.COM
Roman Lutz[2] ROMANLUTZ@MICROSOFT.COM
Michael Madaio[2]* MADAIOM@GOOGLE.COM
The authors are the current maintainers of Fairlearn, and additionally have the following affiliations: [1]*Eindhoven University of Technology,* [2]*Microsoft*

Editor: Alexandre Gramfort

Abstract

Fairlearn is an open source project to help practitioners assess and improve fairness of artificial intelligence (AI) systems. The associated Python library, also named *fairlearn*, supports evaluation of a model's output across affected populations and includes several algorithms for mitigating fairness issues. Grounded in the understanding that fairness is a sociotechnical challenge, the project integrates learning resources that aid practitioners in considering a system's broader societal context.

Keywords: algorithmic fairness, artificial intelligence, machine learning, Python

1. Introduction

As artificial intelligence (AI) impacts more of our everyday lives, there is a growing need to ensure that algorithmic systems do not disproportionately harm minorities, historically disadvantaged populations, and other groups considered sensitive from an ethical or legal perspective (Crawford, 2013; O'Neil, 2016; Broussard, 2018; Noble, 2018; Benjamin, 2019). Fairness of AI systems is a topic of multiple academic venues,[1] a priority for regulators (EC, 2021; OSTP, 2022), and a focus of several open source projects (Lee and Singh, 2021).

In this paper, we describe Fairlearn, an open source project that seeks to help data science practitioners with assessing and improving fairness of AI systems. The project consists of a Python library, called *fairlearn*, accompanied with various learning resources. Both the library and the learning resources are licensed under MIT license and available online.[2] The library aims to provide an easy-to-use API that blends well with popular libraries of the Python ecosystem, including *scikit-learn* (Pedregosa et al., 2011), *pandas* (McKinney, 2010), *matplotlib* (Hunter, 2007), *TensorFlow* (Abadi et al., 2015), and *PyTorch*

∗. Michael is currently employed by Google, but contributed to this work while at Microsoft.
1. `https://facctconference.org/` (FAccT)
`https://www.aies-conference.com/` (AIES)
2. `https://github.com/fairlearn/fairlearn`
`https://fairlearn.org`

(Paszke et al., 2019). Through our learning resources, we hope to provide practitioners with the knowledge and skills to effectively assess and mitigate unfairness.

Fairlearn is a community-driven project with independent governance,[3] following a code of conduct adapted from the Contributor Covenant.[4] The project is under active development and welcomes community contributions to the source code and the learning resources.

Fairlearn Perspective on AI Fairness. In Fairlearn, we consider AI fairness through the lens of fairness-related harms (Crawford, 2017), by which we mean negative impacts for groups of people, such as those defined in terms of race, gender, age or disability status.

Development of Fairlearn is firmly grounded in the understanding that fairness of AI systems is a sociotechnical challenge (cf. Green, 2021). Because there are many complex sources of unfairness—some societal and some technical—it is not possible to fully "de-bias" a system or to guarantee fairness (e.g., Blodgett et al., 2020). Instead, our goal is to help practitioners assess fairness-related harms, review the impacts of different mitigation strategies, and make trade-offs appropriate to their scenario. This may sometimes mean advocating for not deploying the system at all (Baumer and Silberman, 2011). AI fairness is related to, but distinct from anti-discrimination laws (Xiang and Raji, 2019), so our documentation avoids (mis)use of legal terminology (Watkins et al., 2022) and encourages users to understand what fairness means for their sociotechnical context before applying or adapting Fairlearn.

Fairlearn largely focuses on two types of fairness-related harms: *allocation harms* and *quality-of-service harms*. Allocation harms occur when AI systems are used to allocate opportunities or resources in ways that can have significant negative impacts on people's lives, for example, when an AI system for recommending patients into high-risk care management programs is less likely to select Black patients than white patients of similar health (Obermeyer et al., 2019). Quality-of-service harms occur when a system does not work as well for members of one group as it does for members of another group, for example, when a computer vision system has higher error rates for images of women with darker skin than for images of men with lighter skin (Buolamwini and Gebru, 2018).

2. Fairness Assessment

One of the key goals of the *fairlearn* library is to support fairness assessment. The goal of fairness assessment is to answer the question: *Which groups of people may be disproportionately negatively impacted by an AI system and in what ways?* In the context of allocation and quality-of-service harms, this means to evaluate how well the system performs for different population groups by calculating some performance metric, like an error rate, on different slices of data. This is called *disaggregated evaluation* (Barocas et al., 2021).

`MetricFrame` class. The primary tool for disaggregated evaluation in the *fairlearn* library is the `MetricFrame` class in the `fairlearn.metrics` module. Its API combines *scikit-learn* and *pandas* conventions. In its simplest form, `MetricFrame` is initialized by providing one or more metric functions together with input arrays `y_true`, `y_pred`, and `sensitive_`

3. The project started in May 2018 as a Microsoft open source project and its initial scope was outlined in a Microsoft technical report (Bird et al., 2020), but it transitioned to independent governance in 2021: `https://github.com/fairlearn/governance/blob/main/ORG-GOVERNANCE.md`
4. `https://www.contributor-covenant.org`

`features`. The first two arrays serve as inputs to metric functions, whereas the `sensitive_features` array is used to split the data into slices for disaggregated evaluation. Once a `MetricFrame` is constructed, the disaggregated metrics can be accessed as a *pandas* `Series` (for a single metric) or a *pandas* `DataFrame` (if multiple metrics are provided). `MetricFrame` also enables a comparison of metric values across groups, for example, in terms of differences or ratios. Plotting of results is supported via existing integration of *pandas* with *matplotlib*.

Fairness Metrics. The module `fairlearn.metrics` also provides metric functions that return scalars much like typical *scikit-learn* metrics. For example, functions `demographic_parity_difference` and `equalized_odds_difference` quantify how much the predictions of a given classifier depart from the fairness criteria known as *demographic parity* and *equalized odds* (see, e.g., Hardt et al., 2016). These two metrics are derived from a `MetricFrame` with a specific choice of input arguments. New fairness metrics can be obtained by using the `make_derived_metric` function, which wraps some of the `MetricFrame` functionality.

Comparison of Multiple Models. In addition to assessing fairness of a single model, `fairlearn.metrics` also enables a comparison of multiple models. For example, the function `plot_model_comparison` can be used to create a scatter plot, where each model is represented as a point with one coordinate equal to a metric quantifying overall performance and the other to a metric quantifying fairness, like the metrics from the `fairlearn.metrics` module.

3. Algorithmic Mitigation of Fairness-related Harms

The *fairlearn* library includes several methods for mitigating fairness-related harms. Many of the included methods are *meta-algorithms* in the sense that they act as wrappers around any standard (i.e., fairness-unaware) machine learning algorithms. This makes them quite versatile in practice. All of the implementations follow the API conventions of *scikit-learn*.

Following Barocas et al. (2019), *fairlearn* mitigation algorithms can be divided into three groups according to when they are applied relative to model training:

Pre-processing. Algorithms in this group mitigate unfairness by transforming input data before it is passed to a standard training algorithm. For example, `CorrelationRemover` in the module `fairlearn.preprocessing` applies a linear transformation to input features in order to remove any correlation with sensitive features. It follows the API of a *scikit-learn* transformer and therefore can be incorporated in a *scikit-learn* pipeline.

In-training.[5] Algorithms in this group directly train a model to satisfy fairness constraints. For example, the meta-algorithm `ExponentiatedGradient` in the module `fairlearn.reductions` implements the reduction approach of Agarwal et al. (2018, 2019). This meta-algorithm supports a wide range of fairness constraints and wraps any standard classification or regression algorithm, such as `LogisticRegression` from `sklearn.linear_model` or `XGBRegressor` from `xgboost`. An input to a reduction algorithm is an object that supports training on any provided (weighted) data set as well as a data set that includes sensitive features. The goal is to optimize a performance metric (such as classification accuracy) subject to fairness constraints (such as an upper bound on a difference between false negative rates).

5. Also called *in-processing* by some authors (Kamiran et al., 2013).

As another example, `AdversarialClassifier` and `AdversarialRegressor` in the module `fairlearn.adversarial` implement the adversarial mitigation approach of Zhang et al. (2018). These algorithms simultaneously train two neural network models, a predictor model and an adversarial model. The predictor model seeks to minimize the prediction loss function while also ensuring that the adversary model cannot infer sensitive features from the predictor outputs. The predictor and adversary neural nets can be defined either as *PyTorch* modules or *TensorFlow* models.

Post-processing. Algorithms in this group transform the output of a trained model. For example, `ThresholdOptimizer` in the module `fairlearn.postprocessing` implements the approach of Hardt et al. (2016), which takes in an existing (possibly pre-fit) machine learning model, uses its predictions as a scoring function, and identifies a separate threshold for each group defined by a sensitive feature in order to optimize some specified objective (such as balanced accuracy) subject to specified fairness constraints (such as false negative rate parity). The resulting classifier is a thresholded version of the provided machine learning model.

4. Learning Resources

Tackling fairness-related harms requires more than technical tools alone (Holstein et al., 2019). In a community-based effort, we have developed a comprehensive set of *learning objectives* that highlight what practitioners should know or be able to do when assessing and improving fairness of AI systems. These objectives are the basis for our learning resources.

To avoid divorcing technical and social aspects of AI fairness, our learning resources are integrated in the API reference and user guide of the *fairlearn* library. Besides coding examples and explanations, our user guide covers important concepts central to understanding machine learning models as part of a sociotechnical system, such as construct validity (Jacobs and Wallach, 2021) and the risks of abstracting away social context (Selbst et al., 2019).

Examples are crucial when learning to view fairness from a sociotechnical perspective. We provide tutorials (e.g., Gandhi et al., 2021) and example notebooks downloadable in the Jupyter format (Kluyver et al., 2016). We try to ensure that each notebook describes a real-world or realistic deployment context, focuses on real harms to real people, and avoids *abstraction traps* (Selbst et al., 2019).

The data sets provided in the module `fairlearn.datasets` also serve an educational role, as we use them to highlight sociotechnical aspects of fairness, with sections of the user guide highlighting fairness-related issues with several popular benchmark data sets.

5. Conclusions

Fairlearn is built and maintained by contributors with a variety of backgrounds and expertise. We believe that meaningful progress toward fairer AI systems requires input from a breadth of perspectives. We therefore encourage researchers, practitioners, and other stakeholders to contribute to Fairlearn as we experiment, learn, and evolve the project together.

Acknowledgments

We would like to thank Sarah Bird, Brandon Horn, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker for their critical contributions to the initial Fairlearn project (Bird et al., 2020). We would also like to thank all the members of the Fairlearn community who have contributed to the project in various ways, including documentation, code, bug reports, feature requests, and participation in our community calls. In particular, we would like to thank Michael Amoako, Alexandra Chouldechova, Parul Gupta, Laura Gutierrez Funderburk, Abdul Hannan Kanji, Kenneth Holstein, Lisa Ibañez, Sean McCarren, Manojit Nandi, Ayodele Odubela, Rens Oostenbach, Alex Quach, Kevin Robinson, Allie Saizan, Bram Schut, and Vincent Warmerdam for their valuable contributions.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from `https://www.tensorflow.org`.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A Reductions Approach to Fair Classification. In *International Conference on Machine Learning*, pages 60–69, 2018.

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair Regression: Quantitative Definitions and Reduction-Based Algorithms. In *International Conference on Machine Learning*, pages 120–129, 2019.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. 2019. `https://www.fairmlbook.org`.

Solon Barocas, Anhong Guo, Ece Kamar, Jacquelyn Krones, Meredith Ringel Morris, Jennifer Wortman Vaughan, W. Duncan Wadsworth, and Hanna Wallach. Designing Disaggregated Evaluations of AI Systems: Choices, Considerations, and Tradeoffs. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, pages 368–378, 2021.

Eric PS Baumer and M Six Silberman. When the implication is not to design (technology). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2271–2274, 2011.

Ruha Benjamin. *Race After Technology: Abolitionist Tools for the New Jim Code*. Polity, 2019.

Sarah Bird, Miroslav Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A Toolkit for Assessing and Improving Fairness in AI. Technical Report MSR-TR-2020-32, Microsoft, May 2020. `https://aka.ms/fairlearn-whitepaper`.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of" bias" in nlp. *arXiv preprint arXiv:2005.14050*, 2020.

Meredith Broussard. *Artificial Unintelligence: How Computers Misunderstand the World*. MIT Press, 2018.

Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.

Kate Crawford. The Hidden Biases in Big Data. *Harvard business review*, 1(4), 2013.

Kate Crawford. The Trouble with Bias. NeurIPS keynote, 2017. `https://www.youtube.com/watch?v=fMym_BKWQzk`.

European Commission (EC). Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts, 2021. COM (2021) 206 final.

Triveni Gandhi, Manojit Nandi, Miroslav Dudík, Hanna Wallach, Michael Madaio, Hilde Weerts, Adrin Jalali, and Lisa Ibañez. Fairness in AI Systems: From Social Context to Practice using Fairlearn. Tutorial presented at the 20th annual Scientific Computing with Python Conference (Scipy 2021), Virtual Event, 2021.

Ben Green. The contestation of tech ethics: A sociotechnical approach to technology ethics in practice. *Journal of Social Computing*, 2(3):209–225, 2021.

Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, 2016. `https://arxiv.org/abs/1610.02413`.

Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miroslav Dudík, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 1–16, 2019.

J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Abigail Z. Jacobs and Hanna Wallach. Measurement and Fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pages 375—385, 2021.

Faisal Kamiran, Toon Calders, and Mykola Pechenizkiy. Techniques for discrimination-free predictive models. In Bart Custers, Toon Calders, Bart W. Schermer, and Tal Z. Zarsky, editors, *Discrimination and Privacy in the Information Society - Data Mining and Profiling in Large Databases*, volume 3 of *Studies in Applied Philosophy, Epistemology and Rational Ethics*, pages 223–239. Springer, 2013. doi: 10.1007/978-3-642-30487-3_12. URL https://doi.org/10.1007/978-3-642-30487-3_12.

Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter Notebooks – A Publishing Format for Reproducible Computational Workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016.

Michelle Seng Ah Lee and Jat Singh. The Landscape and Gaps in Open Source Fairness Toolkits. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, 2021.

Wes McKinney. Data Structures for Statistical Computing in Python . In *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010.

Safiya Umoja Noble. *Algorithms of Oppression: How Search Engines Reinforce Racism*. NYU Press, 2018.

Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting Racial Bias in an Algorithm Used to Manage the Health of Populations. *Science*, 366 (6464):447–453, 2019.

Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown, 2016.

White House Office of Science and Technology Policy (OSTP). A Blueprint for an AI Bill of Rights, 2022. https://www.whitehouse.gov/ostp/ai-bill-of-rights/.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Andrew D Selbst, danah boyd, Sorelle A Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 59–68, 2019.

Elizabeth Anne Watkins, Michael McKenna, and Jiahao Chen. The four-fifths rule is not disparate impact: a woeful tale of epistemic trespassing in algorithmic fairness. *arXiv preprint arXiv:2202.09519*, 2022.

Alice Xiang and Inioluwa Deborah Raji. On the Legal Compatibility of Fairness Definitions. Workshop on Human-Centric Machine Learning at NeurIPS, 2019. `https://arxiv.org/abs/1912.00761`.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating Unwanted Biases with Adversarial Learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.