Fairlearn: A toolkit for assessing and improving fairness in AI

Version: September 22, 2020

Sarah Bird*, Miroslav Dudík*, Richard Edgar*, Brandon Horn*, Roman Lutz*, Vanessa Milan*, Mehrnoosh Sameki*, Hanna Wallach*, Kathleen Walker†

**Microsoft*, †*Allovus Design*

Summary

We introduce Fairlearn, an open source toolkit that empowers data scientists and developers to assess and improve the fairness of their AI systems. Fairlearn has two components: an interactive visualization dashboard and unfairness mitigation algorithms. These components are designed to help with navigating trade-offs between fairness and model performance. We emphasize that prioritizing fairness in AI systems is a sociotechnical challenge. Because there are many complex sources of unfairness—some societal and some technical—it is not possible to fully "debias" a system or to guarantee fairness; the goal is to mitigate fairness-related harms as much as possible. As Fairlearn grows to include additional fairness metrics, unfairness mitigation algorithms, and visualization capabilities, we hope that it will be shaped by a diverse community of stakeholders, ranging from data scientists, developers, and business decision makers to the people whose lives may be affected by the predictions of AI systems.

Fairness in AI

Artificial intelligence (AI) has transformed modern life via previously unthinkable feats, from self-driving cars and machines that can master the ancient boardgame Go to more "everyday" developments, such as customer support chatbots and personalized product recommendations. But, at the same time, these new opportunities have also raised new challenges; among these, most notably, are challenges that have highlighted the potential for AI systems to treat people unfairly. Indeed, the fairness of AI systems is one of the key concerns facing society as AI plays an increasingly important role in our daily lives.

Prioritizing fairness in AI systems is a sociotechnical challenge. AI systems can behave unfairly for a variety of reasons, some societal, some technical, and some a combination of both societal and technical. For example, some AI systems behave unfairly because of societal biases that are reflected in the datasets used to train them or because of societal biases in the assumptions and decisions made (either explicitly or implicitly) by teams throughout the AI development and deployment lifecycle. Other AI systems behave unfairly not because of societal biases, but because of dataset characteristics—such as too few data points about some group of people—or because of characteristics of other system components, such as user experiences that fail to account for the needs of some group of people. Given the many complex sources of unfairness, it is not possible to fully "debias" a system or to guarantee fairness; rather, the goal is to mitigate fairness-related harms as much as possible.

It can be difficult to distinguish between reasons why AI systems behave unfairly, especially because these reasons are not mutually exclusive and often exacerbate one another. Therefore, we define whether an AI system is behaving unfairly in terms of its impacts on people—i.e., in terms of fairness-related harms—and not in terms of specific causes, such as societal biases. AI systems can result in a variety of fairness-related harms, including harms involving people's individual experiences with AI systems or the ways that AI systems represent the groups to which they belong. For example:

- AI systems can unfairly allocate opportunities, resources, or information.

  Example: An automated resume-screening system trained using the resumes of people currently employed in the tech industry, where women are already underrepresented, may inadvertently withhold employment opportunities from women.

- AI systems can fail to provide the same quality of service to some people as they do to others.

  Example: A facial recognition system that does not consistently recognize the faces of people with particular skin tones will not provide the same quality of service to users who have those skin tones.

- AI systems can reinforce existing societal stereotypes.

  Example: A machine translation system that yields "She is a nurse" when translating "O bir hemşire" in Turkish, a gender-neutral language, into English will reinforce gender stereotypes.

- AI systems can denigrate people by being actively derogatory or offensive.

  Example: A chatbot will denigrate people if it learns to generate hate speech from intentionally malicious users.

- AI systems can over- or underrepresent groups of people, or treat them as if they don't exist.

  Example: An image search system that predominantly returns images of men in response to the query "chief executive officer" may underrepresent non-male executive officers.

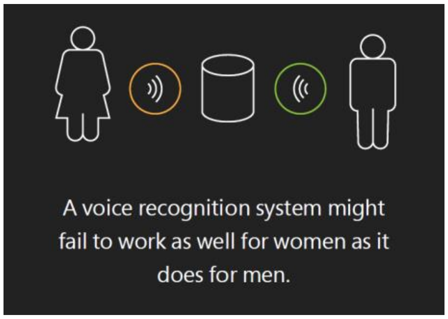
These types of harm are not mutually exclusive; a single AI system can exhibit more than one type. Fairness-related harms can have varying severities, but the cumulative impact of even relatively "non-severe" harms can be extremely burdensome or make people feel singled out or undervalued.

How Fairlearn can help

Because fairness in AI is a sociotechnical challenge, there is no software tool that will "solve" fairness in all AI systems. That is not to say that software tools cannot play a role in developing fairer AI systems—simply that they need to be precise and targeted, part of a holistic strategy that considers the sociocultural context of the systems being developed, and supplemented with additional resources.

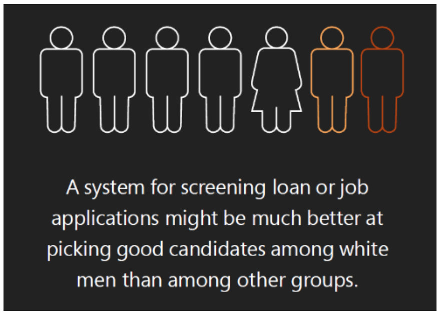
Fairlearn is one such tool. It is an open source toolkit that empowers data scientists and developers to assess and improve the fairness of their AI systems. Fairlearn focuses on negative impacts—specifically, allocation harms and quality-of-service harms—for groups of people, such as those defined in terms of race, sex, age, or disability status. Fairlearn’s fairness metrics and interactive dashboard can help with assessing which groups of people might be negatively impacted by a model, while Fairlearn’s unfairness mitigation algorithms can help with mitigating unfairness in classification and regression models.

Example of a quality-of-service harm



A voice recognition system might fail to work as well for women as it does for men.

Example of an allocation harm



A system for screening loan or job applications might be much better at picking good candidates among white men than among other groups.

There are many aspects of fairness that are beyond the scope of Fairlearn. For example, Fairlearn cannot mitigate stereotyping harms, denigration harms, or over- or underrepresentation harms (except indirectly, when these harms arise as a result of allocation harms or quality-of-service harms). Also, Fairlearn does not focus on broader societal aspects of fairness, such as justice or due process.

Prioritizing fairness in AI often means making trade-offs based on competing priorities; there are seldom clear-cut answers. It is therefore important to be explicit and transparent about those priorities and assumptions. Moreover, there is no single definition of fairness that will apply equally well to all AI systems in all settings. Fairlearn therefore enables data scientists and developers to select a fairness metric that is appropriate for their setting, to navigate trade-offs between fairness and model performance, and to select an unfairness mitigation algorithm that best fits their needs.

Assessing and mitigating unfairness with Fairlearn

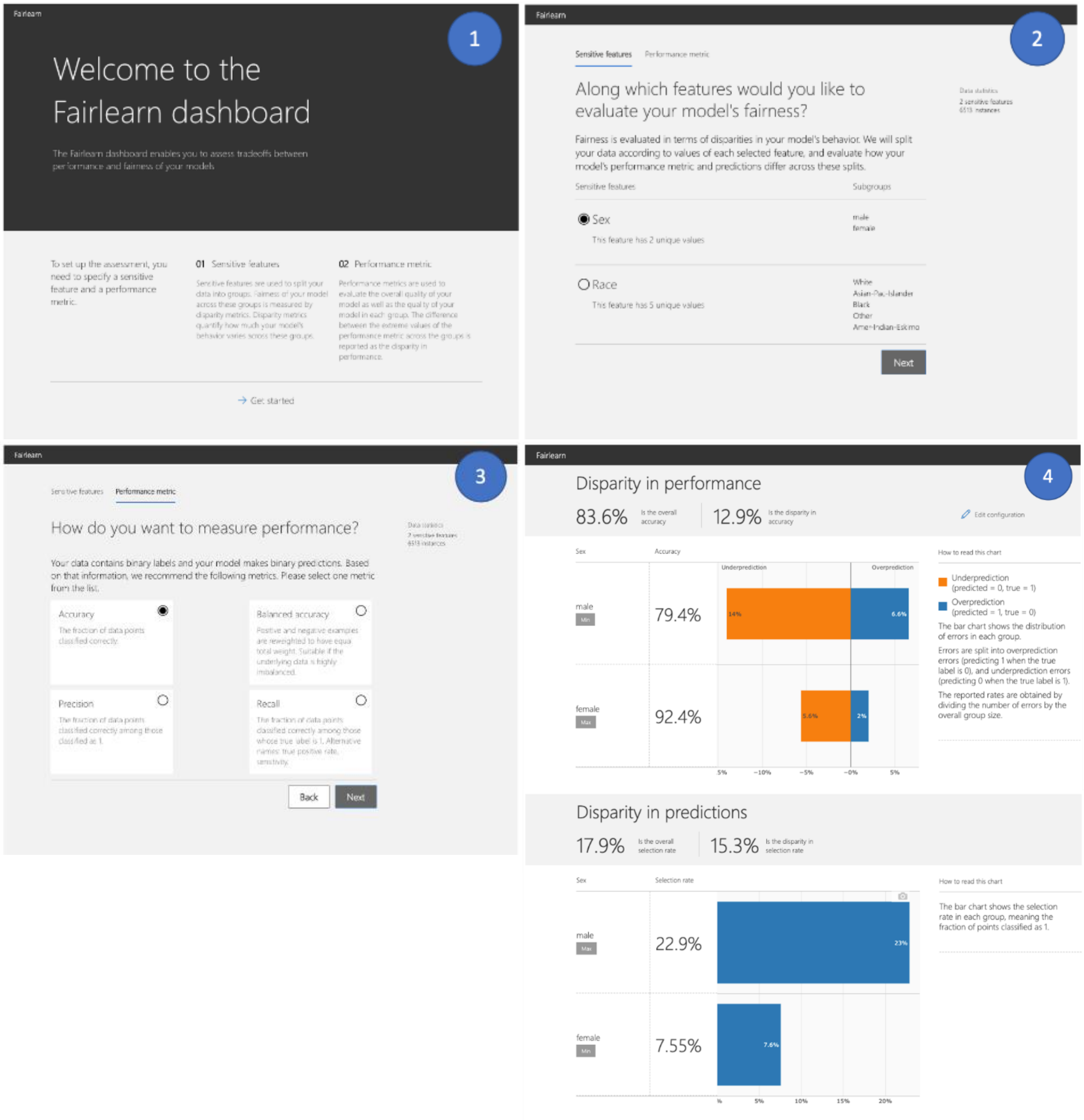
Fairlearn has two components: an interactive visualization dashboard and mitigation algorithms. These components are designed to help with navigating trade-offs between fairness and model performance.

Interactive visualization dashboard

Fairlearn’s interactive visualization dashboard can help users to (a) assess which groups of people might be negatively impacted by a model and (b) compare multiple models in terms of their fairness and performance. Fairlearn supports a wide range of fairness metrics for assessing a model’s impacts on different groups of people, covering both classification and regression tasks. The classification fairness metrics that are supported by Fairlearn include demographic parity, equalized odds, and worst-case accuracy rate; the regression fairness metrics include worst-case mean squared error and worst-case log loss. Each metric provides a different way of quantifying fairness with respect to groups of people, where the groups are defined in terms of a sensitive feature like “sex,” “age,” or “disability status.”

When setting up the dashboard for fairness assessment, the user selects (a) the sensitive feature (e.g, “sex” or “age”) that will be used to assess the fairness of one or multiple models and (b) the performance metric (e.g., accuracy rate) that will be used to assess model performance.

Example of the Fairlearn dashboard setup and an assessment of a single model.



Welcome to the Fairlearn dashboard

The Fairlearn dashboard enables you to assess tradeoffs between performance and fairness of your models

To set up the assessment, you need to specify a sensitive feature and a performance metric.

01 Sensitive features

Sensitive features are used to split your data into groups. Fairness of your model across these groups is measured by disparity metrics. Disparity metrics quantify how much your model's behavior varies across these groups.

02 Performance metric

Performance metrics are used to evaluate the overall quality of your model as well as the quality of your model in each group. The difference between the extreme values of the performance metric across the groups is reported as the disparity in performance.

→ Get started

Sensitive features | Performance metric

Along which features would you like to evaluate your model's fairness?

Fairness is evaluated in terms of disparities in your model's behavior. We will split your data according to values of each selected feature, and evaluate how your model's performance metric and predictions differ across these splits.

Data statistics
2 sensitive features
6513 instances

Sensitive features	Subgroups
● Sex This feature has 2 unique values	male female
○ Race This feature has 5 unique values	White Asian-Pac-Islander Black Other Amer-Indian-Eskimo

Next

Sensitive features | Performance metric

How do you want to measure performance?

Data statistics
2 sensitive features
6513 instances

Your data contains binary labels and your model makes binary predictions. Based on that information, we recommend the following metrics. Please select one metric from the list.

● Accuracy
The fraction of data points classified correctly.

○ Balanced accuracy
Positive and negative examples are reweighted to have equal total weight. Suitable if the underlying data is highly imbalanced.

○ Precision
The fraction of data points classified correctly among those classified as 1.

○ Recall
The fraction of data points classified correctly among those whose true label is 1. Alternative names: true positive rate, sensitivity.

Back | Next

Disparity in performance

83.6% Is the overall accuracy | 12.9% Is the disparity in accuracy

Edit configuration

Sex	Accuracy
male (Min)	79.4%
female (Max)	92.4%

How to read this chart

🟧 Underprediction (predicted = 0, true = 1)
🟦 Overprediction (predicted = 1, true = 0)

The bar chart shows the distribution of errors in each group.

Errors are split into overprediction errors (predicting 1 when the true label is 0), and underprediction errors (predicting 0 when the true label is 1).

The reported rates are obtained by dividing the number of errors by the overall group size.

Disparity in predictions

17.9% Is the overall selection rate | 15.3% Is the disparity in selection rate

Sex	Selection rate
male (Max)	22.9%
female (Min)	7.55%

How to read this chart

The bar chart shows the selection rate in each group, meaning the fraction of points classified as 1.

These selections are used to generate visualizations of a model’s impacts on groups defined in terms of the sensitive feature (e.g., accuracy rate for “female” and accuracy rate for “male,” as defined in terms of the “sex” feature). The dashboard also allows the user to compare the fairness and performance of multiple models, enabling them to navigate trade-offs and find a model that fits their needs.

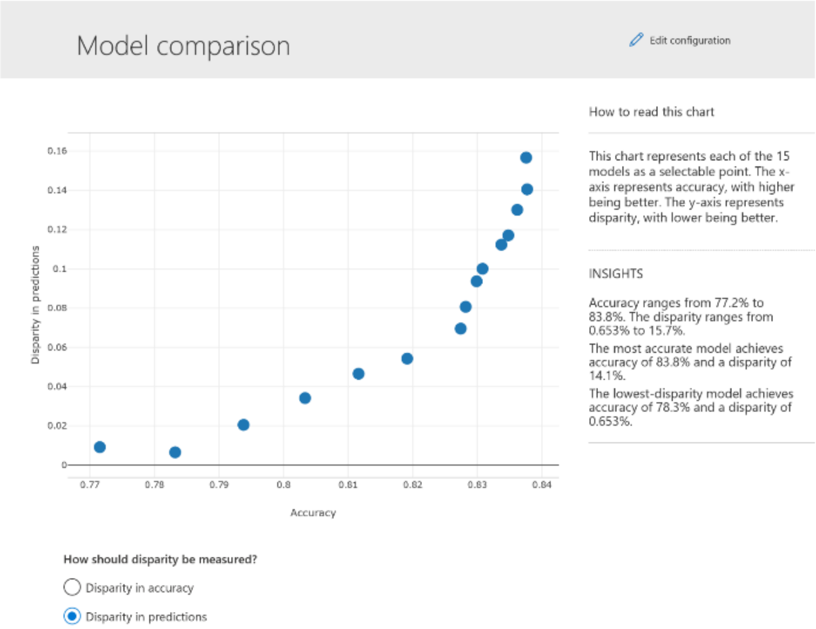
Unfairness mitigation algorithms: Postprocessing and reductions

Fairlearn’s unfairness mitigation algorithms can help users to improve the fairness of their AI systems. Fairlearn includes two types of mitigation algorithms: postprocessing algorithms and reduction algorithms. Both operate as “wrappers” around any standard classification or regression algorithm.

Fairlearn’s postprocessing algorithms take an already-trained model and transform its predictions so that they satisfy the constraints implied by the selected fairness metric (e.g., demographic parity) while maximizing model performance (e.g., accuracy rate); there is no need to retrain the model. For example, given a model that predicts the probability of defaulting on a loan, a postprocessing algorithm will try to find a threshold above which an applicant should get a loan. This threshold typically needs to be different for each group of people (defined in terms of the selected sensitive feature). We emphasize that this limits the scope of postprocessing algorithms, because sensitive features may not be available to use at deployment time, or may be inappropriate to use or (in some domains) prohibited by law.

Fairlearn’s reduction algorithms treat any standard classification or regression algorithm as a black box, and iteratively (a) re-weight the data points and (b) retrain the model after each re-weighting. After 10 to 20 iterations, this process results in a model that satisfies the constraints implied by the selected fairness metric while maximizing model performance. (The fact that it is possible to find such a model by merely re-weighting the data and retraining a standard algorithm is, at first glance, surprising, but this approach is backed by mathematical theory.) We note that reduction algorithms do not need access to sensitive features at deployment time, and work with many different fairness metrics. These algorithms also allow for training multiple models that make different trade-offs between fairness and model performance, which users can compare using Fairlearn’s interactive visualization dashboard.

Comparison of multiple models using the Fairlearn dashboard.



How to read this chart

This chart represents each of the 15 models as a selectable point. The x-axis represents accuracy, with higher being better. The y-axis represents disparity, with lower being better.

INSIGHTS

Accuracy ranges from 77.2% to 83.8%. The disparity ranges from 0.653% to 15.7%.

The most accurate model achieves accuracy of 83.8% and a disparity of 14.1%.

The lowest-disparity model achieves accuracy of 78.3% and a disparity of 0.653%.

How should disparity be measured?

○ Disparity in accuracy

● Disparity in predictions

Fairlearn as a community effort

Fairlearn is a community-driven open source project, to be shaped through stakeholder engagement. Its development and growth are guided by the belief that meaningful progress toward fairer AI systems requires input from a breadth of perspectives, ranging from data scientists, developers, and business decision makers to the people whose lives may be affected by the predictions of AI systems.

How we got here

Fairlearn started as a Python package to accompany the research paper, "A Reductions Approach to Fair Classification." The package provided a reduction algorithm for mitigating unfairness in binary classification models—a setting that was commonly studied in the machine learning community. The paper and the Python package were well received, so some of the co-authors wanted to translate the research into an industry context. However, they discovered that practitioners typically need to address more fundamental fairness issues before they can use specific algorithms, and that mitigating unfairness in binary classification models is a relatively rare use case. They also discovered that fairness assessment is a common need, along with access to domain-specific guides to fairness metrics and unfairness mitigation algorithms. Additionally, many use cases take the form of regression or ranking, rather than classification. As a result of these insights, fairness assessment and use-case notebooks became key components of Fairlearn. Fairlearn also focuses on machine learning tasks beyond binary classification.

Where we want to go next

As an open-source project, Fairlearn strives to incorporate the best of research and practice. AI is a rapidly evolving field, and fairness in AI is all the more so. We therefore encourage researchers, practitioners, and other stakeholders to contribute fairness metrics, unfairness mitigation algorithms, and visualization capabilities to Fairlearn as we experiment, learn, and evolve the project together.

There are many areas for future enhancement and growth. For example, Fairlearn currently supports only "group fairness"—i.e., fairness with respect to groups of people, such as those defined in terms of race, sex, age, or disability status—and not other conceptualizations of fairness, such as "individual fairness" or "counterfactual fairness." Fairlearn also currently includes only three unfairness mitigation algorithms (one postprocessing algorithm and two reduction algorithms), although we note that these algorithms are not restricted to classification tasks. Besides adding fairness metrics, conceptualizations of fairness, and unfairness mitigation algorithms, we also hope that Fairlearn will expand to cover more complex machine learning tasks in areas like counterfactual reasoning, computer vision, and natural language processing. We also anticipate integrating Fairlearn with interpretability tools, such as InterpretML. Ultimately, we hope that Fairlearn will become more than a software tool—a vibrant community and resource center that provides not only code, but also resources like domain-specific guides for when and when not to use different fairness metrics and unfairness mitigation algorithms.

Get started with Fairlearn.

Acknowledgements

We would like to thank Janhavi Mahajan, Sangeeta Mudnal, Miruna Oprescu, Chinmay Singh, Mark Soper, and Vincent Xu, as well as members of the Fairlearn community, for their contributions to Fairlearn. Some of the material presented here was based on content developed by Fairlearn contributors, Microsoft's Aether Fairness and Inclusiveness Working Group, and others.