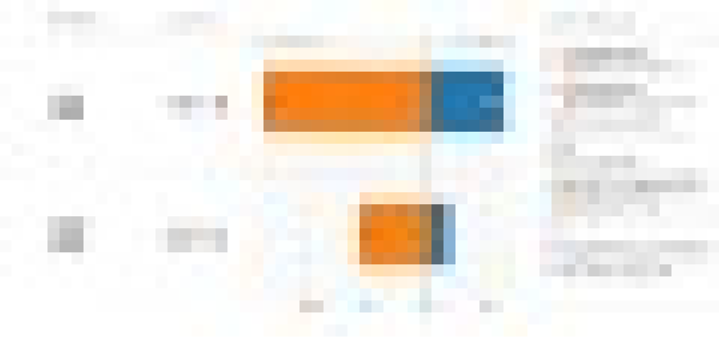
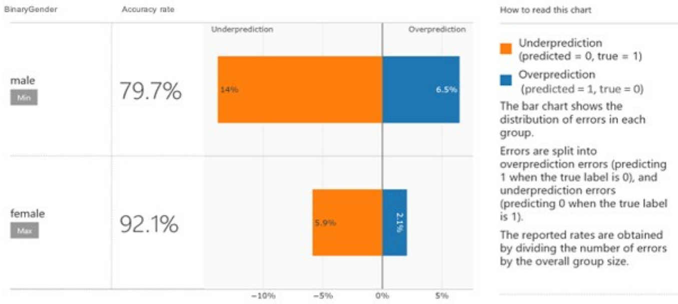
medium.com

How to Assess AI System’s Fairness and Mitigate Any Observed Unfairness Issues

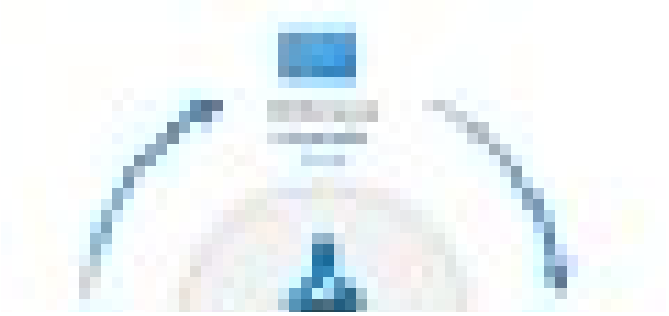
ODSC - Open Data Science

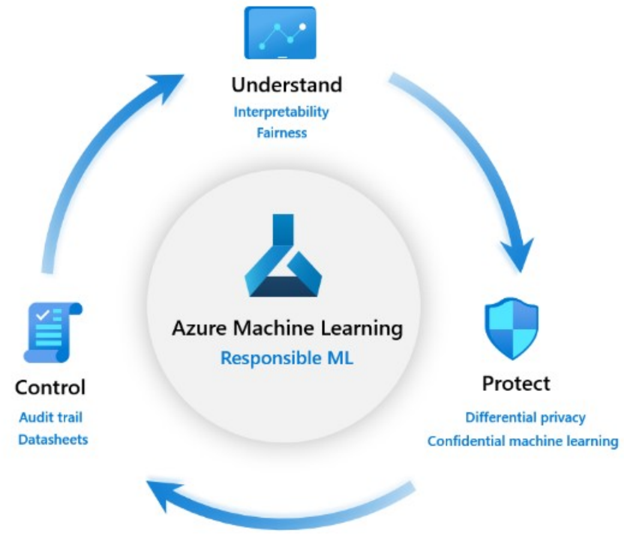
10-13 minuten





As we are leveraging data for making significant decisions that affect individual lives in domains such as health care, justice, finance, education, marketing, and employment, it is important to ensure the safe, ethical, and responsible use of AI. In collaboration with the Aether Committee and its working groups, Microsoft is bringing the latest research in responsible AI to Azure: these new responsible ML capabilities in Azure Machine Learning and our open source toolkits, empower data scientists and developers to ***understand*** machine learning models, ***protect*** people and their data, and ***control*** the end-to-end machine learning process.





In 2015, Claire Cain Miller wrote on The New York Times that there was a widespread belief that software and algorithms that rely on data were objective. Five years later, we know for sure that AI is not free of human influence. Data is created, stored, and processed by people, machine learning algorithms are written and maintained by people, and AI applications simply reflect people's attitudes and behavior.

Data scientists know that no longer accuracy is the only concern when developing machine learning models, fairness must be considered as well. In order to make sure that machine learning solutions are fair and the value of their predictions easy to understand and explain, it is essential to build tools that developers and data scientists can use to assess their AI system's fairness and mitigate any observed unfairness issues.

This article will focus on **AI fairness**, by explaining the following aspects and tools:

1. **Fairlearn**: a tool to assess AI system's fairness and mitigate any observed unfairness issues
2. How to use **Fairlearn** in **Azure Machine Learning**
3. What we mean by fairness
4. **Fairlearn algorithms**
5. **Fairlearn dashboard**
6. Comparing multiple models
7. Additional resources and **how to contribute**

Fairlearn is a Python package that empowers developers of artificial intelligence (AI) systems to assess their system's fairness and mitigate any observed unfairness issues. Fairlearn contains mitigation algorithms as well as a Jupyter widget for model assessment. The Fairlearn package has two components:

- A *dashboard* for assessing which groups are negatively impacted by a model, and for comparing multiple models in terms of various fairness and accuracy metrics.
- *Algorithms* for mitigating unfairness in a variety of AI tasks and along a variety of fairness definitions.

There is also a collection of Jupyter notebooks and an a detailed API guide, that you can check to learn how to leverage Fairlearn for your own data science scenario.

The Fairlearn package can be installed via:

```
pip install fairlearn
```

or optionally with a full feature set by adding extras, e.g. pip install fairlearn[customplots], or you can clone the repository locally via:

```
git clone git@github.com:fairlearn/fairlearn.git
```

In Azure Machine Learning, there are a few options to use Jupyter notebooks for your experiments:

If you'd like to bring your own notebook server for local development, follow these steps:

1. Use the instructions at **Azure Machine Learning SDK** to install the Azure Machine Learning SDK for Python
2. Create an **Azure Machine Learning workspace**.
3. Write a **configuration file**
4. Clone **the GitHub repository**.

```
git clone git@github.com:fairlearn/fairlearn.git
```

5. Start the notebook server from your cloned directory.

```
jupyter notebook
```

For more information, see Install the Azure Machine Learning SDK for Python.

b) Get Fairlearn samples on DSVM

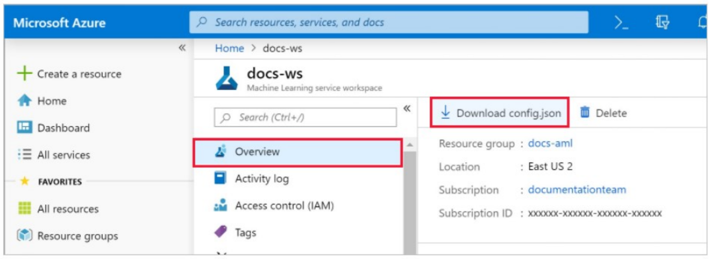
The Data Science Virtual Machine (DSVM) is a customized VM image built specifically for doing data science. If you create a DSVM, the SDK and notebook server are installed and configured for you. However, you'll still need to create a workspace and clone the sample repository.

1. **Create an Azure Machine Learning workspace**.
2. Clone **the GitHub repository**.

```
git clone git@github.com:fairlearn/fairlearn.git
```

3. Add a workspace configuration file to the cloned directory using either of these methods:

- In the Azure portal, select **Download config.json** from the **Overview** section of your workspace.



Resource group : docs-aml
Location : East US 2
Subscription : documentationteam
Subscription ID : xxxxxx-xxxxxx-xxxxxx-xxxxxx

- Create a new workspace using code in the configuration.ipynb notebook in your cloned directory

  4. Start the notebook server from your cloned directory:

```
jupyter notebook
```

Fighting against unfairness and discrimination has a long history in philosophy and psychology, and recently in machine learning. However, in order to be able to achieve fairness, we should first define the notion of it. An AI system can behave unfairly for a variety of reasons and many different fairness explanations have been used in literature, making this definition even more challenging. In general, fairness definitions fall under three different categories as follows:

- *Individual Fairness* — Give similar predictions to similar individuals.
- *Group Fairness* — Treat different groups equally.
- *Subgroup Fairness* — Subgroup fairness intends to obtain the best properties of the group and individual notions of fairness.

In Fairlearn, we define whether an AI system is behaving unfairly in terms of its impact on people — i.e., in terms of harms. We focus on two kinds of harms:

- *Allocation harms*. These harms can occur when AI systems extend or withhold opportunities, resources, or information. Some of the key applications are in hiring, school admissions, and lending.
- *Quality-of-service harms*. Quality of service refers to whether a system works as well for one person as it does for another, even if no opportunities, resources, or information are extended or withheld.

We follow the approach known as group fairness, which asks: Which groups of individuals are at risk of experiencing harm? The relevant groups need to be specified by the data scientist and are application-specific. Group fairness is formalized by a set of constraints, which require that some aspect (or aspects) of the AI system's behavior be comparable across the groups. The Fairlearn package enables the assessment and mitigation of unfairness under several common definitions.

Fairlearn contains the following algorithms for mitigating unfairness in binary classification and regression:

AlgorithmDescriptionClassification/RegressionSensitive featuresfairlearn.
reductions.
ExponentiatedGradientBlack-box approach to fair classification described in **A Reductions Approach to Fair Classification**binary classificationcategoricalfairlearn.
reductions.
GridSearchBlack-box approach described in Section 3.4 of **A Reductions Approach to Fair Classification**binary classificationbinaryfairlearn.
reductions.
GridSearchBlack-box approach that implements a grid-search variant of the algorithm described in Section 5 of **Fair Regression: Quantitative Definitions and Reduction-based Algorithms**regressionbinaryfairlearn.
postprocessing.
ThresholdOptimizerPostprocessing algorithm based on the paper **Equality of Opportunity in Supervised Learning**. This technique takes as input an existing classifier and the sensitive feature, and derives a monotone transformation of the classifier's prediction to enforce the specified parity constraints.binary classificationcategorical

Fairlearn dashboard is a Jupyter notebook widget for assessing how a model's predictions impact different groups (e.g., different ethnicities), and also for comparing multiple models along different fairness and accuracy metrics.

To assess a single model's fairness and accuracy, the dashboard widget can be launched within a Jupyter notebook as follows:

from fairlearn.widget import FairlearnDashboard

```
# A_test containts your sensitive features (e.g.,
age, binary gender)# sensitive_feature_names
containts your sensitive feature names# y_true
contains ground truth labels# y_pred contains
prediction
labelsFairlearnDashboard(sensitive_features=A_test,
sensitive_feature_names=['BinaryGender', 'Age'],
y_true=Y_test.tolist(),                    y_pred=
[y_pred.tolist()])
```

After the launch, the widget walks the user through the assessment set-up, where the user is asked to select:

1. the sensitive feature of interest (e.g., binary gender or age)

2. the accuracy metric (e.g., model precision) along which to evaluate the overall model performance as well as any disparities across groups.

   These selections are then used to obtain the visualization of the model's impact on the subgroups (e.g., model precision for females and model precision for males). The following figures illustrate the set-up steps, where binary gender is selected as a sensitive feature and the accuracy rate is selected as the accuracy metric:

Image for post

After the set-up, the dashboard presents the model assessment in two panels, as summarized in the table, and visualized in the screenshot below:

PanelDescriptionDisparity in accuracy This panel shows:

1. the accuracy of your model with respect to your selected accuracy metric (e.g., accuracy rate) overall as well as on different subgroups based on your selected sensitive feature (e.g., accuracy rate for females, accuracy rate for males);
2. the disparity (difference) in the values of the selected accuracy metric across different subgroups;
3. the distribution of errors in each subgroup (e.g., female, male). For binary classification, the errors are further split into overprediction (predicting 1 when the true label is 0), and underprediction (predicting 0 when the true label is 1).

Disparity in predictions This panel shows a bar chart that contains the selection rate in each group, meaning the fraction of data classified as 1 (in binary classification) or distribution of prediction values (in regression).

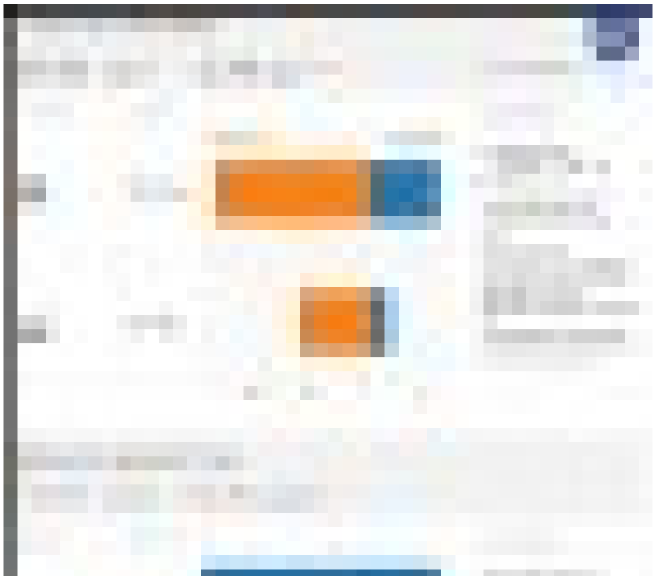


Image for post

An additional feature that this dashboard offers is the comparison of multiple models, such as the models produced by different learning algorithms and different mitigation approaches, including:

- fairlearn.reductions.GridSearch
- fairlearn.reductions.ExponentiatedGradient
- fairlearn.postprocessing.ThresholdOptimizer

As before, the user is first asked to select the sensitive feature and the accuracy metric. The model comparison view then depicts the accuracy and disparity of all the provided models in a scatter plot. This allows the user to examine trade-offs between algorithm accuracy and fairness. Moreover, each of the dots can be clicked to open the assessment of the corresponding model.

The figure below shows the model comparison view with binary gender selected as a sensitive feature and accuracy rate selected as the accuracy metric:

Image for post

For references and additional resources, please refer to:

- Fairlearn GitHub repo: www.aka.ms/FairlearnAI
- Azure Machine Learning: www.aka.ms/AzureMLDoc
- Responsible ML at Microsoft Build: www.aka.ms/Build2020ResponsibleML
- Responsible ML on Azure: www.aka.ms/AzureResponsibleML
- Responsible ML documentation:

www.aka.ms/ResponsibleMLDoc

- Discrimination-aware Data Mining: **http://pages.di.unipi.it/ruggieri/Papers/kdd2008.pdf**
- A Survey on Bias and Fairness in Machine Learning: **https://arxiv.org/pdf/1908.09635.pdf**
- Can an Algorithm Hire Better Than a Human? **https://www.nytimes.com/2015/06/26/upshot/can-an-algorithm-hire-better-than-a-human.html**

To contribute please check this contributing guide.