Demonstration Paper

FairnessEval: a Framework for Evaluating Fairness of Machine Learning Models

Andrea Baraldi
University of Modena and Reggio Emilia
Modena, Italy
andrea.baraldi96@unimore.it

Matteo Brucato
Microsoft
Redmond, USA
mbrucato@microsoft.com

Miroslav Dudík
Microsoft
New York, USA
mdudik@microsoft.com

Francesco Guerra
University of Modena and Reggio Emilia
Modena, Italy
francesco.guerra@unimore.it

Matteo Interlandi
Microsoft
Redmond, USA
mainterl@microsoft.com

ABSTRACT

Automated decision-making systems can potentially introduce biases, raising ethical concerns. This has led to the development of numerous bias mitigation techniques. Choosing a fairness-aware model often requires trial and error, as it is difficult to predict whether a mitigation measure will meet user requirements or how it will affect metrics like accuracy and runtime.

Existing fairness toolkits lack a comprehensive benchmarking framework. To bridge this gap, we present FairnessEval, a framework specifically designed to evaluate fairness in Machine Learning models. FairnessEval streamlines dataset preparation, fairness evaluation, and result presentation, while also offering customization options. In this demonstration, we highlight the functionality of FairnessEval in the selection and validation of fairness-aware models. We compare various approaches and simulate deployment scenarios to showcase FairnessEval effectiveness.

1 INTRODUCTION

The ethical considerations surrounding automated decision-making systems have prompted thoughtful discussions, with the goal of ensuring that the deployment of AI predictive models in real-world scenarios does not disproportionately or unjustly impact historically marginalized populations and groups [2]. The result is a growing collection of techniques and tools devoted to addressing bias and discrimination issues (see, e.g., the surveys [4, 9, 10]). These approaches are typically categorized into three groups depending on when they are implemented during the model training process. Pre-processing techniques aim to enhance fairness by altering the training data before it is fed into the Machine Learning (ML) algorithm [3, 5, 8]. In-processing techniques involve modifying the ML algorithm to address fairness during training [11–13]. Finally, post-processing techniques directly analyze and adjust the outputs of an already-trained model [6].

Implementing fairness-aware approaches in real-world situations presents significant challenges. One of the main issues is that while fair approaches enhance fairness on the metric they target, their performance on other metrics, such as accuracy and time efficiency, may be unpredictable [7]. Therefore, before being used in production, the ML models require multiple evaluations taking into account various assumptions related to the sensitive attributes and the metrics used to measure fairness. Moreover, as to standard model selection, an algorithm might perform well in certain data subsamples, but not necessarily in others, requiring extensive trial and error tuning. This persists throughout the application life cycle and can worsen due to external factors like concept drift.

Users aiming to incorporate fairness in their ML models often do not start from scratch. There are many open-source toolkits providing access to key mitigation algorithms and benchmark datasets. For example, AIF360 [1], Aequitas [2], and Fairlearn [3] provide datasets and implementations of many fairness-aware ML approaches. The Fairness dashboard [4] included in the Microsoft Responsible-AI-Toolbox, and the What-if Tool [5] are two tools that offer user interfaces to explore and assess bias in datasets and model predictions. The Responsible AI Tracker [6] is a JupyterLab Extension that allows users to evaluate models developed in Jupyter Notebook environments and compare them. The Amazon Sage Maker [7], a proprietary, non-open source framework requiring a paid subscription, allows users to build, train, and deploy machine ML models but it does not natively support fairness models, datasets, and metrics. Nevertheless, all these approaches do not provide any support for managing and running evaluations involving different datasets and mitigation strategies. To bridge this gap, we developed FairnessEval: a Python framework providing tools for data preparation, dataset generation, model evaluation, and result presentation of fairness-aware models. FairnessEval assists users in automatically organizing and executing experiments, collecting and presenting results. It also includes a dataset generation tool that allows users to create synthetic datasets with user-selected biases on specific protected attributes. FairnessEval is fully customizable and extensible. It provides a web interface and can be integrated directly into Python scripts or Jupyter Notebook environments.

The demonstration will show how the framework supports the tasks of selection and validation of fairness-aware models.

[1] https://github.com/Trusted-AI/AIF360
[2] https://github.com/dssg/aequitas
[3] https://github.com/fairlearn/fairlearn
[4] https://github.com/microsoft/responsible-ai-toolbox/blob/main/docs/fairness-dashboard-README.md
[5] https://pair-code.github.io/what-if-tool/
[6] https://github.com/microsoft/responsible-ai-toolbox-tracker
[7] https://aws.amazon.com/sagemaker/

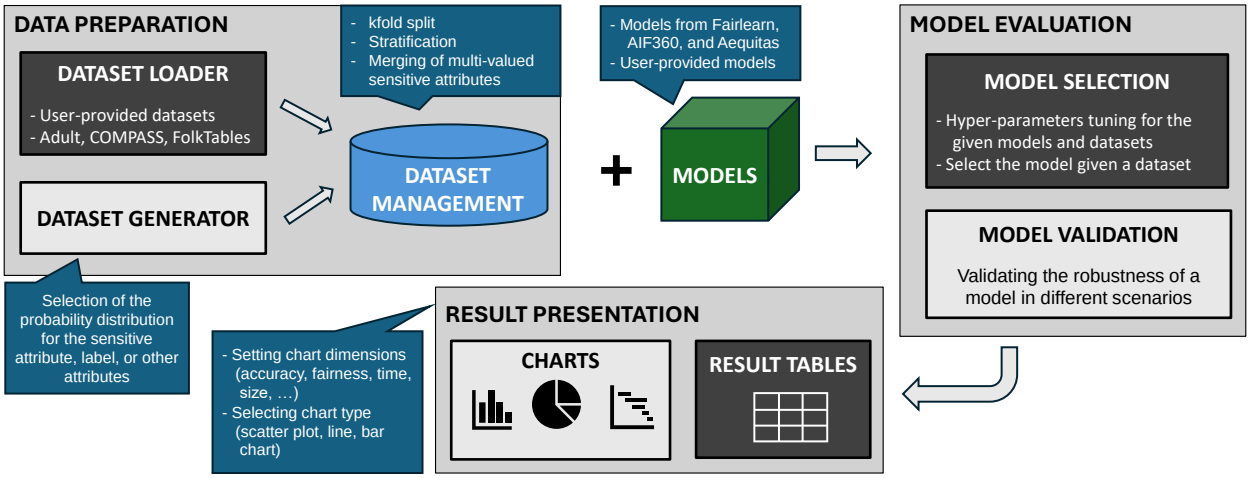


Figure 1: *FairnessEval Workflow.*

In particular, we will present scenarios that simulate the process of implementing a fair approach in production. While we have prepared predefined use cases covering pre-processing, in-processing, and post-processing mitigation strategies, attendees are encouraged to propose their own datasets (including synthetic ones generated using the data preparation component) and mitigation algorithms for evaluation.

2 FairnessEval ARCHITECTURE

FairnessEval consists of three main components, each representing a step in building a fair ML model: Data Preparation (Section 2.1), fairness-aware Model Evaluation (Section 2.2), and Result Presentation (Section 2.3). Figure 1 shows the workflow and the main responsibilities for each component. Users interact with the framework through either a user interface or Python APIs. FairnessEval is an open-source GitHub project (https://github.com/softlab-unimore/fairnesseval)[8].

2.1 Data Preparation

In FairnessEval, users have two options for selecting datasets to analyze: (1) choosing existing datasets (either uploaded by the user or pre-loaded in the framework) or (2) generating synthetic datasets with user-defined levels of bias in the sensitive attributes. Furthermore, the Data Preparation component offers functionalities for the analysis and management of sensitive attributes, such as transforming multi-valued sensitive attributes to binary, and merging multiple sensitive attributes into a single attribute. It also includes functions for splitting datasets using various strategies (like stratification with respect to the label, to the sensitive attributes, or both), thereby enabling different types of experiments.

Pre-loaded Datasets. The interface preloads datasets commonly used as benchmarks for experimental evaluations of fairness-aware approaches, including COMPAS [9], Adult [10], and Folktables [11]. These pre-loaded datasets include large datasets with multi-valued sensitive attributes (e.g., Folktables contains millions of records and multi-valued sensitive attributes). Alternatively, users can upload their own datasets via a specific interface or directly into a dedicated directory. The current version only supports CSV files.

Dataset Generator. The synthetic data generator creates datasets with various sensitive attributes, such as gender or race, and determines outcomes based on predefined probability distributions. The dataset is built by probabilistically determining each individual's outcome, denoted as Y_i, based on the category of the sensitive attribute, i.e. the probability of a positive outcome $Y_i = 1$ varies based on the value of the sensitive attribute A_i. For instance, we might consider the gender attribute with three categories: men (denoted as "m"), women ("w"), and non-binary individuals ("nb"). We may set parameters such that if A_i corresponds to "m", Y_i is set to 1 with a probability of 0.7, while for "nb" it is 0.6, and for "w" 0.65. To introduce additional fairness issues, we incorporate a modification step for each category, and inverse probability adjustment to control disparities. If A_i is "m", for example, we adjust Y_i with certain probabilities. If Y_i is initially 0, we flip it to 1 with a probability of 0.2; if Y_i is 1, we flip it to 0 with a probability of 0.1. Otherwise, Y_i remains unchanged. Similar adjustments are applied with varying probabilities to control the disparities across different groups. We complete the dataset with additional features ($j = 1, \ldots, d$). Each feature, denoted as X_{ij}, is derived from Y_i with a probability of $\frac{1}{2} + \text{eps}_j$, and its complement $(1 - Y_i)$ with the remaining probability. This randomization process is performed independently for each feature to provide diversity in the dataset. The parameter eps_j is the switching probability for feature j, allowing different levels of randomness across features. By default, we consider 10 features (d) and 1 million samples (n), with (eps) controlling the difficulty of fitting a classifier. A smaller value of epsilon suggests a more challenging task for the classifier, thereby encouraging a thorough examination of fairness across different subsets of the data.

2.2 Model Evaluation

In this component, FairnessEval allows users to run models against the prepared datasets. Users can specify a list of fairness-aware models (among the ones provided by the toolkits AIF360

[8] A video of the demonstration is available at https://www.youtube.com/watch?v=jWcZ0LGB3Zg
[9] https://github.com/propublica/compas-analysis/
[10] https://archive.ics.uci.edu/dataset/2/adult
[11] https://github.com/socialfoundations/folktables

or Fairlearn, or directly provided by the users [12]). The models are evaluated in batch, and the outputs passed to the Result Presentation component.

This component serves two primary purposes for the user: *model selection* and *model validation*. For model selection, the user can undertake two distinct tasks: (1) determining the hyperparameters that maximize a specific objective metric for a given dataset, or (2) selecting the most suitable model from a list of pre-configured models for the dataset in question. Similarly, the validation task requires users to specify a fairness-aware model, a list of datasets, and a strategy for split and stratification.

2.3 Result Presentation

FairnessEval provides a range of pre-configured diagrams to illustrate the efficiency and effectiveness of fairness-aware models. Users need to select the analysis dimensions to be depicted in the diagrams, as well as the metrics used for each dimension. Many analysis dimensions can impact the problem, including accuracy, fairness, time, dataset size, and model hyperparameters. For each dimension, users can select from a variety of possible metrics. Standard metrics for evaluating the accuracy of the ML model are implemented, such as accuracy rate, precision, recall, F1-score for classification problems, and various error measures for regression problems. Fairness can be evaluated using several metrics that emphasize demographic parity (e.g., disparate impact) and equalized odds (e.g., true positive rate balance).

Typical diagrams include: *accuracy vs. fairness plots* (Figures 3a and 3b), which highlight tradeoffs between fairness and accuracy (which sometimes decreases when the fairness improves); *fairness vs. time plots* (Figure 3c), which show how the search for fairness normally increases training times; *fairness (or accuracy) vs dataset fraction plots* (Figure 4), which show how the learning curve is influenced by the size of the training set.

3 FairnessEval DEMONSTRATION

The demonstration will highlight the capabilities of FairnessEval in the two operational scenarios it offers: the *selection* and *validation* of a fairness-aware model.

Model Selection. The scenario simulates the business task of selecting a suitable fairness-aware model for a given dataset. This process is inherently experimental, and FairnessEval supports the user through the required iterations to obtain a model with satisfying performance characteristics. As previously mentioned, applying a fairness-aware approach may reduce accuracy and increase training time compared with the unmitigated model in a way that can only be determined through experimental evaluations.

The goal of the scenario is to evaluate the performance of various ML models in predicting income levels using demographic data from the "Adult" dataset, also known as the "Census Income" dataset. This dataset contains demographic information such as age, education, and occupation, and is commonly used for classification tasks to predict whether an individual's income exceeds $50K per year. The "Adult" dataset contains sensitive attributes, such as race and sex. We evaluate the performance of five types of models available in our framework: an unmitigated

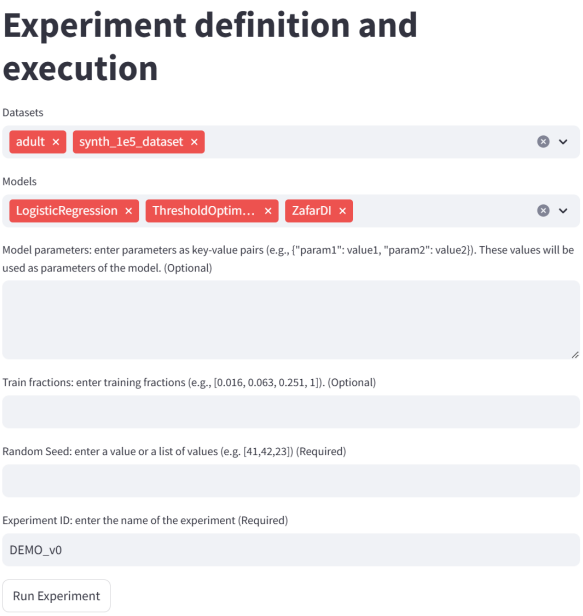


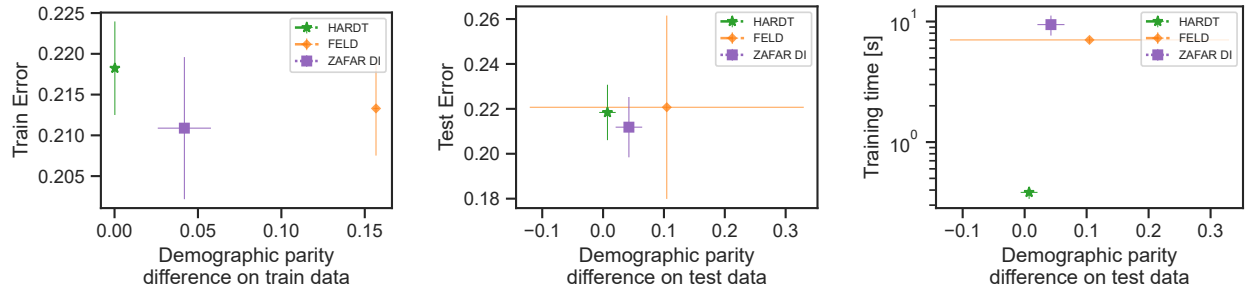
Figure 2: FairnessEval User interface for the definition of the experiment settings.

(LogisticRegression), a reduction approach (Expgrad) [13], a preprocessing (Feld), an in-processing (Zafar) and a post-processing approach (Hardt). As shown in Figure 2, the FairnessEval graphical interface supports the user in selecting the dataset (Adult is already preloaded by FairnessEval), the models, their hyperparameters and the task performed (model selection). The default setting involves evaluating models on a 3-fold cross-validated dataset that is stratified by both the target variable and the sensitive attribute. In this scenario, we generate three charts for presentation. Two charts evaluate accuracy vs. fairness in the training set and test set. The third chart evaluates fairness vs. time performance. We then decide to measure fairness via the demographic parity difference, and accuracy by plotting the error. FairnessEval automatically manages the experiments, by planning and performing (if required by the user) repeated runs with different random seeds and k-fold stratification to ensure the robustness of the results. Figures 3a and 3b show that the models Hardt and ExpGrad obtain the best performance. Feld performs as the unmitigated approach, and Zafar slightly better. Regarding the runtime performance, as depicted in Figure 3c, we notice that Hardt is the most efficient fairness-aware approach. However, Hardt is required to access the sensitive attribute since it implements a post-processing strategy. If this is not possible or allowed by the application, the second choice is ExpGrad, which is one order of magnitude slower.

Model Validation. This scenario simulates the task of evaluating and assessing the performance of a fairness-aware model under varying conditions. The goal is to measure how a preselected model performs in terms of both accuracy and fairness across different datasets and scenarios, simulating real-world conditions where datasets may vary or change over time. Users can test models using both real and synthetic datasets, allowing for a controlled evaluation of performance under different

[12] User-defined fairness-aware models have to simply implement `fit` and `predict` methods over Pandas dataframe data to be included in FairnessEval. Further instructions are provided in the repo.

[13] Among the in-training techniques to fairness, the reduction approaches wrap a generic ML approach and optimizes it over a user-selected level of fairness. ExpGrad [1] is one of such reduction approaches.



(a) Fairness vs accuracy in the training set. **(b) Fairness vs accuracy in the test set.** **(c) Fairness vs training time.**

Figure 3: Model Selection in FairnessEval.

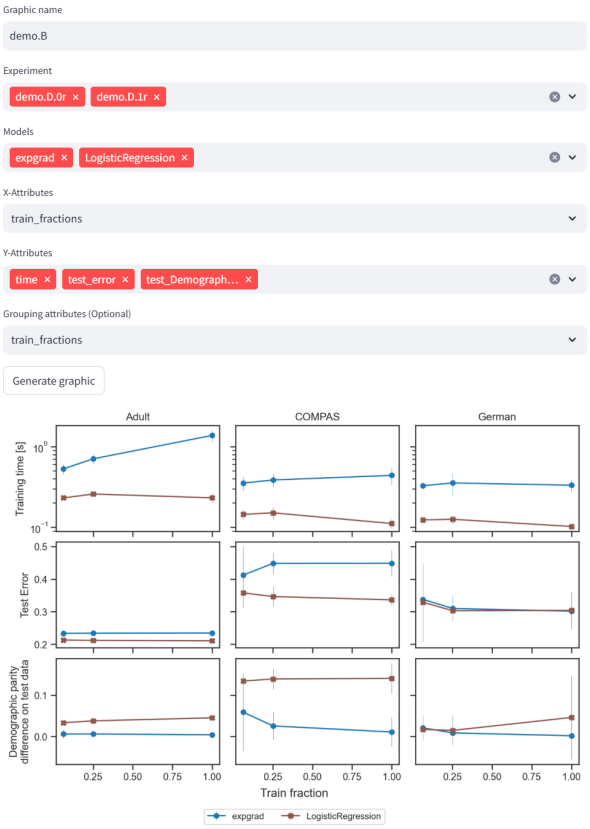


Figure 4: Model Validation: time, accuracy, fairness vs. dataset fractions.

operational contexts. Synthetic datasets are especially useful for exploring edge cases or specific conditions that may not be present in available data, providing a clearer picture of how models respond to certain biases or fairness constraints. The process involves experimenting with various model parameters, dataset splits, and sampling strategies to explore the model's behavior and identify the configuration that best meets the business needs. For example, the goal could be to evaluate the robustness of a fairness-aware approach like ExpGrad by testing its performance on different dataset fractions. This enables the identification of how the model's fairness and performance are affected when trained on smaller portions of the data. Figure 4 shows the plots generated by FairnessEval, where accuracy, fairness, and runtime performance are measured on three fractions of the dataset. We can notice that for all datasets the approach converges even with small fractions of the data.

4 CONCLUSION

The paper introduces FairnessEval: a tool for evaluating fairness in Machine Learning models. FairnessEval provides users with a set of tools for data preparation, model evaluation, result presentation, and specifically tailored for fairness use cases. During the demo, attendees can interact with the framework using the pre-loaded datasets and models, and by uploading their own.

REFERENCES

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

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. *Fairness and Machine Learning: Limitations and Opportunities.* http://www.fairmlbook.org.

[3] Flávio P. Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. 2017. Optimized Pre-Processing for Discrimination Prevention. In *NIPS*. 3992–4001.

[4] Simon Caton and Christian Haas. 2023. Fairness in Machine Learning: A Survey. *ACM Comput. Surv.* (2023).

[5] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *KDD*. ACM, 259–268.

[6] Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*.

[7] Maliha Tashfia Islam, Anna Fariha, Alexandra Meliou, and Babak Salimi. 2022. Through the Data Management Lens: Experimental Analysis and Evaluation of Fair Classification. In *SIGMOD Conference*. ACM, 232–246.

[8] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[9] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* 54, 6 (jul 2021). https://doi.org/10.1145/3457607

[10] Dana Pessach and Erez Shmueli. 2023. A Review on Fairness in Machine Learning. *ACM Comput. Surv.* 55, 3 (2023), 51:1–51:44.

[11] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *WWW*. ACM, 1171–1180.

[12] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *AISTATS (Proceedings of Machine Learning Research)*, Vol. 54. PMLR, 962–970.

[13] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 335–340.