FairPilot: An Explorative System for Hyperparameter Tuning through the Lens of Fairness[*]

Francesco Di Carlo[†] Nazanin Nezami[‡] Hadis Anahideh[§] Abolfazl Asudeh[¶]

Abstract

Despite the potential benefits of machine learning (ML) in high-risk decision-making domains, the deployment of ML is not accessible to practitioners, and there is a risk of discrimination. To establish trust and acceptance of ML in such domains, democratizing ML tools and fairness consideration are crucial. In this paper, we introduce FairPilot, an interactive system designed to promote the responsible development of ML models by exploring a combination of various models, different hyperparameters, and a wide range of fairness definitions. We emphasize the challenge of selecting the "best" ML model and demonstrate how FairPilot allows users to select a set of evaluation criteria, and then displays the Pareto frontier of models and hyperparameters as an interactive map. FairPilot is the first system to combine these features, offering a unique opportunity for users to responsibly choose their model.

1 Introduction

Predictive analytics and machine learning (ML) have become increasingly prevalent in sensitive decision-making domains, such as healthcare, finance, criminal justice, and education. These domains involve high-stakes decisions with potentially significant impacts on people's lives [14, 20, 24]. For instance, in the education sector, predictive analytics and ML can be used to predict student outcomes, identify at-risk students, and provide personalized learning plans. However, the use of these technologies in sensitive decision-making domains is a complex and multifaceted issue.

While ML holds a significant promise for high-risk domains, its acceptability among practitioners can be influenced by several factors. Two key factors are the lack of democratization in deployment and the potential to exacerbate inequalities. The former refers to the effort to make ML tools and technology widely accessible to a diverse group of users with different expertise and background. The latter refers to the fact that ML tools can perpetuate biases and discrimination if not designed and implemented carefully. For example, a model that assigns lower probabilities of success to students from certain demographic groups (such as students of color or students from low-income backgrounds) could exacerbate existing inequalities and result in unfair treatment for these students. This can create a disconnection between the use of ML and the values of equity and inclusion that are central to the mission of such sensitive domains (e.g., education).

Consider a domain practitioner who aims to develop a machine-learning model for a prediction task. Recent advances in AI have introduced a wide range of choices for each ML model that can be employed to make predictions. Having multiple options is beneficial as it gives the flexibility of choice to the data scientist. On the other hand, it is not clear which model is "better" for the given task. Therefore, exploring different options before making the final selection is cumbersome:

1. *Hyperparameters*: ML models are often associated with various hyperparameters that require predefined values before training the model. The selection of hyperparameters may highly affect the developed model and its performance. The combination of different values for hyperparameters makes the exploration space for each model *exponentially large* to the number of its hyperparameters.

2. *Time complexity*: Training ML models is resource hungry task that is coupled with a large number of ML models to be trained and the exponential space of hyperparameters choices, makes the responsible model selection process overwhelming, particularly for practitioners.

3. *Multiple evaluation criteria*:Model selection solely based on maximizing the accuracy is not enough, at least for sensitive decision-making environments since there often exists a trade-off between the fairness and (accuracy) performance of a model [7].

[*] This project was supported in part by the Institute of Education Sciences (R305D220055) and the National Science Foundation (NSF 2107290).

[†] University of Illinois Chicago. (fdicar2@uic.edu)

[‡] University of Illinois Chicago.(nnezam2@uic.edu)

[§] University of Illinois Chicago. (hadis@uic.edu)

[¶] University of Illinois Chicago. (asudeh@uic.edu)

On top of that, fairness is not a unique notion; there are many (more than 21) fairness definitions [5, 29], and those are often in trade-off with each other [17]. Even more unwieldy, the relationships between the fairness notions (and model performance) are context-specific and are not clear apriori [1]. Therefore, a data scientist is unlikely to be able to come up with a formula to combine all fairness and model performance criteria into a single objective.

Responsible model development without the aid of **assistive exploratory tools** is formidable, if not infeasible, task for data scientists. Conversely, despite extensive advances by the Fair ML community, there is a research gap and a desperate need for *interactive* explorative systems to help data-scientist practitioners develop ML models responsibly [3, 4, 18].

This paper is an attempt towards filling this gap. To the best of our knowledge, **FairPilot** is the first interactive system that enables users to explore a combination of (a) various ML models and (b) different hyperparameter combinations while considering (c) a wide range of fairness definitions. FairPilot allows the user to select a set of evaluation criteria and shows the Pareto frontier of models and hyperparameters as an interactive map to the users allowing them to choose a proper model or explore other combinations of evaluation criteria before responsibly finalizing their choice.

The Pareto frontier represents the optimal trade-offs between various fairness measures and accuracy in a set of models. Users can visually compare models on the Pareto frontier and choose the model that best fits their needs and preferences, depending on the specific application. For example, if the decision-making process disproportionately affects certain racial or gender groups, the preference may be for the model with the highest fairness (based on a specific metric), even if it comes at the expense of some accuracy. Conversely, in other applications, accuracy may be the more important consideration, and the model that provides the highest accuracy may be preferred even if it has lower fairness. Ultimately, the choice of model will depend on the specific application and the relative importance of each objective.

Technically speaking, the core idea underlying FairPilot is that the space of the model/hyperparameter combinations is independent of the space of model evaluation criteria. This enables designing a grid-search algorithm over the full factorial of the model and hyperparameters during the *preprocessing time* and collecting (and indexing) comprehensive information that creates an interactive environment for the user exploration phase.

The paper is structured as follows. Section 2 provides a brief summary of the existing related studies. Section 3 provides technical background on Pareto Frontier Multi-objective optimization problems and how it is adapted for the FairPilot context. In sections, 4 and 5, the architecture and user interface options provided by FairPilot are explained. Section 6 presents a case study on the *ELS* dataset to explore the hyperparameter space with FairPilot for model selection.

2 Related Work

The democratization and accessibility of machine learning have been a rapidly growing area of research in recent years. Many scholars have investigated methods and tools to make machine learning more accessible to non-experts and to promote more equitable access to the benefits of this technology. An open-source machine learning platform that allows non-experts to build and deploy predictive models has been proposed in [28]. Many others have proposed visualization techniques to help users better understand and interpret machine learning results [8]. Another area of research has been the development of automated machine learning (AutoML) tools, which can automate many of the tasks typically performed by machine learning experts. AutoML tools, such as those presented in [11], can make it easier for non-experts to create effective ML models without needing to understand the complexities of the underlying algorithms and techniques.

Most ML models feature a set of hyperparameters that must be predefined for training, and the choice of these hyperparameters significantly affects the model's ability to make correct predictions [9, 30]. Therefore, several studies focus on constructing grid or random search strategies [6, 31], or optimizing for the set of hyperparameters using sequential learning techniques (e.g., Bayesian optimization) [27, 31]. However, an accurate model can be biased and unfair in prediction since inherent trade-offs exist in using ML for decisions that impact social welfare.

Fairness in ML involves a growing body of research dedicated to identifying and addressing biases in algorithms [5, 22]. Fair-ML promotes the idea that algorithms should not produce biased or discriminatory outcomes and that their decisions should be impartial and equitable for all individuals, groups, or intersectional subgroups [12, 15]. On the other hand, a fair model can be inaccurate and useless in practice. Therefore, several studies aimed at designing interventions to balance the trade-off between fairness and accuracy in predictive modeling tasks [16, 19, 32].

One resolution to handle the fairness and accuracy trade-off is attainable by designing a fairness-aware hyperparameter search approach. A fair hyperparameter-tuning tool promotes consideration of fairness in the choice of hyperparameter for model selection by measuring the fairness metrics once the model has been trained and optimized for accuracy. [26] considers algorithmic policies such as hyperparameters for balancing the social welfare and private objective (e.g., profit) based on individual fairness. [25] designs a generally constrained Bayesian optimization framework and reveals that accurate and fair solutions are achievable by acting solely on the hyperparameter. [10] provides a simple and flexible intervention to incorporate fairness objectives in ML pipelines by proposing the fair variants of hyperparameter optimization algorithms such as Fair Random Search, Fair TPE, and Fairband.

3 Preliminaries

Fairness-aware exploration on the space of hyperparameters of machine learning models can be considered as a Multi-objective optimization problem where accuracy and fairness are conflicting objectives.

Multi-objective optimization [21,23] considers problems with more than one objective function to be optimized simultaneously. Two popular directions for multi-objective optimization are 1) combining the optimization criteria in form of a single function (Equation 3.1) and 2) identifying the Pareto frontier.

$$\max_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x}) \tag{3.1}$$

In Equation 3.1 $\mathcal{X} \in \mathbb{R}^d$ is the bounded search space such that $\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U, \forall \mathbf{x} \in \mathcal{X}$ with $\mathbf{x}^L$ and $\mathbf{x}^U$ be the coordinate-wise lower and upper bounds, and $\mathbf{f} : \mathcal{X} \to \mathbb{R}^M$ is the vector function of the M multiple objectives, $\mathbf{f} = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))$.

In the context of FairPilot, the hyperparameter tuning problem can be considered as a black-box function $\mathbf{f}(\mathbf{x})$, where the accuracy $f_A(\mathbf{x})$ and fairness $f_F(\mathbf{x})$ objectives are defined over a set of input hyperparameters $\mathbf{x} \in \mathcal{X}$. This implies that the task of selecting the optimal model requires considering multiple objectives, each with its own set of hyperparameters, making it a complex optimization problem. Due to the conflicting nature of objectives, there is no unique solution that optimizes all combination functions $\mathbf{f}$. In such environments specifying a meaningful function $\mathbf{f}$ is challenging and often not practical for ordinary users.

Therefore, our aim is to find a set of equally desirable solutions that hold a trade-off between the objectives, known as *Pareto Frontier*. The Pareto set contains only dominant (aka non-dominated) solutions that cannot be strictly dominated by any other solutions in at least one objective [13].

Suppose we aim to explore the space of hyperparameters for solving a binary classification task where the target variable $Y \in \{0,1\}$ defines 1 for positive, and 0 for negative outcomes, and $S \in \{0,1\}$ is the sensitive attribute (e.g., race) represent two different population subgroups (e.g., White and Black). Now, for a given set of input hyperparameters, the accuracy metric can be simply defined as the fraction of correct classification predictions or $f_A(\mathbf{x}) = P(\hat{Y} = 1|Y = 1) + P(\hat{Y} = 0|Y = 0)$. Moreover, there are several fairness metrics that can be used to measure bias in predictive modeling for binary classification tasks such as *statistical parity*, *equal opportunity*, and *predictive parity*. For instance, *statistical parity* metric is of the form $f_F(\mathbf{x}) = P(\hat{Y} = 1|S = 0) - P(\hat{Y} = 1|S = 1)$. Detailed explanations and formulas of various fairness metrics are provided in several fair ML studies [22, 29].

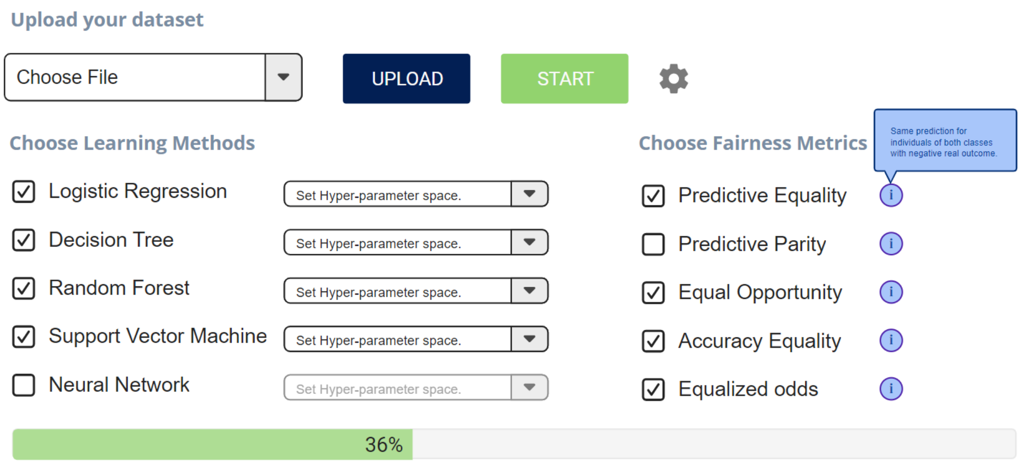
FairPilot is an interactive exploration tool designed to find non-dominated hyperparameters given a dataset and user-defined model and fairness metric choices. FairPilot outputs the set of non-dominated hyperparameters, $\mathcal{P} = \{\mathbf{x}^{(i)} \in \mathcal{X} | \nexists \mathbf{x}^{'} \in \mathcal{X}, f_F(\mathbf{x}^{'}) \leq f_F(\mathbf{x}^{(i)}), f_A(\mathbf{x}^{'}) \geq f_A(\mathbf{x}^{(i)})\}$ that can be employed to train models in a fairness-aware manner. For each ML model, $\mathcal{P}$ reveals the set of hyperparameter values that can be used to train the model such that is strictly better than other models based on at least one of the unfairness f_F or accuracy f_A metrics.

4 System Specification

4.1 Architecture and Implementation FairPilot is a web-based interactive system that explores the influence of hyperparameters on the trade-off between accuracy and fairness in predictive modeling. The goal is to support users in selecting the optimal combination of hyperparameters for training a model considering the fairness of the prediction outcome. FairPilot automatically trains models on a discretized space of user-defined hyperparameters and constructs the Pareto frontier by considering both accuracy and fairness metrics. The Pareto frontier contains the non-dominated solutions where the set of hyperparameters has produced the best results either from an accuracy or fairness perspective. FairPilot is developed using *Python 3.10.9*, and the web interface is created using *Flask* framework. The system generates interactive plots using the *Plotly* library.

5 User Interface

The following section describes the main components of the FairPilot user interface (UI).



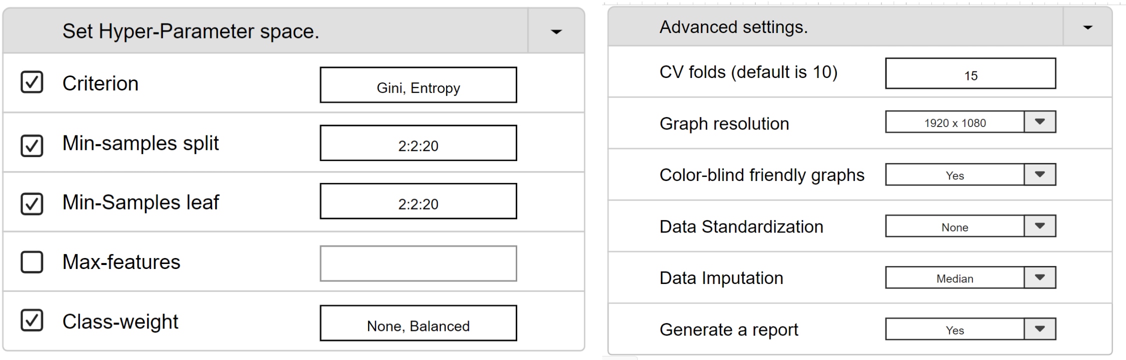
☑ Logistic Regression — Set Hyper-parameter space.
☑ Decision Tree — Set Hyper-parameter space.
☑ Random Forest — Set Hyper-parameter space.
☑ Support Vector Machine — Set Hyper-parameter space.
□ Neural Network — Set Hyper-parameter space.

☑ Predictive Equality ⓘ
□ Predictive Parity ⓘ
☑ Equal Opportunity ⓘ
☑ Accuracy Equality ⓘ
☑ Equalized odds ⓘ

Same prediction for individuals of both classes with negative real outcome.

36%

Figure 1: FairPilot User Interface.



Set Hyper-Parameter space.

☑	Criterion	Gini, Entropy
☑	Min-samples split	2:2:20
☑	Min-Samples leaf	2:2:20
□	Max-features	
☑	Class-weight	None, Balanced

Advanced settings.

CV folds (default is 10)	15
Graph resolution	1920 x 1080
Color-blind friendly graphs	Yes
Data Standardization	None
Data Imputation	Median
Generate a report	Yes

(a) Selection of the Hyperparameter Space.

(b) Advanced settings.

Figure 2: Drop down menus in the input sections.

5.1 Dataset Selection Section. The UI for FairPilot is shown in Figure 1. To begin using FairPilot, the user must first specify the dataset of interest. The dataset consists of a set of predictor variables and a categorical response variable. Once the dataset is uploaded, the user must select the target variable. The user is then asked to specify the sensitive attributes.

Learning Methods Section. Users can select from various learning models. The tool can handle a variety of machine learning models, ranging from basic models to more complex ones like Neural Networks.

Hyper-Parameter Space Section. This section is displayed as a dropdown menu, as shown in Figure 2(a), where the user can manually input the hyperparameter space or use the default values. It is worth noting that the set of hyperparameters varies for each model, and our system provides default values for each. The backend of our system performs feasibility checks on user-defined values, and in case of improper range definition, a notification is sent to the user.

Fairness Metrics Section. The user proceeds to the Fairness Metrics Section, where they can choose the fairness metrics of interest for their application. By clicking on the "info" button, a brief description of each metric is provided. In our system, fairness is defined with regard to sensitive attributes, which describe social characteristics of individuals that are considered private, and can potentially result in discrimination when used in decision-making. These attributes are often related to aspects of an individual's identity, such as race, gender, and age. Our tool can handle a broad range of fairness metrics, however, based on our previous research, certain group fairness metrics are highly positively or negatively correlated, while others are completely orthogonal [1].

Advanced Settings Section. Before performing the exploration, the user can define additional options by clicking on the gear wheel icon, as shown in Figure 2(b). These settings include:

- The number of cross-validation folds to use for each combination of hyperparameters.
- The resolution of the plots in case of export.
- The option to generate color-blind-friendly plots.
- The ability to perform various data standardization and imputation pre-processing techniques.
- An automated report generation for the results.

FairPilot's default setting considers 5-fold cross-validation, average resolution, color-friendly plots, standard scalar, and removing NA values for data pre-processing with automated report generation enabled.

Loading Bar Section. After the user finishes setting up the desired options for the learning process, they can run the FairPilot tool by clicking on the Start button. At this point, a loading bar will appear at the bottom of the page, as shown in Figure 1, to provide an estimate of the exploration progress. We utilize the *tqdm* Python library to display the loading bar.

5.2 Output Sections. Once the learning process is complete, FairPilot generates the outputs as listed below. It is worth noting that the plots presented in this paper are resulted from analyzing the ELS dataset, which we shall elaborate on later in §6.

- **Individual Model Pareto Frontiers.** FairPilot generates interactive plots that display an estimated Pareto frontier for each combination of learning methods and fairness metrics. An example is provided in Figure 3. By hovering over any point on the plot, the user can see the exact hyper-parameter combination that produced that performance. Furthermore, colors, symbols, and sizes, as shown in Figure 3(e), can be used to assist with interpretations (e.g., which hyper-parameters have the most significant impact on the final performance), and an interactive legend allows the user to focus on specific hyperparameters.
- **Multi-Model Pareto Fronts.** FairPilot generates an interactive plot for each fairness metric that displays an estimated Pareto frontier for all of the ML methods combined, allowing the user to check the overall performance of the framework. Similar to the Individual Model Pareto Frontiers, the user can hover over any point on the plot to see the hyper-parameter combination that produced that performance. Different colors are used to identify different ML methods, and an interactive legend allows the user to focus on specific methods. An example is provided in Figure 3(b).
- **Superimposed Pareto Frontiers** FairPilot produces an interactive graph that superimposes the Pareto frontiers for each ML method, given a fairness metric. This graph allows the user to understand which learning method is the most promising overall and if any one method dominates the others. Figure 3(c) provides a preview of this feature. A similar plot can be generated for an individual machine learning (ML) model using all fairness metrics. This interactive plot will display the estimated Pareto frontier for the ML model and allow the user to explore the impact of different fairness metrics on the model's performance.
- **Superimposed Individual Model Pareto Frontiers** Given a single ML method, FairPilot produces an interactive graph that superimposes the Pareto frontiers for multiple fairness metrics. This can be used when the user is interested in more than one metric at a time. A preview is given in 3(d).
- **Pareto Data Frames.** FairPilot generates data frames for each fairness metric that include all the Pareto points and their corresponding hyperparameter combinations. Additionally, the data frames allow the user to check the value of other fairness metrics (for example, on the data frame showing the Pareto points for the Predictive Equality metric, the user can also check the corresponding Equalized Odds values). Table 2 provides a preview of this feature.

Our system offers several interactive features that allow users to customize the visualizations and explore the data in more detail. In addition to the pop-ups of information when hovering over data points, users can select groups of points on the legend to remove them from the visualization. They can also modify the hyperparameters that determine the size, color, and shape of the points in real time, allowing them to gain more insight into how changes to the hyperparameters affect model performance. Furthermore, our system offers basic functionalities such as panning, zooming, fullscreen view, and the ability to download, which allow users to interact with the visualizations and explore the data in different ways.

6 Case Study

Dataset. In this section, we present a case study to demonstrate how FairPilot can be used in practice. We utilize the Education Longitudinal Study of 2002 (ELS) dataset [1]), which is a longitudinal study designed to provide trend data about critical transitions experienced by students. For our analysis, we consider the *highest level of degree* as the target variable and create a binary classification task to predict whether an individual's highest degree earned is above or below a bachelor's degree. To be specific, we assign a label of 1 to students who have obtained a college degree (i.e., a bachelor's degree or higher) as the favorable outcome, while students who have not achieved a college degree are assigned a label of 0 as the unfavorable outcome. We consider the sensitive attribute of "race", which includes five groups of students: White, Black, Hispanic, Asian, and Multi-racial (MR). To evaluate fairness, we divide this attribute into two sub-categories: category 0 comprising Black, Hispanic, and MR students, and category 1 comprising Asian and White students. This separation is based on precedent analysis of the ELS database in [2]. Data cleaning is performed to identify and rename the missing values (based on the documentation) and remove the observations that have many missing attributes (> 75% of the attributes are missing). The non-categorical data is then standardized by subtracting the mean and scaling to unit variance, but no imputation is performed. At this point, pre-processing is considered concluded and it is possible to proceed to model testing and training.

Experimental setup. We evaluated four ML methods on the dataset: Decision Tree Classifier (DT), Random Forest Classifier (RF), Logistic Regression (LR), and Support Vector Classifier (SVC). Due to the need for interoperability in the education domain, we did not use Neural Network (NN), even though it is available in our tool. The hyperparameters used for each ML method are presented below along with a brief description.

Decision Tree Classifier: The `criterion` is a function used to measure the quality of a split relative to the ideal case of perfect separation. The `min sample split` is the minimum number of samples required to split an internal node. The `minimum sample leaf` is the minimum number of samples required for a node to become a leaf. The `max features` is the number of features to consider when looking for the best split. The `class weight` is used to assign different weights to the classes in the dataset during the training process.

Random Forest Classifier: Same hyperparameters of the Decision Tree Classifier are used, with the addition of `bootstrap`, which is a boolean attribute that determines whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

Logistic Regression: The hyperparameters used in logistic regression are `penalty`, which specify the norm of the penalty, and `C`, which is the inverse of regularization strength.

Support Vector Classifier: Two main hyperparameters in SVC are `C`, which is the inverse of the regularization strength, and `kernel`, which specifies the kernel type to use in the SVC algorithm.

Table 1 shows the range of the attributes that define the hyperparameter space for different considered ML models.

Table 1: Hyperparameters Space

Model	Hypeparameter	Range
Decision Tree	criterion	[gini, entropy]
	max features	[None, sqrt, log2]
	min samples split	[2, 4, 8, 12, 16, 20]
	min samples leaf	[1, 4, 8, 12, 16, 20]
	class weight	[None, balanced]
Random Forest	criterion	[gini, entropy]
	max features	[None, sqrt, log2]
	min samples split	[2, 4, 8, 12, 16, 20]
	min samples leaf	[1, 4, 8, 12, 16, 20]
	class weight	[None, balanced]
	bootstrap	[False, True]
Logistic Reg.	C	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
	penalty	[l2, none]
SVC	C	[0.001, 0.01, 0.1, 1, 10, 100, 1000]
	kernel	[linear, poly, rbf, sigmoid]

Evaluation. In addition to *accuracy*, we evaluate the performance of our models using five fairness metrics, namely: *predictive parity*, *predictive equality*, *equal opportunity*, *accuracy equality*, and *equalized odds*. The evaluation is conducted using 10 different data splits for training and testing. We collect the mean and variance of both accuracy and fairness metrics across all 10 runs. For each ML model and hyperparameter combination, we train and test the models through a brute-force approach. We then construct the Pareto Front using the trained models using the two dimensions of fairness and accuracy. We use the Pareto Front to identify the optimal model settings.

Results. FairPilot is an assistive tool for exploring the modeling space and investigating the trade-off between accuracy and fairness. It serves two distinct objectives: model interpretation and model selection. For model interpretation, in Figure 3(a), which displays the *accuracy* vs equal opportunity Pareto front when using a Random

[1] For more information about ELS, visit the main website: `https://nces.ed.gov/surveys/els2002/avail_data.asp`

Forest. We can observe that the `bootstrap` is an influential hyperparameter in determining the performance of the model. When `bootstrap` =True, the fitted models generally have higher accuracy and lower *equal opportunity*. FairPilot provides a customizable color-coding and allows the user to select the hyperparameter of interest, resulting in reports that align with their perspective and facilitate a better interpretation of the results. In terms of model selection, the user can check the best possible configurations at a glance, by hovering on the Pareto front points as shown in Figure 3(e), for which we use the Decision Tree classifier and *predictive equality* for fairness evaluation. Alternatively, the user can examine various configurations on the corresponding Pareto Data Frame (as shown in Table 2). This makes it effortless and intuitive for the user to select the appropriate ML model for their specific application.

Figure 3(b) shows *accuracy* vs *predictive equality* Pareto front considering all different ML models. The results indicate that SVC achieves the best performance in terms of the *predictive equality*, while the Random Forest classifier tends to be more accurate. Once again, a clear trade-off is observed. Comparing the model with the highest accuracy against the model with the highest fairness, we have to sacrifice 10% in accuracy to achieve a 25% increase in fairness. The user has the option to choose any other point on the Pareto front, based on the application and the desired level of emphasis on accuracy and fairness.

In Figure 3(c), which displays the superimposed Pareto fronts of each ML method taken individually when considering *accuracy* vs *equalized odds*, we can observe that no single ML method dominates over the others. SVC is capable of producing fairer results, while RF has better accuracy. However, a trade-off exists in each Pareto Front, indicating that there is no single best model for all situations.

Figure 3(d) shows the Pareto front of accuracy and fairness for the DT model using various fairness metrics. The results indicate that sacrificing accuracy for fairness is more achievable for the *equal opportunity* metric (yellow line) compared to other metrics. Notably, by comparing the endpoints of *equal opportunity* Pareto front, we observe that a 7% increase in fairness can be attained with a mere 3% loss in accuracy. However, the importance assigned to fairness and accuracy in the model selection process may differ for each user when considering other fairness metrics.

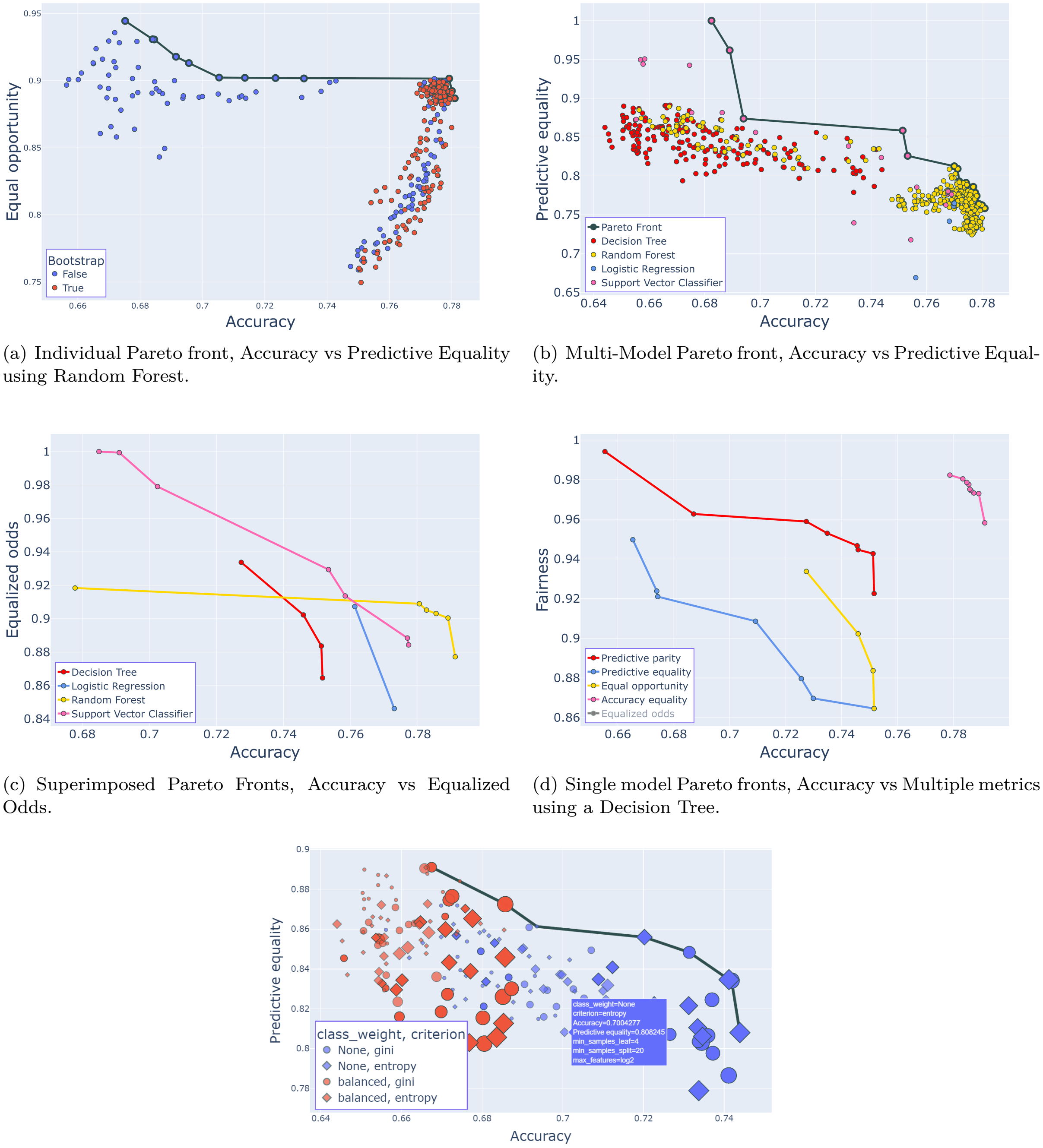
Figure 3(e) demonstrates the level of personalization, interactivity, and density of information provided by the plots in FairPilot. This graph allows the user to immediately observe the effects of three hyperparameters at the same time: `class weight` is represented using a color code, `criterion` is indicated using different symbols, and `min samples leaf` is shown by varying the size and transparency of the markers. By hovering over any point on the graph, the user access the exact configuration used, along with the corresponding objective values. As we can see, a 'balanced' class weight is able to obtain generally fairer models while accuracy is higher if we set the class weight as equal for both classes.

7 Conclusion and Future Work

This paper has presented FairPilot, an innovative solution for addressing the challenges associated with deploying machine learning models in high-risk decision-making domains while promoting fairness. FairPilot allows users to explore various models, hyperparameters, and fairness definitions and displays the Pareto frontier of models and hyperparameters as an interactive map. This unique combination of features offers users an opportunity to responsibly choose their ML models based on their application and objectives. FairPilot's ability to explore and visualize the Pareto frontier of models and hyperparameters enables users to make informed decisions and trade-offs between accuracy and fairness. The tool also enables users to explore the impact of various hyperparameters on model performance and fairness, which can be especially useful when dealing with complex models.

In the future, we plan to expand FairPilot's range of fairness definitions and integrate new models and hyperparameters. We aim to incorporate state-of-the-art algorithms such as Multi-Objective Bayesian Optimization (MOBO) to enable efficient hyperparameter optimization and to make FairPilot compatible with larger datasets. We also plan to enhance the tool's ability to deal with non-binary sensitive features and multiple sensitive features simultaneously. We believe that this will enable FairPilot to be more widely applicable in a range of decision-making domains.

Furthermore, we aim to improve the accessibility and ease of use of FairPilot for practitioners. This includes incorporating user-friendly interfaces and workflows that allow practitioners to interact with the tool easily and providing documentation and tutorials to guide them through the use of FairPilot. Overall, we believe that FairPilot has the potential to significantly impact responsible AI and decision-making practices by enabling informed decisions based on both model performance and fairness criteria.



(a) Individual Pareto front, Accuracy vs Predictive Equality using Random Forest.

(b) Multi-Model Pareto front, Accuracy vs Predictive Equality.

(c) Superimposed Pareto Fronts, Accuracy vs Equalized Odds.

(d) Single model Pareto fronts, Accuracy vs Multiple metrics using a Decision Tree.

(e) Individual Pareto Front, Demo of the interactive graph using Predictive Parity and a Decision Tree.

Figure 3: Output plots of FairPilot

References

[1] H. Anahideh, N. Nezami, and A. Asudeh, *On the choice of fairness: Finding representative fairness metrics for a given context*, arXiv:2109.05697, (2021).

[2] H. Anahideh, N. Nezami, and D. G'andara, *Auditing fairness and imputation impact in predictive analytics for higher education*, 09 2021.

[3] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al.,

Table 2: Pareto points for Accuracy vs Predictive Parity, using a Decision Tree

Accuracy	**Pred. Par.**	**Pred. Eql.**	**Eql. Opp.**	**Acc. Eql.**	**min_leaf**	**min_split**	**max_features**	**Obj.**	**clf. weights**
0.655	0.741	0.741	0.754	0.950	16	2	log2	entropy	balanced
0.687	0.801	0.801	0.799	0.944	20	2	None	entropy	balanced
0.727	0.788	0.788	0.934	0.987	8	20	log2	gini	None
0.735	0.777	0.777	0.868	0.955	12	2	log2	entropy	None
0.746	0.819	0.819	0.876	0.950	20	2	None	entropy	None
0.746	0.831	0.831	0.902	0.959	20	2	sqrt	entropy	None
0.751	0.828	0.828	0.884	0.949	20	2	None	gini	None
0.752	0.865	0.865	0.865	0.924	12	2	None	gini	None

Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information fusion, 58 (2020), pp. 82–115.

[4] A. Asudeh, *Enabling responsible data science in practice*, ACM SIGMOD Blog, (2021).

[5] S. Barocas, M. Hardt, and A. Narayanan, *Fairness in machine learning*, NIPS, (2017).

[6] J. Bergstra and Y. Bengio, *Random search for hyper-parameter optimization.*, JMLR, 13 (2012).

[7] D. Bertsimas, V. F. Farias, and N. Trichakis, *On the efficiency-fairness trade-off*, Management Science, 58 (2012), pp. 2234–2250.

[8] A. Chatzimparmpas, R. M. Martins, I. Jusufi, and A. Kerren, *A survey of surveys on the use of visualization for interpreting machine learning models*, Information Visualization, 19 (2020), pp. 207–233.

[9] M. Claesen and B. De Moor, *Hyperparameter search in machine learning*, arXiv:1502.02127, (2015).

[10] A. F. Cruz, P. Saleiro, C. Belém, C. Soares, and P. Bizarro, *Promoting fairness through hyperparameter optimization*, in ICDM, 2021, pp. 1036–1041.

[11] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, *Efficient and robust automated machine learning*, NurIPS, 28 (2015).

[12] J. R. Foulds, R. Islam, K. N. Keya, and S. Pan, *An intersectional definition of fairness*, in ICDE, IEEE, 2020, pp. 1918–1921.

[13] V. Furlan et al., *Vilfredo pareto, manuale di economia politica*, Journal of Economics and Statistics, 91 (1908), pp. 826–831.

[14] C. Giordano, M. Brennan, B. Mohamed, P. Rashidi, F. Modave, and P. Tighe, *Accessing artificial intelligence for clinical decision-making*, Frontiers in digital health, 3 (2021), p. 645232.

[15] P. G. John, D. Vijaykeerthy, and D. Saha, *Verifying individual fairness in machine learning models*, in AUAI, PMLR, 2020, pp. 749–758.

[16] F. Kamiran and T. Calders, *Data preprocessing techniques for classification without discrimination*, Knowledge and information systems, 33 (2012).

[17] J. Kleinberg, S. Mullainathan, and M. Raghavan, *Inherent trade-offs in the fair determination of risk scores*, arXiv:1609.05807, (2016).

[18] A. Lewis and J. Stoyanovich, *Teaching responsible data science: Charting new pedagogical territory*, IJAIED, (2021), pp. 1–25.

[19] S. Liu and L. N. Vicente, *Accuracy and fairness trade-offs in machine learning: A stochastic multi-objective approach*, CMS, 19 (2022), pp. 513–537.

[20] S. Lo Piano, *Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward*, Humanit. soc. sci., 7 (2020).

[21] R. T. Marler and J. S. Arora, *Survey of multi-objective optimization methods for engineering*, Struct Multidiscipl Optim., 26 (2004), pp. 369–395.

[22] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, *A survey on bias and fairness in machine learning*, CSUR, 54 (2021), pp. 1–35.

[23] P. Ngatchou, A. Zarei, and A. El-Sharkawi, *Pareto multi objective optimization*, in ISAP, 2005.

[24] Y. Nieto, V. Gacía-Díaz, C. Montenegro, C. C. González, and R. G. Crespo, *Usage of machine learning for strategic decision making at higher educational institutions*, IEEE Access, 7 (2019).

[25] V. Perrone, M. Donini, M. B. Zafar, R. Schmucker, K. Kenthapadi, and C. Archambeau, *Fair bayesian optimization*, in AIES, 2021.

[26] E. Rolf, M. Simchowitz, S. Dean, L. T. Liu, D. Bjorkegren, M. Hardt, and J. Blumenstock, *Balancing competing objectives with noisy data: Score-based classifiers for welfare-aware machine learning*, in ICML, PMLR, 2020, pp. 8158–8168.

[27] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon, *Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020*, in NeurIPS, PMLR, 2021, pp. 3–26.

[28] R. Van De Schoot, J. De Bruin, R. Schram, P. Zahedi, J. De Boer, F. Weijdema, B. Kramer, M. Huijts, M. Hoogerwerf, G. Ferdinands, et al., *An open source machine learning framework for efficient and transparent systematic reviews*, Nature machine intelligence, 3 (2021), pp. 125–133.

[29] S. Verma and J. Rubin, *Fairness definitions explained*, in FairWare, 2018, pp. 1–7.

[30] H. J. Weerts, A. C. Mueller, and J. Vanschoren, *Importance of tuning hyperparameters of machine learning algorithms*, arXiv:2007.07588, (2020).

[31] T. Yu and H. Zhu, *Hyper-parameter optimization: A review of algorithms and applications*, arXiv:2003.05689, (2020).

[32] I. Žliobaitė, *On the relation between accuracy and fairness in binary classification*, in FAT/ML workshop at ICML, vol. 15, 2015.