Bias Aware GridSearch: Integrating Fairness Into Model Selection

Benjamin Chen bhc001@ucsd.edu
Jayson Leach jleach@ucsd.edu
Stephanie Chavez stchavez@ucsd.edu
Anika Garg agarg@ucsd.edu

Emily Ramond eramond@deloitte.com

Abstract

As Artificially Intelligent algorithms become increasingly influential in every aspect of our daily lives, unaddressed biases within these models have the potential to drive unethical decision-making, such as amplifying discrimination against marginalized communities. Efforts to address these concerns have surfaced through tools designed to detect and mitigate bias, which predominantly focus on correcting biases after the model's development phase, leaving limited solutions for incorporating ethical considerations more comprehensively during the model development process. We aimed to address this gap by creating the BiasAwareGridSearch (BAGS) package, a modified version of sklearn's GridSearchCV, that integrates an additional bias detection layer directly into the hyperparameter tuning step. BAGS replaces GridSearch and other tuning methods, seamlessly integrating bias consideration into the standard machine learning process. While BAGS alone did not consistently result in significant bias reduction for our models, we found a correlation between increasing accuracy and decreasing bias, and determined that specific hyperparameters play a large role in bias minimization while others have no effect. Paired with other bias mitigation techniques, our package is a useful tool for developers to further de-bias models by exploring how chosen hyperparameters affect their results.

1 Introduction

As Artificial Intelligence (AI) systems have become increasingly embedded in our daily lives, from diagnosing medical diseases to performing financial risk assessments, the ethical implications of these technologies have gained significant attention. Central to these concerns is the issue of bias in decision making, where algorithms may inadvertently reproduce and amplify societal biases such as racial and gender discrimination.

Recent advancements in AI ethics have led to the development of various methodologies and tools aimed at addressing these ethical challenges. A notable example is IBM's AI Fairness 360 (AIF360) toolkit, which provides a comprehensive collection of various metrics and algorithms to detect and mitigate bias in machine learning models (see A.1). While such tools demonstrate significant progress in the field, they overlook the crucial phase of model training and hyper-parameter optimization.

This gap in the existing landscape of AI ethics solutions forms the basis of our research. Our work integrates ethical considerations directly into the hyperparameter optimization process, a novel approach in the realm of machine learning. Traditional hyperparameter tuning methods, such as those employed by sklearn's GridSearchCV, solely target performance optimization, with no regard for the ethical implications of the resulting models. In contrast, our modified version of GridSearchCV introduces an additional layer of bias consideration, ensuring that ethical parameters are not an afterthought but a fundamental aspect of the model training process.

Our findings indicate that certain hyperparameters have a linear impact on bias, while others have no impact on bias at all. Furthermore, as parameters are adjusted to achieve better accuracy, the bias exhibited also decreases as a result, demonstrating that there is a correlation between increasing accuracy and decreasing bias.

Ultimately, by embedding ethical considerations into the core of hyperparameter optimization, our methodology offers a more proactive approach to fostering fairness in AI and paves the way for other innovative approaches to mitigating bias.

2 Literature Review

As previously noted, machine learning (ML) models are increasingly being used for decision-making in various fields. Alongside it, there is a similarly increasing concern regarding models exacerbating the biases present in data. Existing research has identified the intricate relationship between data, bias, and ML models Gebru et al. (2018). These relationships require careful consideration, as biased models can have profound societal impacts.

Real-world data, which is the foundation on which ML models are built, has bias. The biases in the data stem from historical, cultural, and societal influences. Their recognition has given rise to ethical considerations that extend beyond ML model development. It requires a broader lens and an examination of the societal implications of biased models, particularly when these models impact decision-making processes relied upon by marginalized

individuals and communities. Thus, Hildebrandt (2019) argues that the framing of bias is not merely a technical issue but a socio-political and ethical concern that requires careful consideration. The author discusses the potential inclusion of bias in every step of the machine learning pipeline, from data collection and pre-processing to model training and deployment.

To reduce the impact of bias in this process, researchers have proposed and developed various solutions, some of which we implemented. Bias identification metrics, such as disparate impact and statistical parity difference (Pagano et al. 2023), offer quantitative measures to assess the extent of bias in ML models. These metrics provide a portion of objectivity in evaluating the fairness of models, especially when comparing models to each other.

Additionally, bias mitigation strategies are employed throughout the ML model development process. These strategies are categorized as pre-processing, in-processing, and post-processing techniques. Pre-processing involves handling biased data before it enters the model. In-processing techniques adjust the learning algorithm to mitigate bias during training. Post-processing techniques focus on refining model outputs. Integration of these techniques ensures a comprehensive approach to addressing bias throughout the pipeline of model creation. The study by Mehrabi et al. (2021) tackles a comprehensive review of the current state, as of July 2021, of bias identification and mitigation methods. This included a variety of bias metrics and different pre-processing, in-processing, and post-processing mitigation techniques. Similarly, the aforementioned AI Fairness 360 (AIF360) toolkit is IBM's approach to ethical AI, containing numerous bias identification metrics and mitigation algorithms at all stages of model life cycles (see A.1).

Our BiasAwareGridSearchCV adds to the expanding catalog of mitigation algorithms, particularly as an in-processing step. It also provides insights into how decisions made during the hyperparameter tuning step can impact bias exhibited; developing strategies for enhancing the interpretability and explainability of ML models is essential to facilitate the identification and understanding of bias. Models that are more interpretable allow stakeholders to trace the decision-making process and identify potential sources of bias more effectively (Alelyani 2021).

Overall, mitigating bias in ML models presents its own set of challenges. The primary challenge revolves around the trade-off between fairness and performance (Chen et al. 2023). Finding the right balance is challenging, as aggressive bias mitigation may impact the overall accuracy and effectiveness of the model. It is crucial to acknowledge the impossibility of achieving a perfect model. Real-world data is inherently biased, and completely eliminating bias is an impractical goal. The key lies in creating models that are both ethical and accurate, finding the equilibrium between mitigating bias and maintaining model performance. Ultimately, interdisciplinary collaboration between computer scientists, ethicists, legal scholars, and policymakers is essential to develop comprehensive approaches to addressing bias in ML systems (Hildebrandt 2019).

3 Data

To empirically gauge our modified GridSearchCV model, we focused our investigation on four distinct ethics-impacted domains. These domains were selected for their diverse nature and the varying ethical challenges they present, serving as a detailed example for the application of our new model selection process.

We sourced the datasets for each domain from publicly available data. Each dataset contained more than 10,000 observations and supports classification tasks. To process our data, we performed exploratory data analysis to give us a thorough understanding of each individual dataset. Depending on the specific dataset, cleaning methods such as missing value handling and outlier detection were done at the discretion of the author who processed it.

3.1 Criminal Justice

Given the well-documented injustices present in the field of criminal justice, we chose to explore recidivism prediction data. Our focus centers on the use of the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) prediction tool, aiming to understand its biases and contribute insights into how algorithms can introduce more bias to the already unfair criminal justice system. The dataset used is infamous and consists of 2013-2014 COMPAS scores from the Broward County Sheriff's Office in Florida obtained by ProPublica (Larson et al. 2016). It consists of 18,610 observations and 46 features. These features include demographic data such as sex, age, race, criminal history such as number of times incarcerated, and COMPAS predictions such as decile score which is the risk of recidivism rating on a scale of one to ten. The goal of modeling this dataset is to predict the feature `is_recid`, which tracks whether or not a person committed a crime after receiving a COMPAS assessment. In the domain of criminal justice where historical bias is perpetuated, it is important to implement ways of identifying and mitigating bias, as to not create unfair models. It is for this reason that we applied and evaluated models on this data using our algorithm.

3.2 Mortgage Origination

In an effort to recognize the impacts of bias in mortgage origination, we used data made available to the public under the Home Mortgage Disclosure Act (HMDA). The dataset observes mortgage applications in 2017, resulting in 14,285,496 observations and 74 total features. For the purposes of our project, the data has been simplified to follow mortgage applications of conventional loans in California, while following 5 features. These features include demographic variables such as sex and race. The goal of the model is to predict, based on the applicant's given features, whether they would be approved for a mortgage. There is often a disparity within certain demographics being approved for loans, so bias can be factor in this decision, which makes this a relevant topic to use with our algorithm.

3.3 Finance/Income

Another domain rife with ethical issues is within the finance and income sector. Over the past decade, systemic bias in compensation has become increasingly evident, showing that, irrespective of experience, role, or expertise, female workers consistently receive lower pay than their male counterparts of similar caliber. In response to these concerns regarding bias respective to income, we chose to analyze the UCI Adult Income Dataset, which was donated by Barry Becker in 1996 and extracted from the 1994 Census database. This renowned dataset is a key resource in machine learning for examining how demographic and job-related factors influence income levels. It contains 48,842 records and 14 attributes, including essential factors like age, education, and occupation, with the goal of predicting whether an individual earns more than $50,000 per year. As this dataset includes sensitive information, is used for a binary classification problem, and represents a source of rich information regarding a bias-concerning topic, it serves as an excellent case for testing our algorithm.

3.4 Healthcare

To assess the impact of bias in healthcare, we utilized data from the Center for Disease Control (CDC) released in 2015, focusing on the diabetes diagnosis status of patients across the United States. The dataset contains information on 253,680 patients and has 23 features, ranging from smoking status and average physical activity levels to demographic details such as race. Our primary aim with this dataset is to develop a model capable of accurately predicting whether an individual falls into the pre-diabetic, diabetic, or nondiabetic category based on an array of lifestyle and medical variables. Recognizing the significance of investigating bias in healthcare diagnoses, we acknowledge its potential to introduce disparities in treatment and outcomes, particularly among certain demographic groups. Thus, this exploration aligns closely with the central issue our project aims to tackle.

4 Methods

4.1 Data Processing and Exploration

After ingesting our respective datasets, the first step of model development was to conduct exploratory data analysis and determine the distribution of demographic variables in each dataset. In the income, mortgage, and healthcare datasets, given the inadequate representation of each racial group, we focused our investigation on gender-based demographic bias, assessing disparate impact and statistical parity difference between males and females. For the recidivism dataset, our analysis targeted bias related to racial groups.

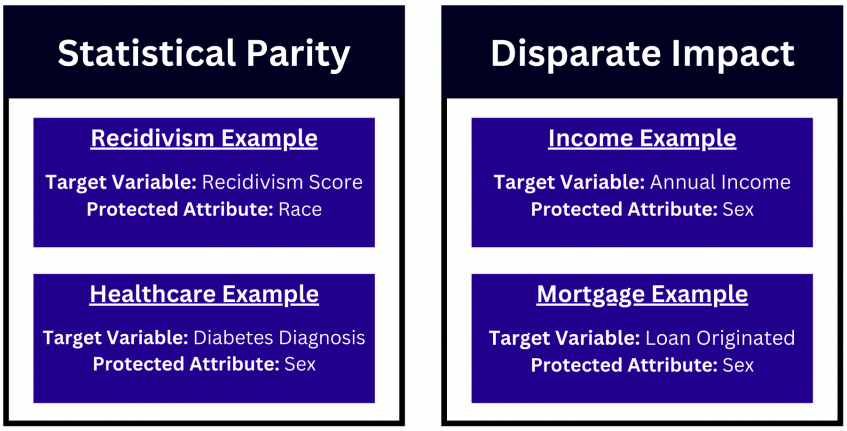

Figure 1: Selected target variables and protected attributes for each classification task

We determined that classification using Random Forest models was best suited for our model development tasks. Subsequently, we performed data preparation procedures, including handling missingness and the one-hot encoding of categorical variables, to ensure ease of training on our data.

4.2 Bias Aware GridSearchCV Development

Classic GridsearchCV uses (stratified) k-fold cross validation, which divides the dataset into 'k' subsets, training and validating the model 'k' times, each time using a different subset as the validation set and the remaining as the training set. The performance of the model is collected between folds, ensuring a holistic evaluation by using every data point for both training and validation, thereby providing a reliable estimate of the model's performance. To integrate bias in the model selection process, we added an additional bias evaluation layer to this process - each k-fold will collect both the model performance and its exhibited bias on that fold. By assessing bias using the same method as accuracy, we ensure a similar degree of thoroughness in measuring exhibited bias.

Since our work considers an bias layer, the process in selecting a "good" model is more complex than simply finding the most accurate one. Methods are included to support searching for a model that fits the users' particular criteria, including finding the least biased model, a balanced model that is the least biased model among models within an accuracy `margin` points away from the highest accuracy, and a balanced model that is the least biased model among the top `threshold` models. Alongside model selection, our work includes plotting functions to explore the relationship between each parameter and bias, and a plotting function to see the distribution of models with respect to exhibited bias and accuracy.

To support the seamless integration of our work into the predominant model selection process, we used the syntax and core features of sklearn's GridSearchCV, ensuring our work remain flexible and user-friendly. Additionally, we left the bias metric function an open argument - users can incorporate any bias metric as is appropriate for their domain with the only criteria being that 0 must be the value representing a fair state.

4.3 Running Bias Aware GridSearchCV

Although BiasAwareGridsearchCV can accept any model and any bias metric, for the purpose of consistency we constrained our work to consider a very specific subspace of the possible arrangements. Across all 4 domains, we maintained using RandomForestClassifier as our model of choice and only considered 1 of 2 bias metrics: disparate impact ratio and statistical parity difference (see A.2).

To standardize the process among our four datasets to make the results comparable, we decided on a single parameter grid, composed of the most relevant hyperparameters of RandomForest classifiers (see A.3).

```
param_grid = {'n_estimators': [50, 100, 200, 500],
 'max_depth': [20, 40, 50, 100],
 'min_samples_split': [2, 5, 10],
 'min_samples_leaf': [2, 5, 10],
 'bootstrap': [True, False],
 'criterion': ['gini', 'entropy'],
 'max_leaf_nodes': [20, 30, 40, 50, 100],
 'class_weight': ['balanced', 'balanced_subsample']}
```

To evaluate the performance of our novel GridSearch, we first developed naive Random Forest Classifiers for each using the regular GridSearch with the given parameter grid for consistency, and evaluated the training and testing accuracy and our chosen bias metrics. Due to the large size of the mortgage and healthcare datasets, running GridSearchCV and the modified BiasAwareGridSearchCV can take significant computational resources and time; for the purposes of reproducibility, a random sample representative of the population was used.

After running the novel GridSearch, we examined four different models the GridSearch produced: the most accurate model, the least biased model, the balanced model, and the optimum model. Training and testing accuracy was calculated along with the respective bias metric for each dataset. The ”best” model was determined by selecting a model that fulfills the criteria explained in the previous section.

5 Results and Discussion

5.1 Model Performance

Comparison between the performance of the Naive GridSearch and BiasAwareGridSearch indicates that there is no significant and consistent reduction in bias across all datasets after applying our algorithm. For our ”best” model (BAGS balanced model), disparate impact reduced by 0.005 for the income dataset and increased by 0.001 for the mortgage dataset, while statistical parity remained the same for the recidivism dataset and increased by 0.033 for the healthcare dataset. In the Mortgage and Recidivism examples, the BAGS least bi-

	Income	Mortgage	Recidivism	Healthcare
Naive GridSearchCV	0.722	0.083	−0.318	−0.003
BAGS Balanced	0.727	0.084	−0.318	−0.035
BAGS Least Biased	0.723	0.073	−0.305	−0.021
BAGS Highest Accuracy	0.728	0.086	−0.318	−0.035

Figure 2: Test Set Bias Metrics by Model Selection Criteria

ased selection criteria chose a model where bias improved slightly when compared to the naive GridSearchCV model. In the Income and Healthcare examples, bias slightly worsened throughout all BAGS selections when compared to the naive model. Overall, the differences between the final test bias metrics are minimal enough that they carry little evidence that BAGS made a notable impact on bias exhibited by the final model.

These changes in bias exhibited can be attributed to randomness and noise within the data, so it is clear that BAGS alone is not effective for bias mitigation; a more holistic approach is necessary. For example, the income dataset initially exhibited a significant disparate impact, particularly revealing a pronounced bias against predicting high income for females. In order to reduce the impact of this bias, data pre-processing techniques can be applied to adjust the weight or size of samples within the dataset, thereby ensuring a more balanced and proportional representation across demographic groups.

5.2 Examining Bias vs. Accuracy

By plotting the Random Forest classifiers produced by our novel GridSearch algorithm, we can examine the tradeoff between our chosen bias metric and accuracy for these models. The red line indicates the selected accuracy threshold for top models to consider when choosing the optimum model that balances bias and accuracy. The goal is to achieve an accuracy that approaches 1 while the bias approaches 0, indicating fairness.

With all four datasets, we can see that there's a correlation between increasing accuracy and decreasing bias - as we approached parameters that produced better accuracy, the bias exhibited also went down as a result. We also observed inherent clustering in accuracy, contrary to our initial expectation of a more continuous spread of models. This indicates that certain parameters play a significant role in influencing both accuracy and bias. The correlation appears appears to be stronger for the two datasets that measured bias using disparate impact, though the trend is consistently linear across all four datasets tested and both bias metrics.

In this context, it is clear that fairness emerges as a natural consequence of accuracy; a model that accurately reflects the underlying data distribution is inherently more likely to produce unbiased predictions across different demographic groups. When a model accurately learns from the data, it focuses on capturing the genuine relationships between features and outcomes, and is less dependent on false correlations and biases that may be present in the dataset.

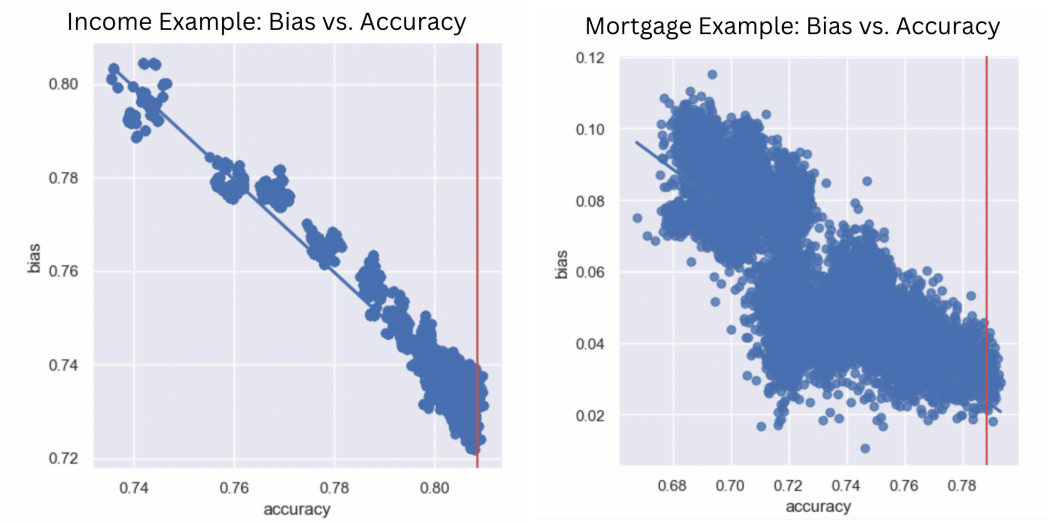

Figure 3: Tradeoff between accuracy and disparate impact for models produced

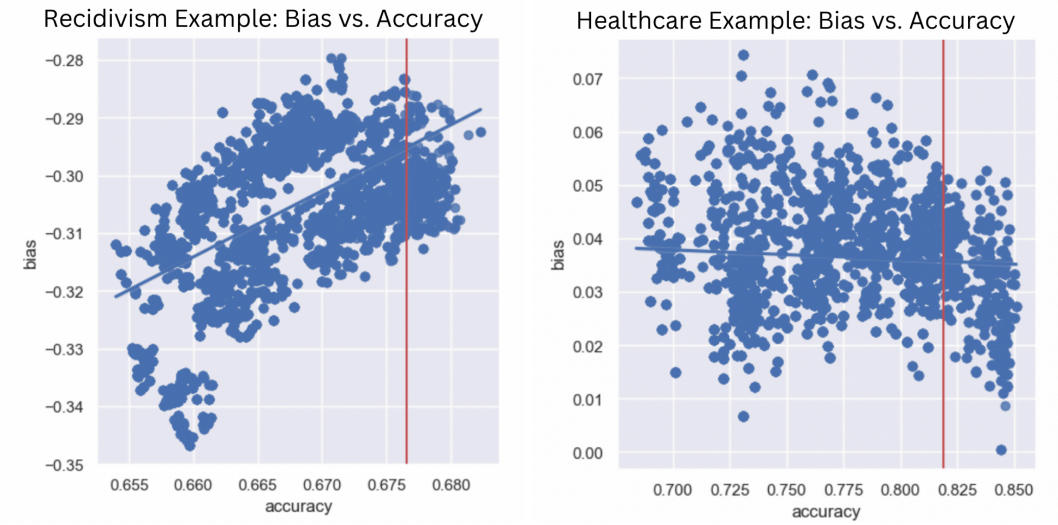

Figure 4: Tradeoff between accuracy and statistical parity difference for models produced

5.3 Exploring hyperparameters

Based on the relationship discovered between bias and accuracy, we created a plot_params function to examine how each hyperparameter specifically had an effect on the bias exhibited by our classifiers.

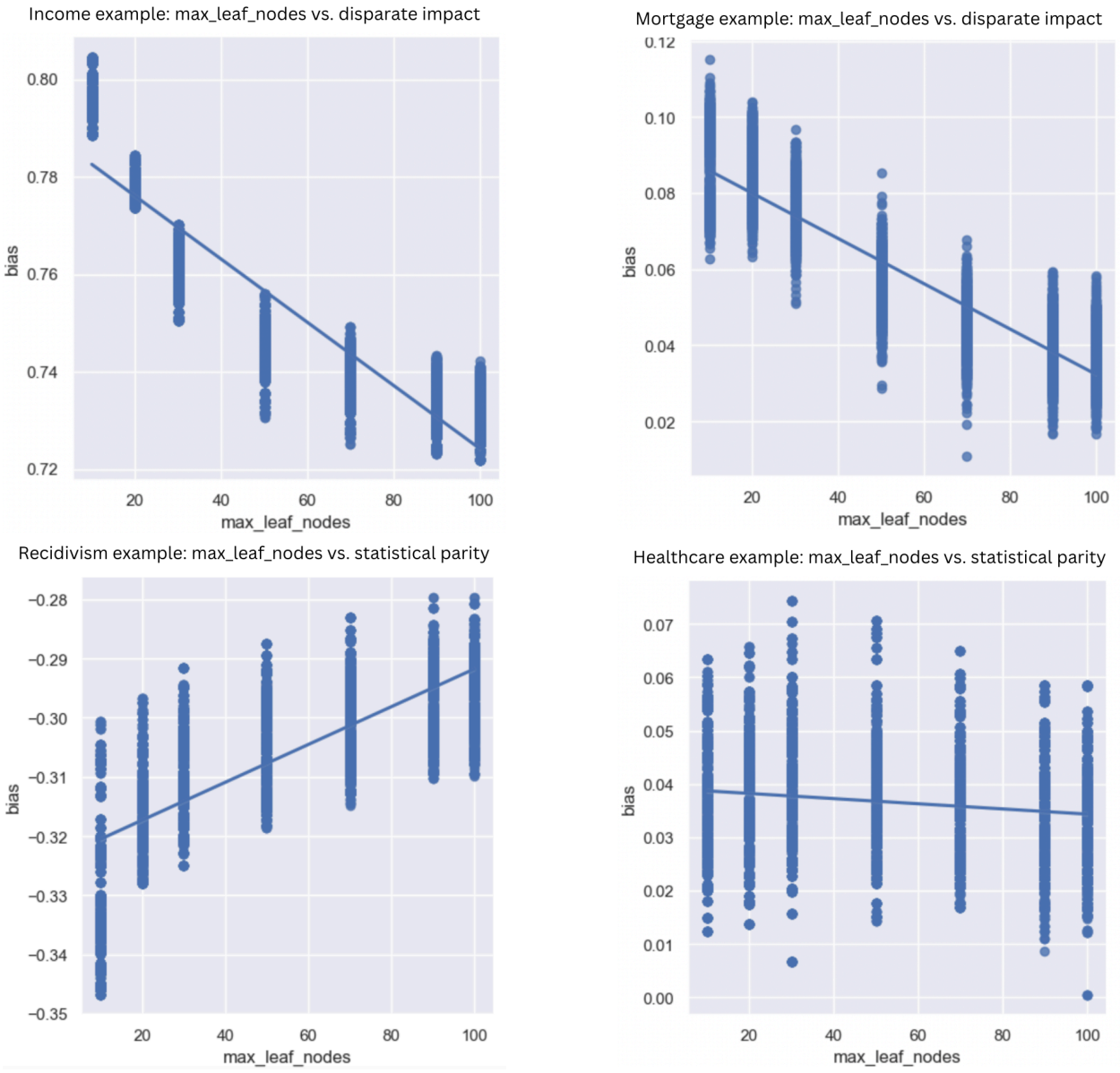

Figure 5: Examining impact of max_leaf_nodes hyperparameter on exhibited bias

The first parameter explored was max_leaf_nodes. We found that by varying the number of maximum leaf nodes in the decision trees, we observed notable changes in performance; higher values resulting in greater bias reduction across all four models. For the purposes of this analysis, our parameter grid tested max_leaf_nodes up to a value of 100, but given the strong positive trend between this parameter and the performance of the model, ex-

panding the grid to include values larger than our threshold would very likely yield higher performing models in terms of both bias and accuracy.

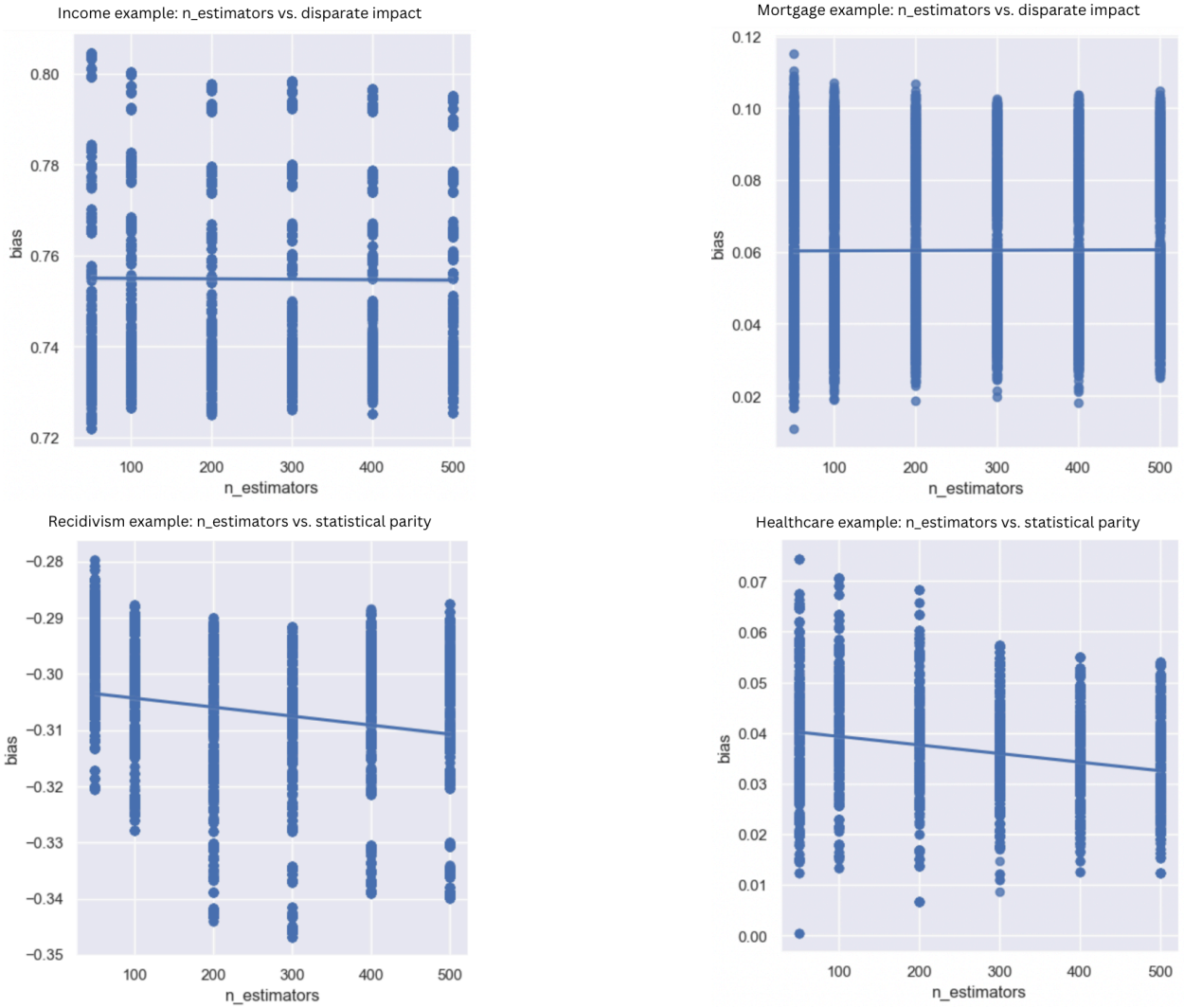

Figure 6: Examining impact of n_estimators hyperparameter on exhibited bias

For the models using disparate impact as the bias metric, n_estimators appeared to have no impact at all in mitigating bias exhibited. However, for the recidivism example using statistical parity, as the number of estimators approached 500, the parameter appeared to minimally worsen the bias of the model; this can potentially be attributed to overfitting or simply noise present within the data. For the healthcare example, the largest number of estimators improved the bias on an average of approximately 0.05 compared to the smallest number of estimators, which can indicate that increasing the n_estimators in this scenario leads to a better representation of the underlying patterns in the data.

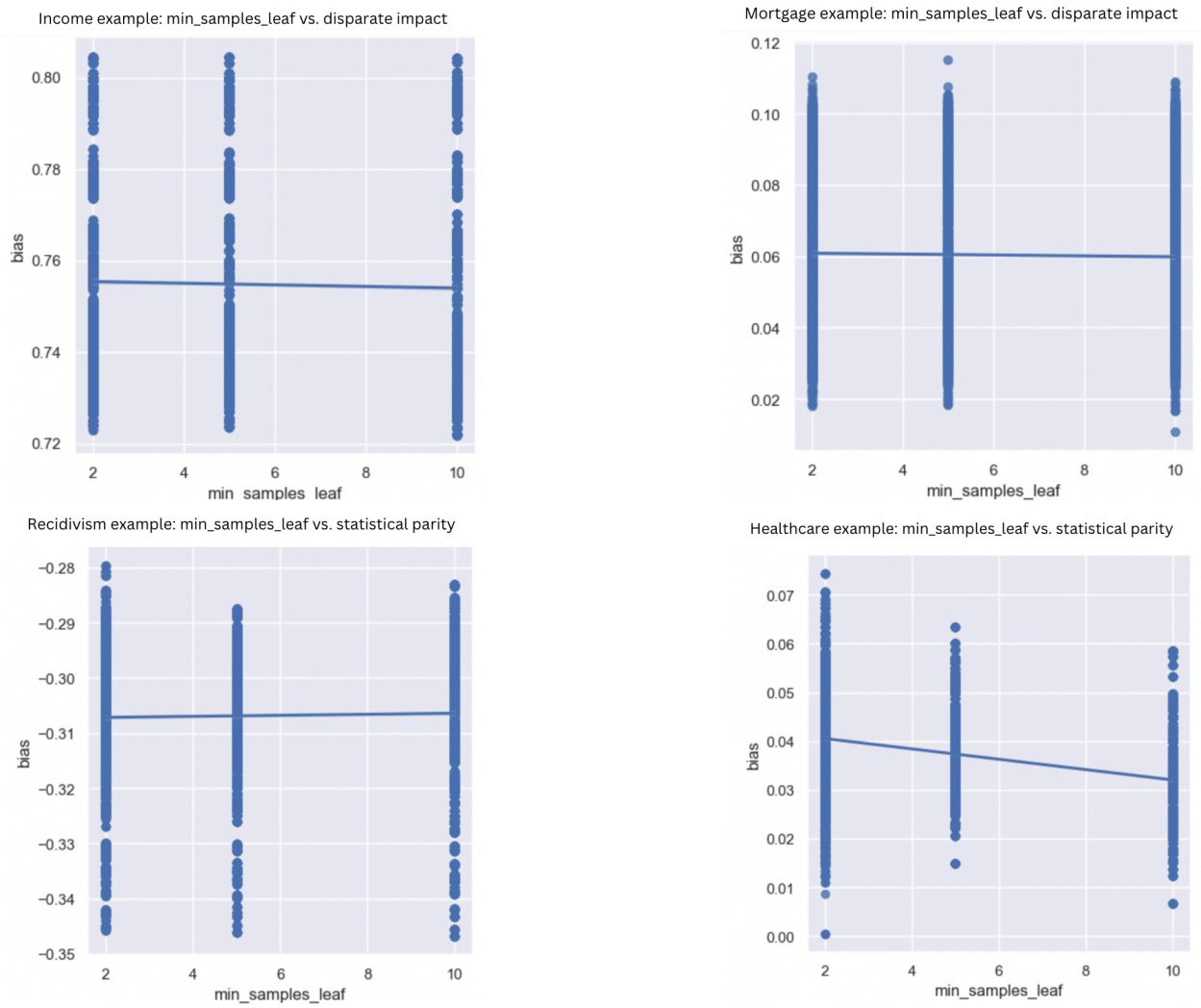

Figure 7: Examining impact of min_samples_leaf hyperparameter on exhibited bias

Adjusting the min_samples_leaf parameter had varying results our datasets. For the income dataset, the maximum number of min_samples_leaf contributed to a 0.005 reduction in bias on average compared to the smallest number of min_samples_leaf tested, and for the mortgage dataset, this parameter had no impact. In regards to statistical parity, min_samples_leaf contributed to a 0.005 reduction in bias for the recidivism dataset and a 0.008 reduction in bias for the healthcare dataset on average. These adjustments may be attributed to randomness and noise, but the results indicate that given a parameter testing grid with larger values of min_samples_leaf, a more linear relationship can potentially be uncovered.

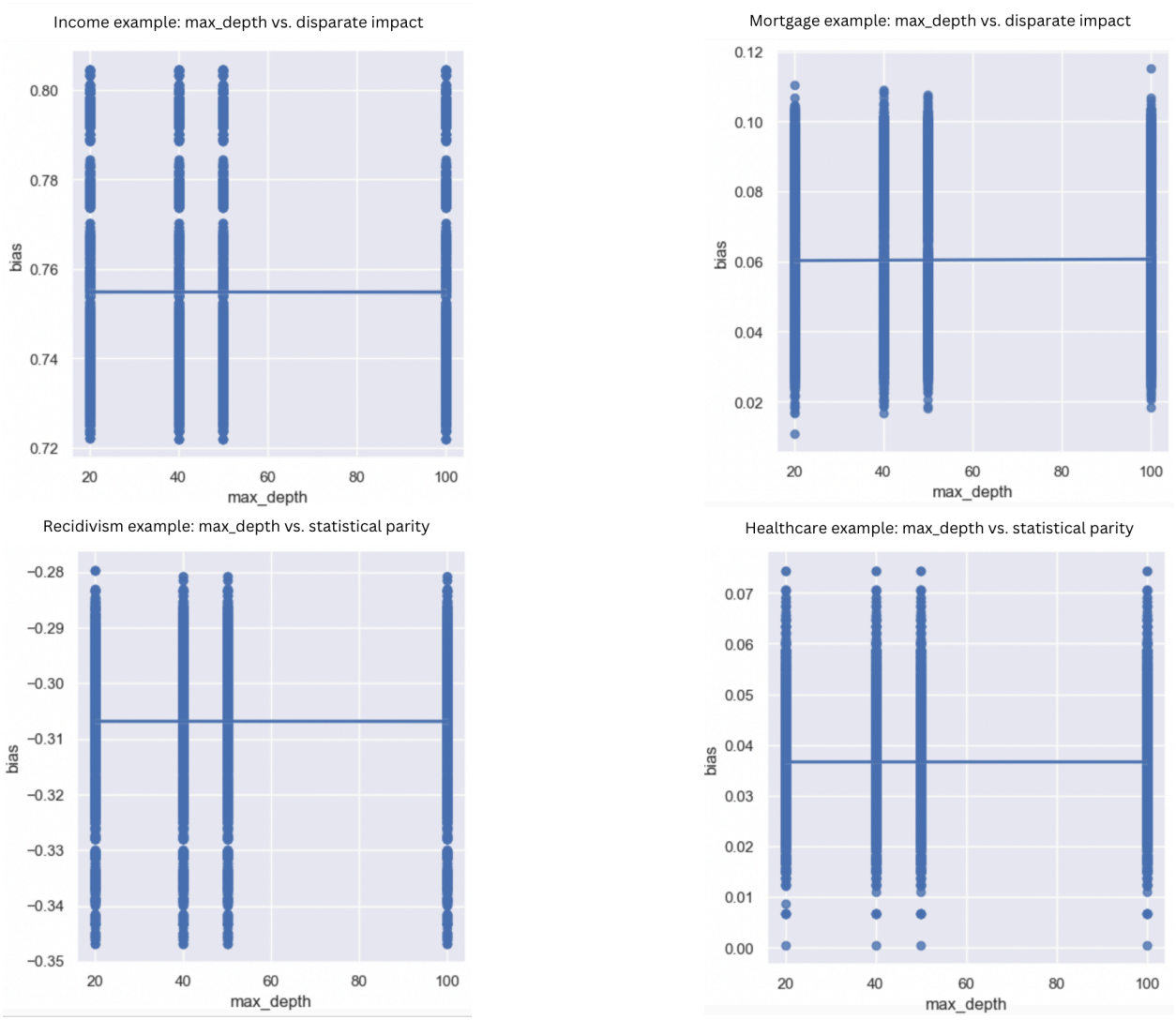

Figure 8: Examining impact of max_depth hyperparameter on exhibited bias

The maximum depth of the Random Forest Classifiers appears to have no impact on the bias exhibited by the model, given the parameter grid we tested. This is surprising because max_depth typically has a significant influence on model complexity and overfitting, which could lead to poorer performance when generalizing to unseen data (Hastie, Tibshirani and Friedman 2009). However, in this case it's possible that the ensemble learning process in random forests is less sensitive to changes in the maximum depth parameter.

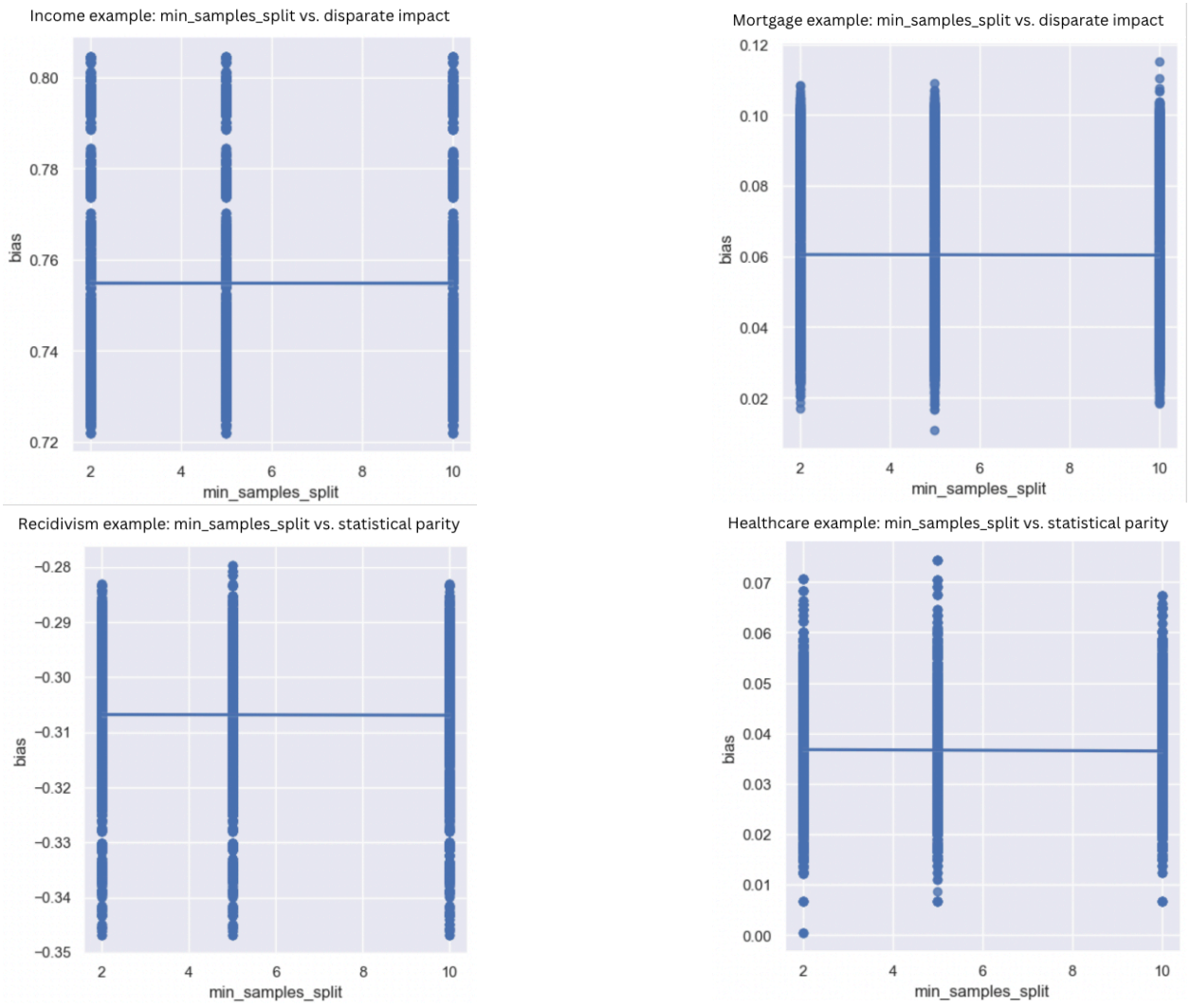

Figure 9: Examining impact of min_samples_split hyperparameter on exhibited bias

Similarly to max_depth, the min_samples_split parameter had no impact on bias exhibited within the constraints of our parameter testing grid. This parameter controls the minimum number of samples required to split an internal node during the tree-building process, and approaching smaller values tends to increase model complexity and the potential of overfitting. Due to the correlation we discovered between bias and accuracy, this result may suggest that within our specified testing values, the minimum number of samples did not have an impact on the overall performance of our models.

Overall, the findings from our hyperparameter plots are limited by the constraints of our chosen parameter grid, but they indicate that max_leaf_nodes is linearly related to both disparate impact and statistical parity, n_estimators is linearly related to statistical parity, and min_samples_leaf may have a relationship with both bias metrics as well. Expanding testing to larger values of all parameters, incorporating different bias metrics, and applying BAGS on a bigger variety of datasets allows us to draw more conclusive results about the relationship between Random Forest hyperparameters and bias.

6 Conclusion

Ultimately, the BiasAwareGridSearch algorithm provides developers with deeper insights into how their chosen hyperparameter tuning methods can have an impact on the bias exhibited by their models. Although BAGS alone did not have a significant and consistent impact in minimizing the bias exhibited by our models compared to naive GridSearch, the visual representations of the hyperparameters' influence on bias and figures depicting the relationship between bias and accuracy provide greater transparency during the model development process. This is crucial for developers and stakeholders to make well-informed decisions and ensure trust in these algorithms.

However, as noted by Aïvodji et al. (2019), 'fairwashing' refers to the practice of superficially portraying a machine learning model as fair without addressing biases present in the data or the model. Although our algorithm is a useful tool for examining bias on a general scale, bias is a complex issue that requires significant knowledge about the domain and careful consideration of the impacts of applying correction tools. Approaches for "correcting" bias that fail to examine the nuances in a particular situation may unintentionally warp or misconstrue results, which can potentially perpetuate and even exacerbate unfairness.

In our next steps, we plan to broaden the algorithm's scope by incorporating additional bias metric functions beyond disparate impact and statistical parity difference. Different bias metrics capture distinct aspects of fairness and discrimination within models, and including these allows for a more holistic approach to correcting bias during the mitigation process.

Secondly, we aim to apply BiasAwareGridSearch on a variety of classifiers, such as XGBoost and LightGBM, to further explore the correlation between bias and accuracy. By doing this, we are looking to explore deeper insights into model selection criteria for developers, helping them make more informed decisions when choosing classifiers for their applications. For example, when presented with a decision between two classifiers, awareness that a specific model has a stronger correlation between bias and accuracy could influence the selection in its favor, even if other bias mitigation measures are not applied.

Finally, we intend to expand our testing to include additional datasets from various domains. This broader testing will allow us to assess the algorithm's performance across different data types and further validate its effectiveness in addressing bias in machine learning models.

References

Alelyani, Salem. 2021. "Detection and Evaluation of Machine Learning Bias." *ArXiv* abs/2101.03014

Aïvodji, Ulrich, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. 2019. "Fairwashing: the risk of rationalization." [Link]

Chen, Zhenpeng, Jie M. Zhang, Federica Sarro, and Mark Harman. 2023. "A Comprehensive Empirical Study of Bias Mitigation Methods for Machine Learning Classifiers." *ACM Trans. Softw. Eng. Methodol.* 32(4). [Link]

Gebru, Timnit, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. "Fairness and Abstraction in Sociotechnical Systems." In *Proceedings of the Conference on Fairness, Accountability, and Transparency*.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer

Hildebrandt, Mireille. 2019. "The issue of bias: the framing powers of ML." *Philosophical Transactions of the Royal Society A* 377(2142), p. 20180153. [Link]

Larson, Jeff, Surya Mattu, Lauren Kirchner, and Julia Angwin. 2016. "How We Analyzed the COMPAS Recidivism Algorithm." *ProPublica*. [Link]

Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. "A Survey on Bias and Fairness in Machine Learning." *ACM Comput. Surv.* 54(6). [Link]

Pagano, Tiago P., Rafael B. Loureiro, Fernanda V. N. Lisboa, Rodrigo M. Peixoto, Guilherme A. S. Guimarães, Gustavo O. R. Cruz, Maira M. Araujo, Lucas L. Santos, Marco A. S. Cruz, Ewerton L. S. Oliveira, Ingrid Winkler, and Erick G. S. Nascimento. 2023. "Bias and Unfairness in Machine Learning Models: A Systematic Review on Datasets, Tools, Fairness Metrics, and Identification and Mitigation Methods." *Big Data and Cognitive Computing* 7(1). [Link]

Appendices

A.1 AIF360

AI Fairness 360 (AIF360) is an open-source toolkit developed by IBM Research that provides algorithms and metrics for measuring and mitigating bias in machine learning models. It includes a wide range of functionalities for bias mitigation across various stages of the machine learning pipeline, including comprehensive bias assessment metrics, data processing tools, post-processing debiasing methods, and fairness evaluation techniques.

A.2 Bias Metrics

A.2.1 Disparate Impact

Disparate impact is the ratio of favorable outcomes of the unprivileged group to the privileged group. It is calculated by the following equation, where Y is the model prediction, X is the protected attribute.

$$1 - DI = P(Y = 1, X = unprivileged)/P(Y = 1, X = privileged)$$

The resulting value indicates the degree of disparity. A value close to 0 implies fairness, a negative value indicates bias in favor of the unprivileged group, and a positive value indicates bias against the unprivileged group.

A.2.2 Statistical Parity Difference

Statistical parity difference is the difference of the rate of favorable outcomes from the unprivileged group to privileged group. It is calculated by the following equation, where Y is the model prediction, X is the protected attribute.

$$SPD = P(Y = 1, X = unprivileged) - P(Y = 1, X = privileged)$$

The resulting value indicates the degree of statistical parity difference. A value close to 0 implies fairness, a negative value indicates bias against the unprivileged group, and a positive value indicates bias in favor of the unprivileged group.

A.3 Random Forest Classifier Hyperparameters

Hyperparameter tuning is the process finding the optimal hyperparameters for a machine learning algorithm to improve its performance. The goal is to search through a specified parameter list and find the combination that results in the best model performance, usually measured using a metric such as accuracy, precision, or recall. The class GridSearchCV from sci-kit-learn is commonly used for hyperparameter tuning; it works by searching through all possible combinations of hyperparameters from a specified grid and using cross-validation to ensure that a model's performance is stable and not affected by any particular way the data is split. Below is a description of the five hyperparameters we chose to test using our novel GridSearch.

`max_leaf_nodes`: This parameter specifies the maximum number of leaf nodes in each decision tree of the Random Forest. Controlling the maximum number of leaf nodes can help prevent overfitting by limiting the complexity of individual trees.

`n_estimators`: This parameter determines the number of decision trees to be used in the Random Forest. Each tree contributes to the final prediction, and having more trees can improve the performance of the model, up to a certain point. Increasing the number of estimators typically leads to better generalization but also increases computational cost.

`max_depth`: This parameter restricts the maximum depth of each decision tree in the Random Forest. Limiting the depth of the trees can help control overfitting by preventing them from becoming too complex and capturing noise in the data.

`min_samples_split`: This parameter sets the minimum number of samples required to split an internal node during the construction of a decision tree. If the number of samples at a node is less than this value, the node will not be split, which can help prevent the tree from being overly sensitive to individual data points.

`min_samples_leaf`: This parameter specifies the minimum number of samples required to be at a leaf node. It ensures that each leaf node has a minimum number of samples, which can help prevent overfitting.