Bias Detection tool for ML Researchers

Solution Name: Bias Detection Tool
Team Name: ZIRI
Link to Video: **https://youtu.be/PbDLE9PCxls**

List of team members:

1. Dr. Vignesh Subbian
2. Nick Souligne
3. Shashank Yadav

Contact information for team's point of contact

- Dr. Vignesh Subbian, Associate Professor, Department of Biomedical Engineering, UA
- Email: vsubbian@arizona.edu
- Phone: 520-621-6559

Abstract:

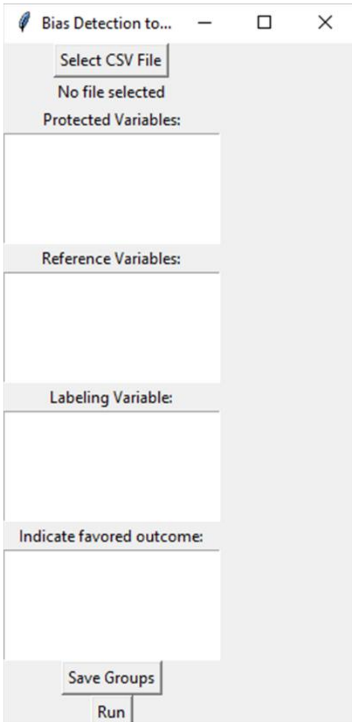
Although AI/ML algorithms offer promise for clinical decision making, that potential has yet to be fully realized in healthcare. Even well-designed AI/ML algorithms and models can become inaccurate or unreliable over time due to a variety of factors such as changes in data distribution, user behavior, or shifts in data capture among others. The NIH's NCATS challenge calls for solutions to help detect and eliminate latent predictive and social bias in these AI/ML models.

Our proposal is a novel tool designed for AI/ML researchers to test their datasets for inherent social biases. Additionally, our tool will then run their dataset through a wide variety of well known ML models to compute algorithmic level predictive biases and display them to the user in an easy to read and navigate GUI. These metrics will include at the data level Class Imbalance, Demographic Disparity, and Jensen Shannon Divergence. At the algorithmic level the tool will compute the statistical parity difference, disparate impact, equal opportunity difference, average odds difference, and Theil index. This solution is lightweight and easy to use for AI/ML researchers will any level of previous programming experience. Furthermore, this tool is easily portable to a wide range of different fields.

GitHub Code:

The repository for this code can be found at https://github.com/NickSouligne/Ziri.github.io

GUI at runtime:



Bias Detection to...

Select CSV File

No file selected

Protected Variables:

Reference Variables:

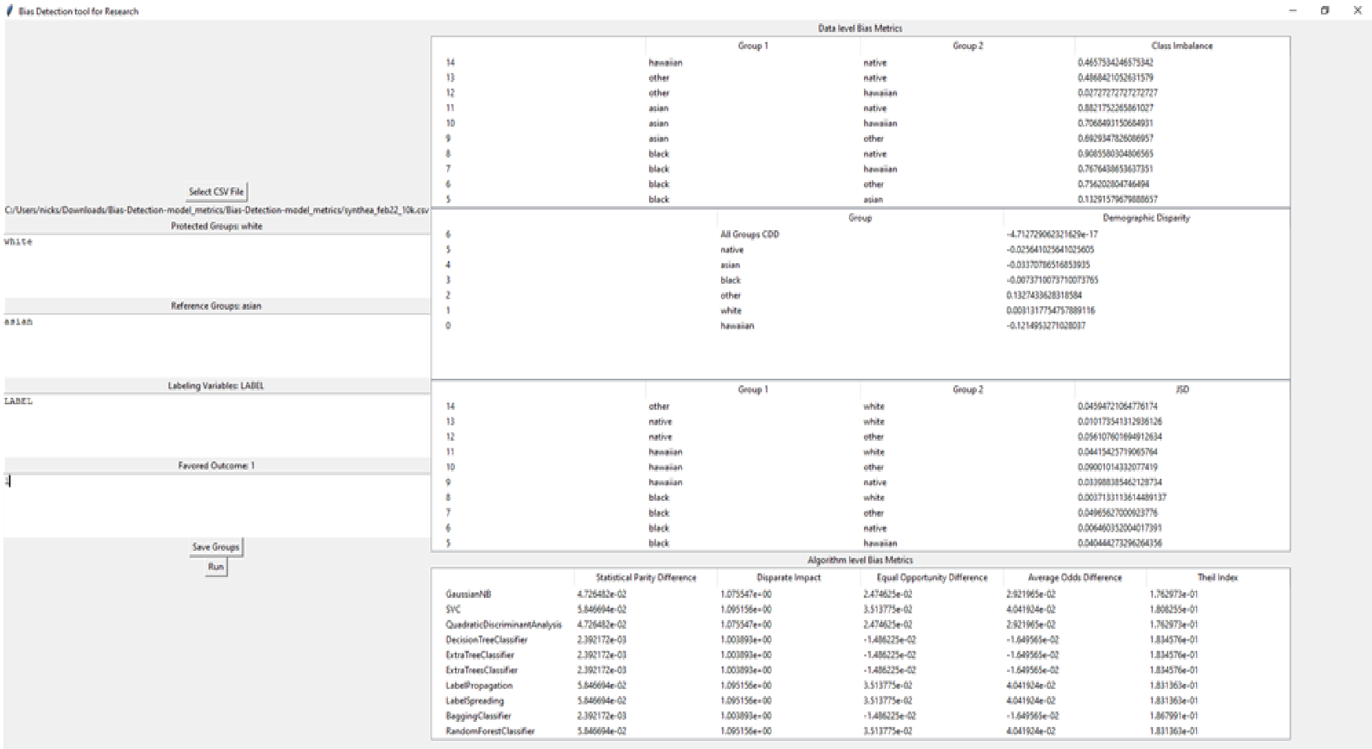
Labeling Variable:

Indicate favored outcome:

Save Groups

Run

GUI after running:



Bias Detection tool for Research

Select CSV File

C:/Users/nicks/Downloads/Bias-Detection-model_metrics/Bias-Detection-model_metrics/synthea_feb22_10k.csv

Protected Groups: white

white

Reference Groups: asian

asian

Labeling Variables: LABEL

LABEL

Favored Outcome: 1

1

Save Groups

Run

Data level Bias Metrics

	Group 1	Group 2	Class Imbalance
14	hawaiian	native	0.4657534246575342
13	other	native	0.4868421052631579
12	other	hawaiian	0.02727272727272727
11	asian	native	0.8821752265861027
10	asian	hawaiian	0.7068493150684931
9	asian	other	0.6929347826086957
8	black	native	0.9085580304806565
7	black	hawaiian	0.7676438653637351
6	black	other	0.756202804746494
5	black	asian	0.13291579679888657

	Group	Demographic Disparity
6	All Groups CDD	-4.712729062321629e-17
5	native	-0.025641025641025605
4	asian	-0.03370786516853935
3	black	-0.007371007371007365
2	other	0.1327433628318584
1	white	0.0031317754757889116
0	hawaiian	-0.12149532710280037

	Group 1	Group 2	JSD
14	other	white	0.04594721064776174
13	native	white	0.010173541312936126
12	native	other	0.056107601694912634
11	hawaiian	white	0.04415425719065764
10	hawaiian	other	0.09001014332077419
9	hawaiian	native	0.033988385462128734
8	black	white	0.0037133113614489137
7	black	other	0.04965627000923776
6	black	native	0.006460352004017391
5	black	hawaiian	0.040444273296264356

Algorithm level Bias Metrics

	Statistical Parity Difference	Disparate Impact	Equal Opportunity Difference	Average Odds Difference	Theil Index
GaussianNB	4.726482e-02	1.075547e+00	2.474625e-02	2.921965e-02	1.762973e-01
SVC	5.846694e-02	1.095156e+00	3.513775e-02	4.041924e-02	1.808255e-01
QuadraticDiscriminantAnalysis	4.726482e-02	1.075547e+00	2.474625e-02	2.921965e-02	1.762973e-01
DecisionTreeClassifier	2.392172e-03	1.003893e+00	-1.486225e-02	-1.649565e-02	1.834576e-01
ExtraTreeClassifier	2.392172e-03	1.003893e+00	-1.486225e-02	-1.649565e-02	1.834576e-01
ExtraTreesClassifier	2.392172e-03	1.003893e+00	-1.486225e-02	-1.649565e-02	1.834576e-01
LabelPropagation	5.846694e-02	1.095156e+00	3.513775e-02	4.041924e-02	1.831363e-01
LabelSpreading	5.846694e-02	1.095156e+00	3.513775e-02	4.041924e-02	1.831363e-01
BaggingClassifier	2.392172e-03	1.003893e+00	-1.486225e-02	-1.649565e-02	1.867991e-01
RandomForestClassifier	5.846694e-02	1.095156e+00	3.513775e-02	4.041924e-02	1.831363e-01

Methodology

Our teams' approach to measure disparity and bias was slightly different than the stated guidelines for the challenge. After a thorough review of the current literature on ML, in particular bias detection and mitigation in ML, and multiple group brainstorming sessions we decided to pivot from a developing a model or algorithm to mitigate bias. What we had learned from our literature review was that there was a dearth of options for researchers to test their datasets for inherent bias and even fewer options to help these researchers find the best model for their datasets. To match this unmet need our team developed a novel Bias Detection tool that allows the user to select their dataset in CSV format, indicate the protected and reference variables, and provide the column pertaining to the labels and what label would be considered favorable.

Upon the user providing the requested inputs in the graphical user interface (GUI), our tool then runs the dataset through our algorithms to calculate the Class Imbalance, Demographic Disparity, and Jensen Shannon Divergence to provide the researcher with an understanding of the data level biases. In particular, we chose these metrics to cast a wide net in terms of what the metrics indicate to the researcher. Class Imbalance indicates whether certain groupings have a larger representation in the dataset. Demographic Disparity indicates whether the proportions of rejected to accepted outcomes are consistent across groupings. Jensen Shannon Divergence indicates the level of similarity between distributions or groupings. When utilizing all three metrics a researcher can get a more complete picture of any potential biases when compared with just using 1 of these metrics. Additionally, this approach has the benefit of measuring prospective bias in the data before running it through any ML models further improving the ability for researchers to recognize bias in their data before spending time and resources training a model on biased data.

At the algorithmic level we looked to identify biases that persist in models that have been trained on the previously input dataset. Utilizing the lazypredict and aif360 software packages, our tool tests all of the models available in sklearn package and outputs our bias detection metrics computed for each of these models. This allows the user to see a comprehensive list of potential options for their model and can decide as to which model would best serve their needs while reducing the bias inherent in the model. The bias metrics we chose for this were statistical parity difference, disparate impact, equal opportunity difference, average odds difference, and Theil index. As with the previously mentioned data level metrics, we aimed to cast a broad net with regards to the information the metrics are conveying. These metrics were implemented via the aif360 package and ran on a number of models generated from the Lazypredict package and sklearn. Importantly, this tool does not save any data from run to run meaning that each time a dataset is added in, and metrics are computed, the models are re-instantiated with no prior learning or exposure to the dataset. This allows the tool to eliminate any potential latent bias that might be introduced over time due to the inherent adaptive learning properties of ML models.

Value Proposition

This tool is unique in that it provides an intuitive GUI for researchers to quantify any bias quickly and easily in their data, as well as provides them with a list of model options and their ability to overcome bias in the data. As the number of researchers utilizing ML continues to grow, it is important that tools are available for all skill levels of researchers. As noticed during the literature review stage, many of the most widely used packages for bias metric detection or mitigation require at least some level of programming or software engineering expertise. While our tool may not be the best option for all use cases, it does fill a niche for researchers without that prior expertise. Additionally, we believe this tool can help inspire trust and confidence in ML development as the tool is easily accessible and works to ensure biases are reduced before they could impact the model in a healthcare setting.

In terms of the biases that we aimed to address, we primarily looked at data level social fairness biases with a secondary goal of providing the researchers with a number of potential ML model options and any potential predictive fairness biases for each model. Allowing the researchers to see how different models handle the potential biases would help to improve the development of these models by reducing the time and resources spent determining and testing what models are appropriate for the specific researchers' goals. Furthermore, with that in mind we are confident that our tool is highly accurate. In our test cases, for data level bias detection, the model showed accuracy consistent with examples found in documentation of Amazon Sagemaker and aif360. The algorithmic level bias metrics accuracy was determined to be of a high level as well as it is a direct implementation of the aif360 package, which is held to be a highly accurate package.

Healthcare Scenario

Our tool works to eliminate bias in ML models before the ML model is ever introduced into a healthcare setting. We believe that this approach will help to foster trust between ML researchers and the healthcare community by removing biased datasets and models before the researchers ever look to implement in healthcare settings where biased models may sour the clinician to ML based approaches in the future. This approach also has the benefit of being adaptable to whatever healthcare setting the researcher may be targeting as the metrics are diverse and the tool can accept any CSV dataset, provided the user accurately inputs their variables. As developed, our tool requires the user to input a new dataset every time they run the program thus reducing any chance of latent bias from adaptive learning in the model. Furthermore, as the program re-instantiates the ML models with each run there is no prior exposure to the dataset. This is an important functional requirement because it allows the researcher to apply the tool to a variety of potential healthcare scenarios without worrying about inadvertently introducing bias by testing different datasets.

Currently, our tool does not offer suggestions on how to handle inherent biases other than offering the algorithmic level bias metrics for a slew of well known ML models. Due to time constraints this feature was removed, but for a post-prototype release our team would ensure that the tool can indeed offer basic advice or explanations of the bias metrics alongside the output results. With the bias metrics in hand, it would then be up to the individual researcher or team of researchers to more closely examine either their dataset or their model based on which metrics show the highest levels of bias. In providing this tool with these metrics we hope that researchers can better optimize their ML models leading to a more fair and better representative model that will in turn lead to better patient outcomes.

Operational Requirements

This tool can be deployed in most computing environments due to the low computational cost. Despite running a bevy of ML models through the bias detection algorithms, the tool does not utilize a graphical processing unit (GPU). While this can lead to a longer run time in certain environments, the lack of GPU support allows for the tool to be more widely distributed and used. The tool does require a fair number of python package dependencies installed into the Python environment. These dependencies include pandas (1.5.3), scipy (1.10.1), sklearn (1.2.1), lazypredict (0.2.12), aif360 (0.5.0), importlib (5.12.0), and tqdm (4.64.1). Testing of this tool was performed in Windows 10 using Python 3.9.16. Additionally, the program utilized about 200 MB of RAM with a brief peak of 1.5 GBs during our testing involving a dataset containing 5 columns and 10,000 records. Despite the number of dependencies that this tool relies upon, none of the packages are considered proprietary information. A focus for our team was to rely on as few dependencies as possible, and when necessary that the dependencies be open source packages as we fully anticipate providing the tool as open source software under the BSD 3 license.

Architectural design was an important consideration for our team. Our goal was to provide a tool that would be easily adaptable to different scenarios or environments and that influenced our choices in packages and codebase design. First, we wanted to encapsulate our functions inside a class to allow for the tool to be represented as an object that can be implemented in other projects. We then encapsulated each of the data level metrics inside a function to allow for access to the individual metrics without having to compute all of them at once. Importantly, we chose to use Tkinter for our GUI design as the Tkinter package is a part of the base Python installation helping to reduce the dependencies the project relies on and further ensure long term support.

Sustainability Plan

Our team envisions this tool being sustained outside of a healthcare organization. While this tool could be used in a healthcare setting, particularly to help with identifying potential bias in EHR systems, it is primarily envisioned to work with ML researchers before the model is deployed in healthcare organizations. This approach puts the burden of sustainability on the researcher and the future development of the tool as well as drastically reduces the cost of upkeep due to the open source nature of the tool. While this approach does have its limitations in detecting and mitigating bias on a continual effort with new ML models, the benefits of tackling the bias problem before deciding on the model should help reduce the possibility of data drift impacting results. This allows for our tool to accurately detect and calculate retrospective metrics for a wide range of ML models, however due to time constraints we were unable to implement functionality for prospective metrics.

We recognize that to further optimize this tool to reduce bias a wide range of perspectives and opinions would need to be considered. Thankfully, our team consists of members with a wide range of backgrounds including Computer Science, Systems and Industrial Engineering, and Biomedical Engineering. Members of the team also had recent experience in working alongside a clinical staff to develop tools for use in the clinical setting that helped inform our functional requirements as well as gave valuable insights as to the types of data the models would be likely to receive. Over the course of the challenge our team reached out to several physicians to obtain their perspectives but were unable to align schedules for interviews, thus we relied on our previous knowledge and a wide range of literature with a focus on ML in clinical settings.

Generalizability Plan

We believe that our tool has the potential for high impact on the development of new ML models. One of the main benefits of our tool is how easily accessible it is, even for users with very little prior programming expertise. As ML and AI research communities continue to grow it is important that we provide tools for all levels of experience, and that is something our tool excels at. While many packages out there currently can perform very similar functions as ours, very few offer as many metrics at both the data and the algorithmic level. Even fewer of these packages offer an intuitive GUI, which we believe is an important component in accessibility for researchers in the field. Additionally, many of the packages currently available require the user to have at least a moderate level of prior software engineering experience to get the most from the package. In targeting the users who may lack the required expertise we can bring these tools to a greater number of researchers who can in turn lead to a larger number of fair and debiased models utilized in healthcare settings.

Another focus for our group was on making the output clear, and easily read. Utilizing the Tkinter package and pandas' data frames we formatted the outputs in a way that reads similarly to a CSV or Excel file. We also took care not to put too much of a burden on the user in terms of data cleaning or pre-processing. By implementing functions from lazypredict and aif360 packages we were able to handle the majority of the data pre-processing without needing the supervision of the user. This further underscores the ability for the tool to handle data from many different settings as the robustness of these packages allows for handling of many different types of data. While our testing has mainly focused on EHR system style datasets, it could feasibly be used for almost any other clinical discipline.

As mentioned previously, this tool is also lightweight in terms of computational complexity and resources. This characteristic of the program allows it to be ran in a wide range of system environments. Currently, this tool has been successfully implemented in Ubuntu Linux, Windows 10, and Windows 7 OS environments. While it has been implemented and used in its current state, there were several functional requirements that were originally envisioned for the tool that were cut due to time constraints. Mainly, we wished to add the ability for the user to select their own personally developed ML model and calculate any inherent latent bias. We also wished to add a slider to the GUI that would allow the user to indicate what portion of the dataset to be used as the training and testing sets. Furthermore, we wished to provide a qualitative assessment of the bias metrics alongside the quantitative metrics to better inform the user as to the meaning of the metrics and how they might indicate bias in the data or model. Finally, while the tool does currently lack these features to suggest follow-up investigations, we believe implementation of such features would be entirely feasible given more time to work on the project.

Implementation Requirements

We believe that the system and human resources required to implement our tool are quite minimal. The system resources required are very low, especially when compared against other packages or tools that may utilize GPU processing. Human resources required are also low as our tool is focused on the researcher or team of researchers developing the model instead of focusing on a clinical implementation of the model. While implementation requirements can drastically differ from environment to environment, we believe that with the combined experience on our team that we developed a tool that should allow for relatively seamless integration into different research settings.

When measuring the success of implementing our tool we mainly look to quantify the reach of the program and how many successfully implemented bias free models were influenced by our tool. The reach of the program can be measured by the amount of traffic to the GitHub repository and the number of forks or implementations of it. Measuring the success of the implementations is a bit trickier and relies more on communication between the development team and the researcher implementing the model with input from the clinicians or patients the model serves. By positioning our tool as an open source software, we hope to drive community engagement with the tool leading to improvements and open communication between users and developers. We hope to utilize this communication to better quantify the success of our tool and to help inform further development.

Lessons Learned

One of the main challenges that our team encountered was the lack of access to healthcare professionals due to scheduling conflicts. While the previous experience in similar domains greatly helped inform design decisions, the experiences were not explicitly designed around ML models and their implementation in clinical settings. Despite this obstacle, we still made a strong effort to take into account literature from relevant settings. We believe that these efforts do still culminate in a prototype tool that will hopefully provide the same level of diagnostic support for ML researchers in any field over any number of years. It is important to note that the main issue in future support of the tool lies mostly in the ongoing support and development of Python and the individual dependencies.

Overall, this challenge was a good experience in thinking through the issues faced by ML model developers and how bias can impact the clinical decision-making. Our team was able to get experience with a wide range of bias detection metrics and the publicly available tools to assist with mitigating these biases. One suggestion for future challenges is to broaden the submission requirements in terms of languages. While Python is typically the gold standard for ML model development, there is increasingly greater levels of support for ML in other languages such as C, Java, or R.