An Assessment of Publicly Available Tools for Identifying Dataset Biases

Victoria Golden
The Volgenau School of Engineering
Applied Information Technology, MS

Fairfax, Virginia
vgolden@gmu.edu

Austin Crow
The Volgenau School of Engineering
Applied Information Technology, MS

Fairfax, Virginia
acrow3@gmu.edu

Patcharaporn Adchariyavivit
The Volgenau School of Engineering
Data Analytics Engineering, MS

Fairfax, Virginia
padchari@gmu.edu

Abstract*—Machine learning is used in many fields and for various types of problems. Since raw data is often incredibly large and can be full of problems, an important concern is making sure these large data sets are as useful as possible. Bias is commonly found in datasets and can go completely unnoticed, causing major problems once this data begins to get used. These biases cause incorrect results often leading to disadvantages to one or more test groups. This problem of bias has led to the development of identification and mitigation tools; there are several public bias detection tools including Linux Foundation AI Fairness 360, Google TensorFlow Fairness Indicators, Pymetrics Audit-AI and DatascienceLab.com Skater available today. This paper provides an assessment of the four listed bias detection tools to determine the best publicly available tool for bias detection and mitigation. Each tool was tested independently before grading them based on criteria. The authors developed criteria relevant to assessing tools of potentially varying development levels and assigning an importance weight to each criterion. The scores from each independent assessment were then compared via the Analytic hierarchy Process to determine which tool was best suited based on the author's criteria to recommend to the public. The results of this study provide the best bias detection and mitigation tool based on the author's criteria and rankings. Further study would include testing on a custom dataset, increasing the number of different datasets for each detection tool to identify each tool characteristic for the suitable dataset.

Keywords— analytical hierarchy process, bias, bias detection tool, machine learning

I. INTRODUCTION

There is no doubt about the usefulness of machine learning in today's data environment. Machine learning algorithms are used across many domains and across a variety of problems. These algorithms are seen in e-commerce such as Amazon's recommendations [1], financing through loan applications, image processing, autonomous vehicles, and speech recognition among many others. Acknowledging the market penetration of machine learning and its relevance to many big data related challenges, it is important to address the effect of bias.

Bias in the context of machine learning refers to "any basis for choosing one generalization over another, other than strict consistency with the instances" [2]. Bias comes from a variety of sources including humans, machines, and nature. Due to it being present in virtually every dataset, machine learning methodologies base their results on a foundation which can lead to possible unfairness in the affected groups. According to Mehrabi et al., fairness is "the absence of any prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics" [3]. Due to the omnipresence of machine learning today, developers should strive to identify the underlying biases that exist in their data in order to ensure no entity is treated unfairly.

Machine learning bias is not something that is as complicated as it may sound, but rather it is an unconscious or conscious skew in data. A great description of bias is "… omissions and deliberate choices of inclusion may show a particular bias" [4]. There are many different types of bias, not all of which are bad. A rule of thumb is to always clean the data, and if something appears to be an outlier, it should be reviewed. The most common examples of data bias are sample, exclusion, measurement, recall, observer, racial, and association bias [5]. If one is cleaning data that one is not familiar with, challenges may arise because it is hard to determine whether the data is skewed in an unintentional way. When cleaning data, it is best to exercise caution as the data may be valuable—without appearing so—during this phase of the data analytics lifecycle.

Bias affects all types of data sets from any time frame and is often harmful when coming up with an

algorithm for machine learning. The machine learning algorithm can only work off the data that is used to teach it. If this skewed data is left as is, it will perpetuate the bias into further generations of data. This perpetual bias can create unfair work environments, whether it be based on gender, ethnicity or any other factor that should not have an impact on the work that is being performed. An example by Stolzfus indicating the effects of underlying bias is, “If [it] perceives that men hold the vast majority of executive jobs, and the machine learning process involves filtering through the raw data set and returning corresponding results, it's going to return results that show a male bias.” [4]. There is no proven reason why men should hold an executive position over women, so a machine learning process like this will only serve to hurt the company using it or show biased data that pushes false reasonings.

Bias is the tendency of statistics which overestimate or underestimate parameters. The simplest way to understand the meaning of bias is the error or distortion which causes inaccuracy in statistics analysis. Bias can leak into analysis results from many root-causes, but the most common root-causes of bias are from sampling errors or a sample that is not representative of an entire population. There are many types of bias. This study is related to data analytics so the types of bias which are explained are the bias types that can most affect data analytics, including

- Selection bias
- Self-selection bias
- Recall bias
- Observer bias
- Survivorship bias
- Omitted variable bias

These biases can negatively influence the analysis results, so the researcher must identify the type of bias and take steps to get rid of bias in their statistical result. Each type of bias is explained in further detail in the following sections.

- Selection bias: It occurs when the researchers select or pick a set of data without distribution. The selected data or representative sample should represent the whole data or population.
- Self-selection bias: This bias is a subset of selection bias but the self-selection bias specifically refers to when the samples are limited by something; another meaning is data users select some data by themselves so the samples won’t cover the entire dataset or population
- Recall bias: Most of this bias is from surveys and interviews because the respondents provide the misinformation unintendedly such as bad memories or misunderstanding.
- Observer bias: This bias relates to the researcher's emotions and sensitivity. The researchers pick or select only the statistical sample or dataset from their result expectation.
- Survivorship bias: Some parts of the data set are ignored by researchers after the pre-selection process. This kind of bias made the dataset miss some points like it keeps only survivors.
- Omitted variable bias: This bias occurs when models leave some important parameters or dataset out.
- Cause-effect bias: It is not a classic bias but this bias should be considered especially for decision-makers.[6]

There are few tools, algorithms, and mitigation approaches that exist in the bias identification realm of machine learning.

II. PROBLEM STATEMENT

With machine learning becoming more popular due to challenges associated with big data, it is incumbent upon the developer to identify any underlying biases present in the dataset. Tools to identify and help mitigate bias are not readily studied nor widely used. These tools have not been incorporated into the data analytics lifecycle. More research must be conducted to help improve fairness to all parties affected by machine learning algorithm results.

III. LITERATURE REVIEW

The article *Ethical AI: its implications for Enterprise AI Use-cases and Governance* talks about how bias is a major problem when it comes to AI and machine learning. It references that while bias is often unintentional, it is nonetheless observed very frequently, causing major problems for training sets. An example of this is “Google Photo labeling pictures of a black Haitian-American programmer as ‘gorilla’”[7]. This is something that is scientifically and morally unacceptable, so the authors are working to find a way to mitigate these problems. With a focus pertaining to training data, specifically historical, representation, and measurement & aggregation bias, the article mentions how training data needs to make sure that it is representative of the entire population being studied. The authors mention TensorFlow Fairness Indicators as being a great tool in order to combat these biases. A major point worth mentioning is that bias mitigation should not be a one-time effort, but something to strive for on a continuous basis. A key issue raised is perpetuating bias through the training data set, which is addressed via the use of Google TensorFlow.

AI Fairness 360 has its methodology and thought process for its usage described in the article *AI Fairness*

360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. The others go into detail about the different reasons to use AI Fairness 360 as well as mention, "[The] architectural design and abstractions enable researchers and developers to extend the toolkit with their new algorithms and improvements, and to use it for performance benchmarking." [8][2].

An example of AI Fairness 360 overviews a study of Fairness Assessment for Artificial Intelligence in the Financial Industry [9] is a fairness evaluation and review of imbalanced data treatment and bias mitigation. The study uses AI Fairness 360 because of its comprehensiveness and usability [10]. The dataset is a default credit card with more than 30,000 credits which contains the amount of the given credit, gender, education, marital status, age, history of past payment, amount of bill statement, and amount of previous payment. There are 5 cases of using different methods to balance treatment and mitigate bias, including Plain LightGBM algorithm, Synthetic Balanced data, Bias Mitigated (by AI Fairness 360), Synthesis Balanced and Bias mitigated data, and Manipulated biased data. Based on focusing on bias mitigation by AI Fairness 360, bias was mitigated in pre-processing using a reweighting method. AI Fairness 360 applies to change weight to training data. Table 1 shows that the difference between groups is eliminated by AI Fairness 360. The following case is the combination between balance data and bias mitigation case. The study used the SMOTE method for balancing data and AI Fairness 360 for bias mitigation. They were comparing every case result together. It could conclude that the best performance case is a combination of balance and bias mitigation methods because of the less false-negative rate. However, from the result as a Table 2, the performance is still not satisfied if it has only bias mitigation. On the other hand, this study shows balancing data has more effect than bias mitigation.

Table 1. Fairness matric before and after mitigation by AI fairness 360

	Before	After
Diff. Statistical Parity	0.0345	0.000
Disparate impact	1.0457	1.000

Table 2. Performance metrics for all cases

	Performance	
	Accuracy	False Negative Rate
Case 1	0.82	0.62
Case 2	0.81	0.23
Case 3	0.82	0.64
Case 4	0.83	0.19

The author declares the term fair to mean "without disparate treatment and disparate impact," with disparate treatment being an intentional act of discrimination and disparate impact being unintentionally giving disproportionate advantage to one group. The chapter mentions the Civil Rights Act of 1964 as an example of disparate impact. "Carolina's Duke Power company adopted the requirement that employees in all departments (except its lowest-paying labor department) have a high school diploma and a minimum score on two paper and pencil tests, the Bennett Mechanical Comprehension Test and the Wonderlic Cognitive Ability Test"[11]. This had a disparate impact because many of the black employees at the time did not have the resources needed to take those tests thus not qualifying for the positions.

According to one paper, Fairness is complicated and has a multi-faceted nature. There are more than 21 mathematical definitions [12], and one definition will give totally different results from another. Developers created AI Fairness 360 for detecting, understanding, and mitigation bias. AI fairness 360 is also the first system that combines bias metric, mitigation algorithms, metric explanations, and industrial usability all together within one tool.

In a tool such as AI Fairness 360, Algorithm categories depend on the location as in Figure 1 where it can interfere into the process such as pre-processing and will be used if the algorithm allows modification of the training data. If the algorithm allows for modifying the learning procedure, then in-processing can be used. If the algorithm can act like a black box (no need for data modification) then the post-processing can be used.

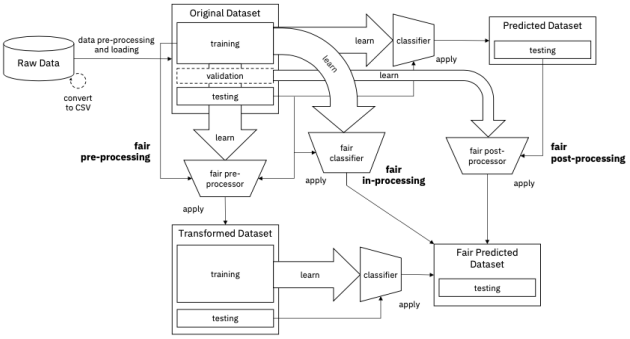

Fig 1. The Fairness Pipeline

Throughout the machine learning lifecycle, algorithms can be classified into 3 main stages, pre-processing, in-processing, and post-processing. According to metrics published regarding AI Fairness 360, there are 9 bias mitigation algorithms in the three main stages for AI Fairness 360 as in shown in Figure 2. All algorithms are transferred from transfer class [13].

- Pre-processing: Reweighting, Optimized, Learning fair representation and disparate-impact remover.
- In-processing: Adversarial debiasing, Prejudice remover and Meta fair classifier

- Post-processing: Equalized Odds, Calibrated Equalized Odds and Reject option classification.

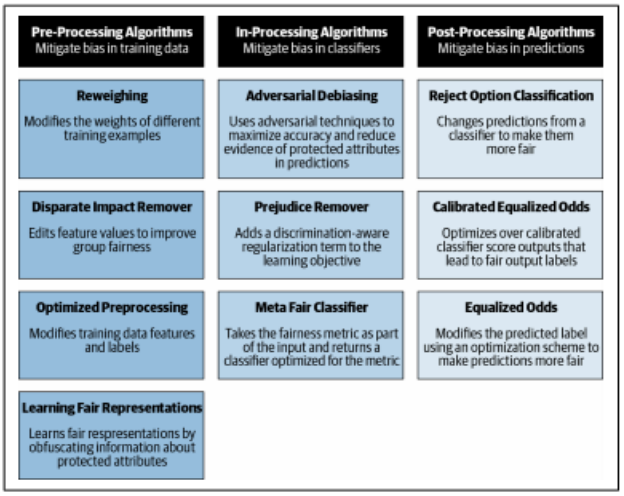

Fig 2. Algorithms in machine learning pipeline of AI Fairness 360

Like the AI Fairness 360 tool, another tool is Audit-AI. This tool audits as one of assessment approaches which can use algorithms or be judged by humans. However, to reduce bias or human error and sensitivity, Pymetrics created a Machine Learning tool for matching job seekers with their suitable job in the organization without being interfered with by human sensitivity.

According to documentation published by Pymetrics, the main core data for Pymetrics' candidate screening service are from psychological studies that are provided on the internet and mobile application in many languages. The purpose of the games is observing how the candidate decides to play or approaches to win the mission, it is not important if the candidate wins or not but rather the approach they take. The candidate needs to play all 12 games. Pymetrics collects more than 2 million users all around the world and has a variety of data in many domains.

Another large problem that was indicated is the use of AI to try and "level the playing field", as using historical training sets can end up putting past biases into code. Pymetrics is mentioned as being a company whose technology aligns with this standard of fairness in technology. Pymetrics essentially takes behavioral science tests and turns it into a game, then proceeds to take the data points gathered to build a success profile for their client to use: "Each time Pymetrics builds a custom success profile based on the performance of locally successful incumbents, the algorithm behind the profile is proactively audited for disparate impact before it is deployed for candidate selection" [11]. This is done by having a test set of individuals who are demographically diverse but are all successful. If part of the algorithm is determined to be disparate, then that data point is de-weighted (made irrelevant to the outcome). Due to the constant back testing, if an algorithm has room for improvement, it gets rebuilt in hopes of diminishing any sort of discrimination.

Bias is introduced into a dataset through a variety of means. Examples include population bias where only survey recipients who responded offer input, instrument-dependent bias [14] where the sampling instrument measures data in a way that alters the input received, among many other types [8] of bias. Bias has many definitions often suited to the project, domain, or objective. One definition of machine learning bias can be defined as "any basis for choosing one generalization over another, other than strict consistency with the instances" [2]. Bias in the context of statistics can be defined as "a model or statistic is unrepresentative of the population" [15]. In either case, bias can drive a machine learning algorithm's results in favor of one subgroup over another, causing unfairness. According to Caliskan et al., "machine learning can acquire stereotyped biases from textual data" [16]. Another example from Datta et al. describes how personalized ads from Google showed discrimination by suggesting ads that promised large salaries more frequently for males as compared to females "simulated male ads from a certain career coaching agency that promised large salaries more frequently than the simulated females…" [17]. Another example by Thelwall shows that a core component of some algorithms is the ability to deduce the meaning of words by associating them with other words that tend to occur within the same document. Using this approach can lead to conservative implications, such as that homemaker is part of the "meaning" of the word "woman," and that programmer is part of the meaning of the term "man" [18].

Bias in big data shows that people often make conscious or unconscious decisions that can have major effects on people's lives. Big Data is often used when creating machine learning algorithms and basing an algorithm on biased data will lead to the same biases down the line. Some of the mitigation strategies are to be diligent when cleaning the data to make sure that there are not obvious biases being shown. Another strategy involves diversifying the data, making sure it comes from different sources, as well as having a robust amount of data sets. As with all data, including big data, bias is always a concern that developers and analysts need to be cognizant of. Nowadays, many developers try to integrate bias mitigation steps by developing private tools that generally have the same goal. There are 3 different steps during the process to incorporate these methods, which include pre-processing, in-processing, and post-processing. The pre-processing starts at the beginning of searching through the data. If a dataset is found to have bias, steps to mitigate can be addressed at this point. Some data sets may contain unwanted biases, an example of which could be selection bias where "it is usually associated

with research where the selection of participants isn't random" [19]. Collecting multiple sources of data is an easy way to prevent selection bias using diverse samples to represent the population. The pre-processing method allows developers to catch unbalanced and unfair data before entering the in-process stage. For in-process, meta-algorithms (machine learning algorithms that learn from other machine learning algorithms) collect fairness metrics as input before returning new classifiers that are optimized in favor of the fairness metric. The last category is post-processing. Because the data was trained already, adjustments based on bias will be trained classifiers. This method spends less time than others because it uses trained data so there is no need to look back to the original dataset. However, for this method the accuracy needs to be validated [20].

IV. METHODOLOGY

The methodology for this assessment began with the identification and selection of publicly available tools advertised by their ability to identify biases within a dataset. Each tool identified is then compared to the other available tools to understand their capabilities. Tools being used for comparison in this project include:

- Linux Foundation: AI Fairness 360
- Google TensorFlow Fairness Indicators
- Pymetrics Audit-AI
- Datascience.com Labs Skater

After understanding how each tool functions, including similarities and dissimilarities, the authors utilized the analytic hierarchy process to score and rank the performance of each tool to ultimately identify the best available option for general use.

The tools then were evaluated utilizing the previously identified ranking mechanism. The authors then selected the "best" tool according to their ranking system, compared and analyzed the implications of utilization by each tool, and recommend the results to the reader. By understanding the capabilities of each tool, the authors recommended a relevant publicly available tool to help improve machine learning projects through the identification of biases in data. The following sections outline the tools assessed in this paper.

A. AI Fairness 360

AI Fairness 360 (AIF360) is an open-source tool that focuses on mitigation. Unfortunately, there are few tutorials showing how the tool works and can be implemented. AIF360 consists of four kinds of classes as outlined in the following list:

- Database Class: Handles all forms of data, entails the training dataset. The dataset can then be broken down further into sub datasets where more attributes can be added or modified.
- Metric Class: These are the classes that perform the group fairness checks to find bias in datasets and models. [12]
- Explainer Class: This class is used with the metric classes to give more detail into the biases found, often in the form of visualizations such as graphs
- Algorithm Class: This class holds the algorithms for bias detection in pre, in, and post-processing.

B. Google: TensorFlow Fairness Indicators

Fairness Indicators is an open-source tool that excels in computational graph visualizations as well as debugging. It is frequently updated with new features. Unfortunately, it is primarily Linux based, with very limited support for Microsoft Windows. Additionally, Fairness Indicators does not have any benchmark tests. A Typical Neural Network Model for TensorFlow is created by collecting a dataset, building models, training network, evaluating, and predicting [22].

C. Pymetrics: Audit-AI

Pymetrics Audit-AI is very easy to install as you simply "pip install audit-AI". There is a focus on the practical and statistical bias. Audit-AI primarily focuses on bias identification rather than both identification and mitigation. "Audit-ai determines whether groups are different according to a standard of statistical significance or practical significance" as well as offering …" tools to check for differences over time or across different regions, using the Cochran-Mantel-Hanzel test" [23].

D. Datascience.com Labs: Skater

Skater is an open-source library that makes the black box model easier to understand. It is actively under development and as such is weak in many areas, exemplified in its lack of support for building interpretable models. "Skater was developed as a Python framework to be a first step for enabling interpretability. Sometimes when the output is strange or unfamiliar, the assumptions is the model is biased but because misunderstood of black box model, so it is difficult to detect".[24]

E. Ranking Mechanisms Method and Evaluation Criteria

The authors decided to use the Analytic Hierarchy Process (AHP) as the means for scoring each tool against each other. This process was developed by Dr. Thomas L. Saaty in the 1970s. The process "allows the decision makers

to visually structure a complex problem in the form of a hierarchy having at least two levels: objectives (criteria for evaluation) and activities (productions, courses of action, etc.).” [25]. The AHP excels in decision making processes when it comes to ranking or priority setting for projects [26]. The authors chose to evaluate the tools based on the following criteria and sub criteria outlined in Figure 3 and the following sections. Each tool will be rated on a scale of 1-10 independently for each sub criterion.

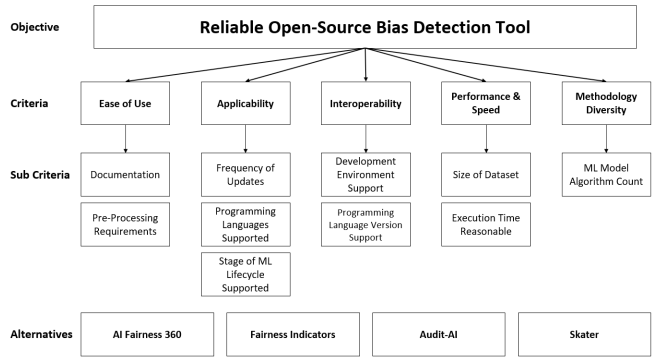

Fig 3. Analytic Hierarchy Criterion Diagram

1) Ease of Use: The applicability of this objective refers to scenarios such as availability of documentation, pre-processing requirements, etc. Tools will be scored based on presence or absence and extensiveness.

2) Applicability: Applicability refers to scenarios such as relevance to machine learning algorithms, frequency of updates, language support diversity, and use throughout a project. Relevance to machine learning algorithms refers to the support a tool provides across issues being tackled by any given machine learning algorithm. Use throughout a project refers to a tool’s ability to support identifying bias in either pre-processing, in-processing, or post-processing stages.

3) Interoperability Across Compute Platforms: The applicability of this objective refers to a tool’s ability to work across the many development environments available in today’s IT landscape, support requirements within a language, etc.

4) Execution Performance/Speed: Each tool will be evaluated on its ability to support varying sizes of datasets as well as its speed to process such varying sizes of datasets

5) Methodology Diversity: Tools will be evaluated on their ability to support multiple machine learning methodologies.

V. RESULTS

Each tool was rated on a scale from one to ten for each outlined sub criterion. This score was then weighed against the overall weight of each criteria and input into an AHP template to produce the overall recommended tool. The following sections outline each tool and the methodology going into each tool’s score per criterion.

A. AI Fairness 360

For the purposes of this evaluation, the authors will be discussing the core AI Fairness 360 modules. It should be noted that AIF360 has a scikit-learn compatible API reference, however, it will not be discussed in this paper.

1) Documentation: The documentation provided by AI Fairness 360 is fairly comprehensive. It covers topics that include algorithms, datasets, explainers, and fairness metrics. This sub criterion received a score of 8 due to the documentation lacking in-depth discussion around using the tool with a custom dataset. Additionally, while the R programming language is supported, most of the documentation is in Python.

2) Custom Dataset Support: While AIF360 does support use of custom datasets, the authors found several issues when attempting to implement custom datasets. Due to lack of documentation, understanding the implementation of a custom dataset using the Standard Dataset function was difficult. When using the built-in preprocessing techniques, each custom dataset processed resulted in errors. The level of effort trying to implement a custom dataset caused this sub criterion to be scored as a 2.

3) Frequency of Updates: AIF360 is actively updated and maintained according to its GitHub history. At the time of this paper, content in various folders has been updated within the past 2 months. This sub criterion scored a 10.

4) Programming Languages Supported: This tool supports both the Python programming language as well as R. The sub criteria received a ranking of 7 due to the lack of complexity for the R programming language as it has only a fraction of the available functions that Python has. This unbalanced development causes this sub criterion to be scored as a 7.

5) Stage of ML Lifecycle Supported: AIF360 supports the three major machine learning life cycle stages: pre-processing, in-processing, and post-processing. Within each major stage, there are multiple algorithms which will be discussed in a later judged sub criterion. This sub criterion received a score of 10.

6) Development Environment Support: The tool does not have any known issues with regards to development environments. This sub criterion scored a 10.

7) Programming Language Version Support: AIF360 supports both Python and R. With regards to Python, it supports version 3.6 through 3.8. As of October 2020, Python’s latest version is 3.9 meaning AIF360 does not support the most current version of the Python language. With regards to R, there is very little documentation surrounding the versions supported. Due to the lack of current version support in Python and lack of documentation surrounding R, this sub criterion was scored a 5.

8) Size of Dataset: According to the documentation and number of examples provided by AIF360, there is no known size limitation when it comes to processing a dataset. Built in functions can split data into training and test data regardless of the number of records. This sub criterion scored a 10.

9) Execution Time Reasonable: While this sub criterion is subjective in nature due to the hardware performing the computations, AIF360 being used on commodity grade hardware performed training of models within reasonable durations of time. This sub-criterion scored a 7.

10) ML Model Algorithm Count: AIF360 supports multiple machine learning algorithms through multiple stages of the machine learning lifecycle. Taking into account that the R language support is not supported as extensively as Python, this sub-criterion scored an 8.

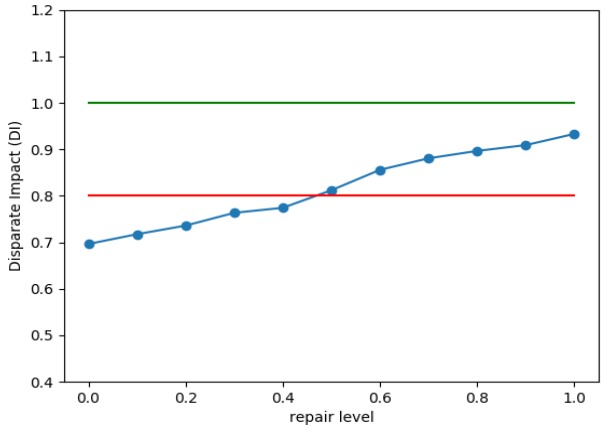

Fig 4. AI Fairness 360 Example of Disparate Impact Remover Algorithm

Figure 4 is an example provided by AI Fairness 360 where the Disparate Impact Remover Algorithm. This algorithm corrects for imbalance selection rates between unprivileged and privileged groups at various levels of repair [27].

B. TensorFlow Fairness Indicators

TensorFlow Fairness Indicators is an open-source tool created by Google. TensorFlow was developed by using C++ and CUDA, but it also supports Python.

1) Documentation: Most of the Fairness Indicator documentation is from TensorFlow Fairness Indicators official website unlike AI Fairness 360. Due to this limited documentation, Fairness Indicators got only a 5 out of 10 for this criterion. It is difficult to find useful documentation through the official website.

2) Custom dataset support: Fairness Indicators does not have much in the way of custom dataset support. Even using the dataset, it is difficult to get the code to work properly or find a tutorial to help. For this reason, Fairness Indicators receive a 1 out of 10 for custom dataset support.

3) Frequency of updates: The first release version of TensorFlow or Initial release is in 2015 which is called v0.5. After that, the developers enhance operation system support for both Mac and GPU. From the first release, developers spent 12 months developing tools to be able to support Windows 7 in 2016. Tensor1 was released in 2017 for machine learning. Then in 2019, Google announced the TensorFlow2 as the latest version which is developed. The frequency of updating is often so It gets 8 out of 10 scores. There is a channel for supporting users to declare problems and feedback to the tool's operator, so it is a way to communicate and able to improve Fairness Indicators.

4) Programming Languages Supported: Fairness Indicators was developed using C++ and CUDA, but it supports both Python and R languages. However, Fairness Indicators supports Python more so than R. The documentation and support for R are weaker than Python. Another factor is Tensor 2 and Tensor1 have some different syntaxes so programmers should remember this when coding in different versions. The sub-criteria rank is 6 out of 10 scores.

5) Stage of ML Lifecycle Supported: Fairness Indicators supports all three-machine learning lifecycle including pre-processing, in-processing and post-processing by providing the Python package. The sub-criterion is ranked a 10.

6) Development Environment Support: The environment support of Fairness Indicators does not have any issues or problems. It can be used in various environments such as Anaconda and it also can run on Jupyter without launch in Anaconda, so the score is 10 out of 10.

7) Programming Language Version Support: Fairness Indicators supports Python versions 3.6, 3.7, and 3.8(requires TensorFlow 2.2 or later). The latest version of Python is 3.9, but Fairness Indicators doesn't support the current version. For R language, there is not much documentation support. Due to these two reasons, the sub-criterion is ranked as a 5.

8) Size of Dataset: There is no mention of the limitation of the dataset that can be used in Fairness Indicators, so the sub-criteria rank is 10 out of 10.

9) Execution Time Reasonable: Due to documentation limitations of the custom dataset and issues getting the tool to work properly in a Jupyter Notebook environment, Fairness Indicators was not able to run even a sample dataset properly. This caused the sub-criterion to be ranked as a 1 out of 10.

10) ML Model Algorithm Count: TensorFlow's Fairness Indicators are an addon to the TensorFlow Model Analysis library. Because of this it is hard to classify which algorithms Fairness Indicators work on. The inability to

separate the packages to ultimately determine a count of supported algorithms results in this sub criterion being scored as a 5.

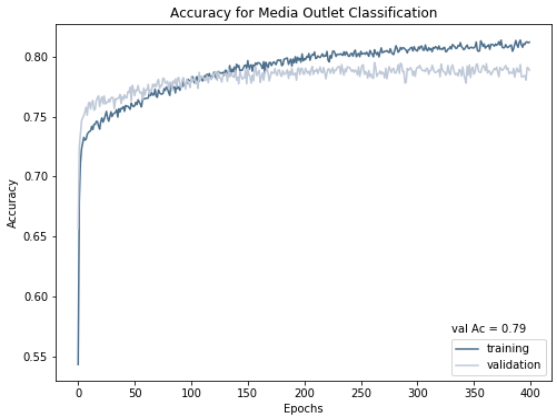

Fig 5. Fairness Indicators example of accuracy for media outlet classification

Figure 5 is an example of bias detection by using TensorFlow's Fairness Indicators. After training, the figure shows the outlet classification with the result of validation and training accuracy curves.[28]

C. Audit-AI

Audit AI is a bias detection tool built-in python by the pymetrics team. The purpose behind the creation of audit AI was to create a tool to both measures and mitigate the effects of discriminatory patterns in training data.[23]

1) Documentation: Audit AI documentation primarily resides in GitHub in the form of examples. On the main website there is an overview of the tool and in what cases the tool would be useful. However, documentation lacks in depth discussion on the tool as well as in depth explanation of the examples. There is an assumption that those wanting to use the tool are very proficient in the field. Due to the level of documentation this criterion has been ranked at a 6

2) Custom Dataset Support: Custom datasets are supported however there were several roadblocks along the way hindering the authors from doing so. Of all the examples provided there are none that use "custom" data, and there is no explanation on what needs to be done to the data in order to make it compatible with the tool. Once understanding what needs to be done to the data set in order to make it compatible it takes a copious amount of time and effort. Extensive background knowledge and experience required to make a properly formatted dataset. Due to the amount of effort needed the criteria has been scored at 2. Any attempt without fail resulted in errors

3) Frequency of Updates: Despite being a fairly new tool, updates are not incredibly frequent. From pre-release in 2018 there have been a total of seven version updates. This being said there has been about two to three updates per year up until July of 2020 with the last update on July 17th 2020. Due to frequent updates in the past but lack of updates as of right now this criterion has been ranked at a 6

4) Programming Languages Supported: Being a tool that was created with the Python libraries in mind, it is only supported by the Python programming language. Audit AI supports the programming language it was made for very well but because it only is supported by one language this criterion has been ranked at a 8.

5) Stages of ML lifecycle supported: Pymetrics Audit AI supports the three major machine learning lifecycle stages, these being pre-processing, in-processing, and post-processing. Of these three major stages, there are algorithms that supplement each of the stages. Due to this this criterion has been ranked as a 10

6) Development Environment Support: As of the time of this paper there are no development environments that have reported not being able to use Audit AI. All the GitHub examples use Jupyter; however, any environment that supports Python and all its libraries will be able to run Audit AI. This criterion has been ranked at a 10 because of this.

7) Programming Languages Version Support: Due to Audit-AI being developed on the back of existing python libraries, specifically pandas and sklearn, Audit-AI supports both Python 2 and Python 3. Due to these being the most common Python versions this criterion has been rated at an 8

8) Size of Dataset: The dataset example used for the purpose of this analysis was the German credit dataset and consisted of 1000 rows and 51 columns. There is no record of a limitation on the size of the dataset. There are built in methods to create both a training and test set from the records provided. This criterion was ranked at 10.

9) Execution Time: We found that the speed at which the algorithms run is dependent on the device. For one of our machines using Jupyter Notebook the "plot_threshold_tests" takes around 9.7seconds to perform. Due to this the criterion has been ranked at 8.

10) ML Model Algorithm Count: When it comes to methodology diversity, there are eight algorithms to highlight. These algorithms consist of 4/5th, fisher, z-test, bayes factor, chi squared, sim_beta_ratio, classifier_prosterior_probabilities and anova. Due to this diversity the criterion has been ranked as an 8.

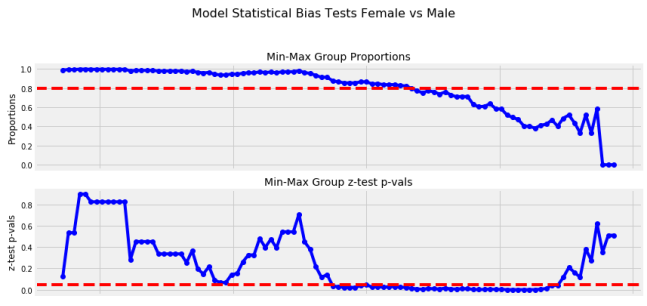

Fig 6. Pymetrics Audit-AI visual output for the German Credit Dataset

Figure 6 is two examples of the German credit data outputs visualized. The first is the Min-Max Group Proportion, this is signifying whether each record is above or below the 4/5ths benchmark. Essentially checking whether the highest passing groups of Male and Female are within 80% of each other. The second test is the Min-Max Group z-test p-values. This test is focusing on whether there is a significant difference between the values and the null hypothesis, since so many of the records are far from 0 it is evident that there is a distinct difference between the Male and Female groups.

D. Skater

1) Documentation: When compared with other tools, Skater is the fewest documentation. There is not much documentation about the demo or tutorial. The sub-criterion is 3 out of 10.

2) Custom dataset support: There is no document support or available tutorials for use of a custom dataset. Most of the documentation surrounding datasets is related to the sklearn dataset repository as Skater publicizes this as an available document source. The criterion is ranked as 1 out of 10.

3) Frequency of update: Skater's last published version release was released in 2018. Looking at the commits in the GitHub repository, the latest updates made were over 11 months ago. Due to this lack of apparent updates, this tool was ranked as a 1 for this sub-criterion.

4) Programming Languages Supported: Most of the documentation that was found was in regard to Python support. The R programming language is supported with reduced capabilities in comparison to Python modules. Due to the disparity between languages, this sub-criterion scored a 7.

5) Stage of ML Lifecycle Supported: This tool supports the in-processing and post-processing stages of the ML lifecycle. Due to the lack of support in the pre-processing stage, this sub criterion scored a 7.

6) Development Environment Support: The latest version of Skater library does not update on conda. The skater can be installed in conda-forge channel in Linux and OS X. Windows installs have been documented as problematic. Due to this inconvenience, the sub-criteria is ranked as 5 out of 10.

7) Programming Language Version Support: Skater supports Python 3.0 or a lower version, but the current version of Python version is 3.9. That means Skater does not support the version of Python from 3.1 until the latest version as 3.9. For R language, it doesn't mention about the specific version of it so the sub-criterion is ranked as a 3.

8) Size of Dataset: There is no limitation of size of dataset on documentation and based on some datasets that are provided, the sub-criterion is ranked as 10 out of 10.

9) Execution time Reasonable: Due to lacking documentation especially for the custom dataset, Skater was unable to be scored with relation to its runtime capabilities. Due to this roadblock the sub-criterion was ranked as 1 out of 10.

10) ML Model Algorithm Count: Skater supports a few different ML models, such as regression, rule-based, classifiers. While not on the scale of some of the other tools, Skater offers somewhat decent support with regards to overall algorithm count. This sub-criterion was ranked as a 3.

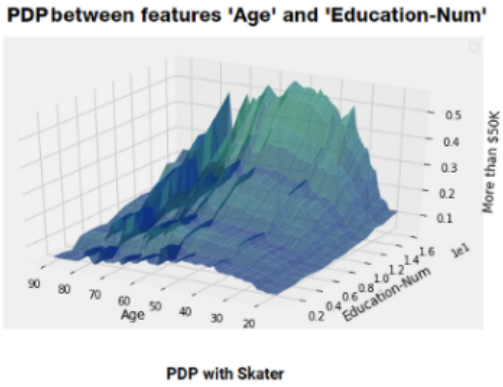

Fig 7. The Effect of Education Level on Earning More Money by Using Skater

Figure 7 is an example of using Skater to study the effect of Education Level on earning more money. Figure 6 shows the running of a deeper model interpretation between Age and Education-Num by visualizing Two-way partial dependence plot.[29]

E. Overall Assessment

	Criteria									
	Ease of Use		Applicability			Interoperability		Performance and Speed		Methodology Diversity
Tool	Documentation (10)	Custom Dataset Support (10)	Frequency of Updates (1)	Programming Languages Supported (3)	Stage of ML Lifecycle Supported (4)	Development Environment Support (3)	Programming Language Version Support (1)	Size of Dataset (5)	Execution Time Reasonable (3)	ML Model Algorithm Count (3)
AIF360	8	2	10	7	10	10	5	10	7	8
TensorFlow Fairness Indicators	5	1	8	6	10	10	5	10	1	5
Audit-AI	6	2	6	8	10	10	8	10	8	8
Skater	3	1	2	7	7	5	3	10	1	3

Fig 8: Individual Tool Scores Based on Sub-Criteria

Each tool was independently scored on the predetermined AHP criteria on a scale of 1 to 10. The result of each tool is reflected in Figure 8. Each sub-criterion was given a weight of importance determined by the overall assessment. Criteria, such as Documentation and Customer

Dataset Support, were weighted as highly valued attributes whereas frequency of updates and programming language version support were weighted as low importance.

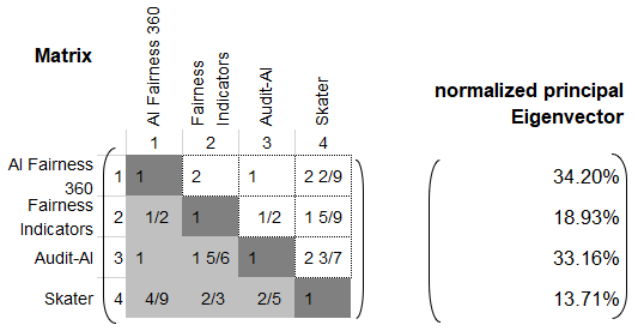

Fig 9. Attribute Matrix and Corresponding Eigenvector Values

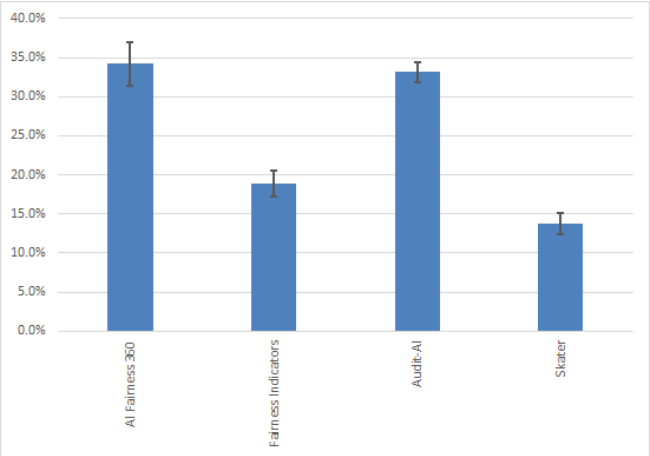

Fig 10: Final Eigenvector Values

This project resulted in two tools that scored higher than the other tools assessed, AI Fairness 360 and Audit-AI. The highest ranked tool was AI Fairness 360 with an Eigenvector value of 34.2%. As indicated in Figures 9 and 10, the tools were compared against each other via an independent tool score per criterion. After each of the scores were recorded per tool, they were input into an Excel template [30] to calculate the Eigenvector value. This calculation resulted in the identification of the chosen tool to recommend based on the authors' assessment.

While AI Fairness 360 was deemed the best tool to use based on the given criteria for this paper, it should be noted that each tool compared in this assessment were created by varying sizes of companies and teams. For instance, while Skater ranked the lowest among all tools, the tool is listed as still in the beta phase. Yet, it still provides several ways to help developers in their efforts to develop machine learning models that will create outcomes that affect many different individuals.

VI. CONCLUSION

Machine learning is an ever-growing, popular domain of artificial intelligence and data science. Due to this popularity, it is important for developers to account for any type of bias in their projects. This paper provides a summary of publicly available tools developed for use in bias identification and mitigation. Like mentioned earlier, bias identification and mitigation tools are not commonly publicized nor available for use. This paper identified four tools that are free to the public and have some level of documentation and community support. It also provides a recommendation based upon the experience the authors had while using each tool. While this paper was able to provide insight into some of the tools available in the market, it highlighted the lack of publicly available tools and subsequently the lack of research surrounding them.

VII. LIMITATIONS AND FUTURE WORKS

One limitation that the authors ran into during their assessment of each tool was being unable to use a custom dataset that would have allowed for consistent testing across all tools. Not every tool has the same output making it difficult to create a one-to-one comparison. Limited documentation of examples using the tools lead to difficulties.

Future works would have a better understanding of the processes that are going on behind the scenes for the identification and mitigation algorithms. A dataset should be found that is known to have bias, preferably one that has been used in other studies. This dataset should have determined protected attributes, significant volume, and contain individual or group bias. This dataset should then be pre-processed according to the best practices recommended by each tool. Additional areas for improvement would include quantifying additional metrics surrounding a tool agnostic custom dataset, such as accuracy, runtime comparisons, among others.

REFERENCES

[1] Hardesty, L. (2019). The history of Amazon's recommendation algorithm. Retrieved 13 February 2021, from https://www.amazon.science/the-history-of-amazons-recommendation-algorithm/

[2] Mitchell, T. M. (1980). The need for biases in learning generalizations (pp. 184-191). New Jersey: Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ.

[3] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2019). A survey on bias and fairness in machine learning. arXiv preprint arXiv:1908.09635.

[4] Stoltzfus, J. (2021). Why is machine bias a problem in machine learning? Retrieved February 13, 2021, from https://www.techopedia.com/why-is-machine-bias-a-problem-in-machine-learning/7/33002

[5] Lim, H. (2020, December 01). 7 types of data bias in machine learning. Retrieved February 14, 2021, from https://lionbridge.ai/articles/7-types-of-data-bias-in-machine-learning/

[6] "Statistical Bias Types explained - part2 (with examples)," Data36, 22-Jul-2020. [Online]. Available: https://data36.com/statistical-bias-types-examples-part2/. [Accessed: 09-Apr-2021].

[7] Biswas, D. (n.d.). Ethical AI: Its implications for Enterprise AI Use-cases and Governance. Retrieved March 13, 2021, from https://www.researchgate.net/profile/Debmalya-Biswas/publication/346789516_Ethical_AI_Explainability_Bias_Reproducibility_Accountability/links/5fd0cdd492851c00f85fb12e/Ethical-AI-Explainability-Bias-Reproducibility-Accountability.pdf

[8] Bellamy, R. K., Mojsilovic, A., Nagar, S., Ramamurthy, K. N., Richards, J., Saha, D., . . . Mehta, S. (2019). AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development, 63*(4/5). doi:10.1147/jrd.2019.2942287

[9] C. Wilson, A. Ghosh, S. Jiang, A. Mislove, L. Baker, J. Szary, K. Trindel, and F. Polli, "Building and Auditing Fair Algorithms," *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[10] M. Almutairi and H. Nobanee, "Artificial Intelligence in Financial Industry", *SSRN Electronic Journal*, 2020. Available: 10.2139/ssrn.3578238.

[11] Trindell, K., Polli, F., & Glazebrook, K. (n.d.). Using Technology to Increase Fairness in Hiring. In *Using Technology to Increase Fairness in Hiring* (pp. 30-36).

[12] Narayanan, A. Translation tutorial: 21 fairness definitions and their politics. In *Conference on Fairness, Account- ability, and Transparency*, February 2018.

[13] T. Mahoney, K. Varshney and M. Hind, "AI Fairness How to Measure and Reduce Unwanted Bias in Machine Learning", 2020.

[14] Dee, D. (2005). Bias and data assimilation. Quarterly Journal Of The Royal Meteorological Society, 131(613), 3323-3343. doi: 10.1256/qj.05.137

[15] Bellamy, R., Mojsilovic, A., Nagar, S., Ramamurthy, K., Richards, J., Saha, D., Sattigeri, P., Singh, M., Varshney, K., Zhang, Y., Dey, K., Hind, M., Hoffman, S., Houde, S., Kannan, K., Lohia, P., Martino, J. and Mehta, S., 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5), pp.4:1-4:15. https://arxiv.org/pdf/1810.01943.pdf

[16] Caliskan, A., Bryson, J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. Science, 356(6334), 183-186. doi: 10.1126/science.aal4230

[17] Datta, A., Tschantz, M., & Datta, A. (2015). Automated Experiments on Ad Privacy Settings. Proceedings On Privacy Enhancing Technologies, 2015(1), 92-112. doi: 10.1515/popets-2015-0007

[18] Thelwall, M. (2018). Gender bias in machine learning for sentiment analysis. Online Information Review, 42(3), 343–354. https://doi.org/10.1108/OIR-05-2017-0153

[19] Selection bias. (2014). Retrieved 16 February 2021, from https://www.iwh.on.ca/what-researchers-mean-by/selection-bias

[20] Ruf, B., & Grari, V. (2019). Understanding and Mitigating Bias [Ebook] (pp. 20-22). AXA. Retrieved from https://axa-rev-research.github.io/static/AXA_Booklet_Bias.pdf

[21] Varshney, K. R. (2019, February 12). Introducing AI FAIRNESS 360, a step Towards TRUSTED AI - IBM Research. Retrieved March 14, 2021, from https://www.ibm.com/blogs/research/2018/09/ai-fairness-360/

[22] TensorFlow pros and Cons - the bright and the dark sides. (2018, September 15). Retrieved March 13, 2021, from https://data-flair.training/blogs/tensorflow-pros-and-cons/

[23] Audit-ai. (2020, July 29). Retrieved March 14, 2021, from https://pypi.org/project/audit-AI/

[24] P. Choudhary, "Interpreting predictive models with Skater: Unboxing model opacity", *O'Reilly Media*, 2021. [Online]. Available: https://www.oreilly.com/content/interpreting-predictive-models-with-skater-unboxing-model-opacity/.

[25] Liberatore, M. (1987). An extension of the analytic hierarchy process for industrial R&D project selection and resource allocation. IEEE Transactions On Engineering Management, EM-34(1), 12-18. doi: 10.1109/tem.1987.6498854

[26] Liberatore, M. (1982). Review of The Analytic Hierarchy Process by Thomas L. Saaty. Amer. J. Math. Manag. Sci., vol. 2, pp. 165- 172

[27] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact." ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015.

[28]"Media Bias Detection using Deep Learning Libraries in Python", *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/media-bias-detection-using-deep-learning-libraries-in-python-44efef4918d1.

[29]"Hands-on Machine Learning Model Interpretation", *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/explainable-artificial-intelligence-part-3-hands-on-machine-learning-model-interpretation-e8ebe5afc608.

[30] K. Goepel, "Implementing the Analytic Hierarchy Process as a Standard Method for Multi-Criteria Decision Making in Corporate Enterprises – a New AHP Excel Template with Multiple Inputs", 2013. Available: 10.13033/isahp.y2013.047.

Our website

AIT 580 Final Project (gmu.edu)