State of the art for fairness detection in Machine Learning dataset

Akhila Sulgante*
Northeastern University

John A. Guerra-Gomez†
Northeastern University

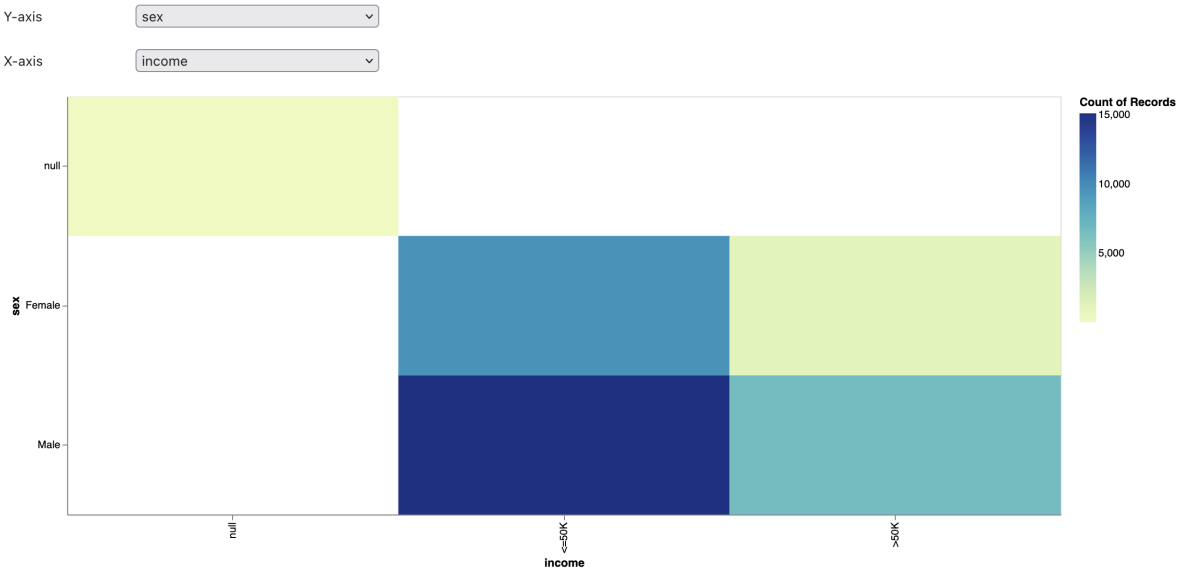

Figure 1: The heatmap above illustrates the attribute correlation between sex and income within the UCI adult dataset. This visual representation reveals a noteworthy pattern: there's a considerably higher count of male records indicating an income >50k compared to the count of female records showing a similar income bracket.

Abstract

This paper addresses the critical issue of bias within ML datasets, uncovering prejudices and inaccuracies [6] that impact the fairness of machine learning models. Beyond technical challenges, biased models perpetuate discrimination across age, race, culture, and other demographics [17]. For instance, Amazon's flawed hiring system discriminated against female candidates, reflecting the systemic biases embedded in ML models [12].

As ML and AI become pervasive, ensuring unbiased decision-making is paramount. The expanding user base, including non-tech specialists like cancer researchers [5], necessitates evolving bias detection tools for broader accessibility. While existing tools like AuditAI, AI360, and Aequitas offer integration into existing workflow and user interfaces, some require in-depth knowledge for bias validation. Other analytics tools like FairVis enable subgroup-level bias identification and comprehensive exploration of ML datasets. Our contributions include a usability study, user interviews, and the identification of key features for an effective bias detection tool. We also propose a web-based, interactive tool emphasizing data visualization and increasing transparency in the Machine Learning models and provide a demo on the Observable platform. Additionally, we present a comparative table detailing the features of both existing and proposed tools.

Index Terms: Machine Learning—Visualization—Visualization techniques—User-interactivity;

*e-mail: sulgante.a@northeastern.edu
†e-mail: jguerra@northeastern.edu

1 Introduction

Bias in the ML dataset indicates prejudices and unfair inaccuracies in the data that is used to train ML models [6]. In simple terms, certain groups of a dataset are overweighted or overrepresented [11]. This bias can have a significant impact on the fairness of ML models. Beyond the technical challenges of ML models producing inaccurate results, their implications extend systemically, perpetuating discrimination based on age, race, culture, or sexual orientation. A real-world case is evident in Amazon's flawed previous hiring system, where research identified the discriminatory nature of their ML model. This system downgraded resumes containing the term "Women," resulting in biased selections that disadvantaged female candidates in the top candidate pool [12].

As ML and AI models invade various aspects of daily life [8], ensuring their absence of bias in decision-making is critical. The impact of biased models ranges from denied loan approvals [14] to unequal sentencing [13], based on systemic disparities. Moreover, the expanding user base beyond tech professionals, including specialists like cancer researchers employing models in their workflows, emphasizes the need for bias detection tools to evolve. While complex bias detection tools with intricate calculations might exist, their accessibility to a wider audience remains limited. Hence, it's pivotal to develop readily available tools that assist each step of bias identification, promoting transparency in model training and outcomes.

Experts have made substantial progress in crafting tools for bias detection. Some outstanding libraries such as AuditAI [2], AI360 [1], FairTest [19], and a few others smoothly integrate into existing workflows, demanding just a few extra lines of code for bias computation. However, certain tools such as Aequitas [18], and Google's What-If tool [4] offer user interfaces facilitating bias identification through simplified steps, often requiring users to possess in-depth knowledge

of terminologies for bias validation. Additionally, a handful of analytics tools such as FairVis [10], SliceTeller [22], and others offer feature visualization and subgroup-level bias identification. These tools enable users to explore ML datasets comprehensively. There are also tools such as Google Facets [3], that allow users to explore their dataset down to a single data point.

In this paper, we present our research contributions outlined as follows:

1. Carried out a Usability study on existing tools
2. Conducted User interviews to pinpoint missing areas
3. Identified critical features for an effective bias detection tool
4. Proposed a web-based, interactive tool emphasizing data visualization
5. Offered a tool demo showcased on the Observable platform
6. Presented a comparative table illustrating features of both existing and proposed tools

After this segment, we'll provide detailed descriptions of the previously mentioned contributions as subsequent sections.

2 Related work

There are already plenty of resources including libraries and tools that aid users in detecting biases in their datasets. Our work takes inspiration and builds on these existing techniques by focusing on user input and interactivity to convert the data and model to a white box. In this section, we will take a deeper look at existing libraries and tools that are used to detect biases in the datasets.

2.1 Library based tools

In this subsection, we will explore libraries that are used to detect biases.

2.1.1 AuditAI [2]

AuditAI is an open-source Python library built upon pandas and sklearn. It was built by Pymetrics. It has been widely adopted in the machine learning community, offering a structured and systematic approach to auditing models and promoting fairness, transparency, and accountability in AI systems. The library uses algorithms such as 4/5th, fisher, z-test, Bayes factor, chi-squared, sim-beta-ratio, and classifier-posterior-probabilities for identifying biases in the classification models and anova, 4/5th, fisher, z-test, Bayes factor, chi-squared, group proportions at different thresholds for regression models. AdultAI offers the option to visualize the results through Matplotlib and Seaborn. It uses explainability and interoperability to offer developers a library assessing and mitigating biases across various domains. AuditAI represents a significant advancement in the pursuit of ethical and unbiased AI solutions but lacks in providing users with an interface. Users can only use AuditAI by installing and importing the library to their notebooks. Users also need to familiarize themselves with the algorithm used for statistical bias computation to grasp fully why their dataset is considered biased.

2.1.2 Fairlearn [9]

Fairlearn is a Python library developed by Microsoft that focuses on access fairness in machine learning models. It provides metrics and algorithms to assess and mitigate bias in Machine Learning and Artificial Intelligence models. The tool decides the model's verdict by measuring its impact on the people. It focuses on two types of harm,

1. Allocation harms: This harm is caused when AI systems either provide or restrict opportunities, resources, or information.

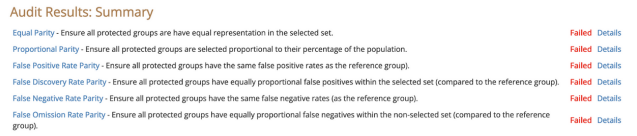

Figure 2: The above image shows a bias report generated from Aequitas user-interface tool.

2. Quality-of-service harms: This harm is measured to check if the quality of service pertains to the system's equal effectiveness for individuals, regardless of whether opportunities, resources, or information are provided or withheld.

The tool uses Demographic parity, Equalized odds, and Equal opportunity as its evaluation metrics. Users can install the tool in Jupyter Notebook using pip or condo and use Matplotlib for visualization purposes.

2.1.3 IBM AI Fairness 360 [1]

IBM AI Fairness 360, often called AI Fairness 360 or AIF360, is an open-source toolkit designed to address biases and fairness concerns in artificial intelligence and machine learning systems. Developed by IBM Research, this library offers a set of algorithms and metrics that enable users to detect, quantify, and mitigate biases across various stages such as pre-process, in-process, and post-process in the machine-learning pipeline. The metrics used to compute bias are Statistical Parity Difference, Equal Opportunity Difference, Average Odds Difference, Disparate Impact, and Theil Index.

The tool is unique because it provides various options for users to mitigate bias. Users can choose the algorithms based on whether they want to handle the bias in pre-process, in-process, and post-process stages. Algorithms available for mitigation include Reweighing, Optimized pre-processing, Adversarial debiasing, and Reject option-based classification. The tool also has a web demo where users can get an overview of the tool, but it needs to be installed with pip or condo to use with its own data and model. It also supports R.

2.2 Interface based tools

In this subsection, we will explore tools featuring User Interfaces, a distinctive aspect that simplifies the detection process for users.

2.2.1 Aequitas [18]

Aequitas, an open-source bias audit toolkit originating from the University of Chicago, is developed explicitly for the machine learning workflow. This tool plays a pivotal role in detecting biases across multiple stages, spanning from data preparation to model evaluation. The tool is versatile, serving as a library that can be integrated into your code or as a user interface. Aequitas employs various metrics to identify biases, including Equal parity (ensuring equal representation among selected groups), Proportional parity (representation proportional to the overall population), False positive parity (equal False Positive Rates across groups), and False Negative parity (equal False Negative Rates among groups). The website version offers a straightforward user experience, accessible even to individuals without extensive machine learning expertise. The instructions are notably clear, comprising four simple steps: uploading the data, selecting the relevant groups, choosing fairness metrics, and generating a comprehensive bias report. The tool provides a "Bias report" as output.

2.2.2 Google's What-If Tool(WIT) [4]

Google's What-If Tool stands as a platform designed to uncover biases within machine learning models, offering a user-friendly interface facilitating the exploration of model behavior, comparison of

outcomes across diverse demographic groups, and detection of potential biases. This tool empowers data scientists and developers to delve deeper into model predictions, evaluating fairness and enhancing transparency in AI systems, thus becoming a crucial asset for promoting equity. Its calculations cover metrics like Demographic parity, Equal Opportunity, Equal Accuracy, and Group Threshold. Offering multiple integration options such as notebook integration or use within the Colab platform (Cloud option), it enables visualization through TensorBoard, albeit not being a web-based tool. The tool facilitates exploration through three distinct ways: the Datapoint editor which allows feature editing and model prediction review, Performance and Fairness, focused on assessing model fairness via the confusion matrix, and Features, which displays histograms for individual features. To identify bias, the tool employs a Confusion matrix, providing users with options like custom thresholds or a single threshold optimized for all data points based on specified cost ratios.

2.2.3 FairTest [19]

FairTest is a Python application developed at Columbia University. FairTest enables developers or auditing entities to discover and test for unwarranted associations between an algorithm's outputs and certain user subpopulations identified by protected features.

The tool works by learning a special decision tree, that splits a user population into smaller subgroups in which the association between protected features and algorithm outputs is maximized. FairTest supports and makes use of a variety of different fairness metrics each appropriate in a particular situation. After finding these so-called contexts of association, FairTest uses statistical methods to assess their validity and strength. Finally, FairTest retains all statistically significant associations, ranks them by their strength, and reports them as association bugs to the user. It leverages the rpy2 Python package, offering a Python interface for the R programming language, and necessitates the installation of R. Besides its standalone library use, It is adaptable for deployment as an online service but only locally. The prototype allows multiple users to conduct investigations. Through a web interface, users can submit investigations and access the respective bug reports once the experiments conclude.

2.3 Visualization focused tool

In this subsection, we will discover tools that focus on visualizing the dataset to identify bias. These tools are helpful in the Data selection stage of the Machine Learning pipeline.

2.3.1 FairSight [7]

FairSight, developed by FairDM, is a platform dedicated to assessing fairness within machine learning models. Built with React and Django, it utilizes D3.js for visualization purposes. This framework remains model-agnostic, aiming to establish a fairness pipeline that seamlessly integrates fairness assessment at each stage, from input to output within the workflow. The tool facilitates bias identification throughout pre-processing, in-processing, and post-processing stages. It offers insights into how various steps in the machine learning process might lead to biased or unfair decision-making, measures existing or potential biases, identifies potential sources of bias, and provides avenues for mitigating bias through diagnostic actions. The rationale for each recommended action is provided subsequently.

2.3.2 FairVis [10]

FairVis stands as a visual analytics platform designed for auditing classification models, particularly focusing on intersectional bias(Intersectional bias acknowledges that individuals may face distinct and compounded forms of discrimination when multiple aspects of their identity interact). This tool allows users to generate data

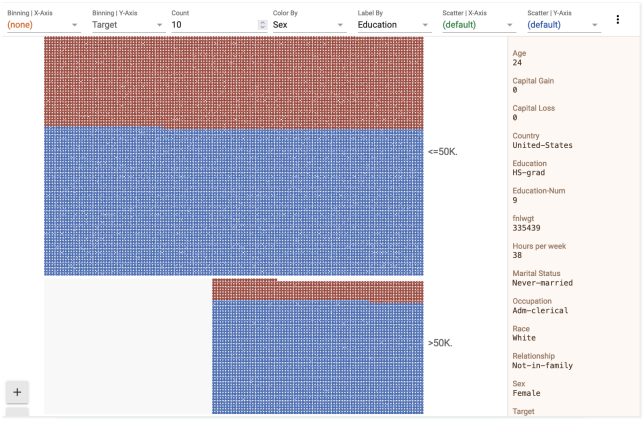

Figure 3: The above image shows the facet's Dive feature. It differentiates datapoint by sex(color) and bins them in income > or <= 50k.

subgroups and investigate whether the model demonstrates disparities in performance across diverse demographic groups. Metrics are calculated with a focus on subgroups, utilizing the Confusion Matrix to identify biases based on these subgroup differentiations. Users can view dataset feature distributions and generate subgroups for analysis. Subgroup performance can be visualized concerning selected metrics, allowing a clear understanding of their behavior. The platform facilitates comparisons between different subgroups, providing detailed insights. Moreover, FairVis identifies suggested underperforming subgroups and suggests similar groups for further investigation.

To utilize the tool, users are required to clone the git repository, install npm, and run it locally. Additionally, a web demo is accessible, offering select datasets for users to experiment with the tool hands-on.

2.3.3 SliceTeller [22]

SliceTeller is a visualization aimed at streamlining the debugging, comparison, and improvement processes for machine learning models, especially those reliant on data slices. By automating the detection of problematic data slices, the tool provides valuable insights into the causes of model biases. It introduces the SliceBoosting algorithm, enhancing the estimation of trade-offs in prioritizing slice optimization. Additionally, this system empowers model developers by facilitating easy comparisons and analyses of various model iterations, allowing them to identify the most suitable version for specific applications. The system's effectiveness is evidenced through the evaluation of three use cases, including two real-world product development scenarios, showcasing SliceTeller's capacity to elevate the debugging and refinement of high-quality machine learning models.

2.3.4 Facets [3]

Google's Facets is a web-based interactive tool. The tool provides users with two highly interactive visualizations that aid in comprehending and analyzing machine learning datasets. With the Overview feature, users can grasp the distribution of values across their dataset features, gaining a quick understanding of its shape. This tool uncovers various issues such as unexpected or missing feature values, training/serving skew, and discrepancies among different dataset splits. On the other hand, Facets Dive provides an interactive view for exploring relationships among data points across diverse features. Users can visualize each data point individually and organize them based on feature values in multiple dimensions. Dive's also helps in identifying classifier failure, pinpointing systematic errors, validating ground truth, and exploring potential new signals for ranking.

3 DESIRED FEATURES FOR AN ML DATASET FAIRNESS ASSESSMENT TOOL

In this section, we will detail the features that a tool should have to effectively detect biases in the ML datasets and models.

3.1 Gathering requirements

Two primary methods used to gather the desired features for the tool are

3.1.1 Conducting Usability study on the existing tools

As our objective was to create an interactive web-based tool, we performed a usability study on various tools featuring user interfaces or web demos. This exploration aimed to comprehend the procedural flow and gather insights on enhancing user interaction and data-model interactivity. The intent was to empower users with the freedom to make informed choices within the tool's interface. This involved noting down key aspects that could augment user engagement and facilitate better decision-making.

Some of the tools assessed during the study are

*we used the UCI adult dataset across all tools

1. Aequitas: We focused on using the website version of the tool to understand the complexity of the process. Aequitas offers user-friendly features, making it accessible to individuals who prefer avoiding code intricacies before dataset selection. The tool allows users to upload their datasets for assessment, following a three-step process leading to the final report. While the interface is straightforward, the tool's technical aspect lies in its explanation of bias metric calculations. However, it lacks in-depth explanations of these metrics, leaving the bias calculation process somewhat opaque. Users receive a generated report after inputting the protected attributes to be audited for bias and Disparity Intolerance, leading to a somewhat black-box approach in the bias assessment.

2. IBM AI Fairness 360: Despite being primarily a library-based tool, it offers users a web demo for familiarization. Beyond bias detection, the tool includes a mitigation plan. However, Users lack permission to upload custom datasets and are limited to pre-existing options. Additionally, the tool lacks the ability for users to select protected groups; instead, it generates a bias report based on the dataset provided. Subsequently, users are prompted to decide on mitigating bias either during pre-processing, in-processing, or post-processing stages, involving data adjustment, classifier modification, or prediction alterations.

3. Google's What-If Tool: The What-If Tool provides extensive platform versatility, offering users the option to utilize it across various platforms such as Colab Notebook, TensorFlow, and cloud AI models. Our evaluation primarily centered on the tool's web demo to explore its ease of use and interactivity. This tool presents unique functionalities, including the Datapoint Editor, enabling users to modify Datapoint features and view model predictions after alterations. Our primary focus remained on assessing Performance and Fairness, while the Features section showcases histograms for individual dataset features. The tool conducts bias assessment via the confusion matrix, allowing users to select thresholds for data assessment. Moreover, it offers clear documentation for users to comprehend each feature comprehensively.

4. FairVis: FairVis stands as an analytic tool centered on visualization, enabling users to explore bias within subgroups. To utilize the tool on a local machine, users are required to clone the git repository and run it using the node package manager. However, for our study, we utilized the web demo of the tool, which provides two datasets for selection and offers a straightforward usage experience. Users select a dataset and are presented with histograms showcasing all dataset features along with subgroup selection options. The tool displays Accuracy, Precision, and Recall metrics and allows the display of a confusion matrix for the chosen subgroup. Notably, FairVis stands out by enabling users to delve deeper beyond a single level of disadvantaged groups.

5. Facets: Facets by Google is a user-friendly, web-based interactive tool designed to help users visualize their machine-learning datasets down to individual data points. The tool offers two distinct views: Facets Overview and Facets Dive, both of which are intuitive and visually engaging. Facets Overview presents data distributions for each feature, effectively segregating numerical values and categorical attributes. On the other hand, Facets Dive enables users to bin features, alter colors, and label data points to represent desired attributes. While providing comprehensive dataset visualization, Facets doesn't offer bias assessment. The tool leaves the identification of bias to users based on metrics and doesn't engage with model training and evaluation.

3.1.2 Interviewing experts to understand the industry standards

We interviewed a Machine Learning Engineer, Data Scientist, and AI Infrastructure Engineer to grasp the workflow involved in developing Machine Learning models and gauge the significance attributed to identifying biases. Deciding on a set of predefined questions, we uniformly queried all participants. Below are the questions asked during these interviews.

- Q1 What procedures do you follow when developing a model?
- Q2 What is the time dedicated to Data Analysis in your workflow?
- Q3 How do you ensure the accuracy of the representation of the data?
- Q4 Could you recount an instance where you recognized bias toward a disadvantaged group in a model?
- Q5 At what stage is it most effective to detect data bias?
- Q6 Which tools do you employ for data examination?
- Q7 Any final reflections on challenges associated with ML models?

The responses to the above questions are compiled below in a collective format.

Q1: The workflow involves several key steps: understanding the problem, assessing the necessity of requiring an ML model, data collection, exploring data (including handling null values, biases, and data correlation), and segmenting data into train, test, and validation sets. Determining the most suitable models by considering pre-trained models but selecting based on dataset alignment, makes it an iterative process. Followed by hyperparameter selection and model evaluation, primarily focused on detecting under-fitting or over-fitting problems.

Q2: The exact response from Participant 1 was "The approach varies based on the problem at hand. For MVP projects, extensive time isn't allocated for data analysis. Typically, readily available online data is utilized, with basic exploration to identify outliers and null values.

In contrast, high-scale projects involve different practices. Online data isn't the preferred choice; instead, clients provide the data. Here, more time is invested in data exploration, emphasizing dimensionality reduction and outlier identification. Additionally, consulting domain experts is common when considering feature removal."

Q3: The response from Participant 2 was "Multiple sampling techniques exist to address data imbalances. In critical tasks, the most effective approach often involves under-sampling."

Q4: The response from Participant 1 was "In the ongoing project aimed at firetruck identification, where simulated data is being used. However, due to limited variation within the dataset, the model tended to classify all boxy-shaped objects as fire trucks."

Q5: The response was standard across all participants and it was emphasized that it is important to identify bias at both Pre and post-model evaluation. The data should be tested and validated to identify any kind of bias or data drift.

Q6: The participants' responses varied: the data scientist highlighted a preference shift based on the project, whereas the Machine Learning Engineer favored Matplotlib, Seaborn, and Dataminer by Skyline. The AI Infrastructure Engineer specified the reliance on internal organizational tools. However, they all agreed that they prefer a web-based tool compared to writing code for visualization.

Q7: All the participants had the same conclusion that the accuracy depends on bias and if a model is impacting a person's life we need to absolutely make sure there are no biases.

From the above two methods, we present the list of features that an ideal bias detection tool should pose.

1. **Visualization [20]**: Visualizations simplify the identification and response to dataset distributions, particularly during the crucial stage of dataset selection for model training. The primary benefit lies in swiftly detecting biases solely through visualization, even without a comprehensive understanding of the dataset.
2. **Bias Metrics [15]**: The tool should offer a range of bias metrics that can capture different aspects of bias, such as disparate impact, disparate treatment, and disparate mistreatment. These metrics help assess fairness from various angles. The data should be tested for equal parity, Proportional parity, False positive parity, and False Negative parity
3. **Explainability [16]**: It should provide explanations and insights as to why the data is biased, such as highlighting the specific features or data points contributing to bias
4. **Compatibility with Frameworks**: The tool should be compatible with common machine learning frameworks and libraries like TensorFlow, PyTorch, sci-kit-learn, etc
5. **Demographic selection**: The tool should allow users to specify and analyze bias for different demographic groups, such as race, gender, age, and other protected attributes. It should also support multiple options to identify biases that affect multiple groups simultaneously
6. **Threshold Setting [21]**: This is the most important feature of the tool. It should allow users to set thresholds for bias metrics to define what constitutes an unacceptable level of bias. This enables users to judge based on specific use cases and fairness requirements

4 Tool Proposal

This project aims to develop a highly interactive user-friendly web-based data visualization and Machine Learning integration tool. The tool will specifically focus on Machine Learning (ML) and Artificial intelligence (AI) datasets, offering users an intuitive platform to explore and analyze complex data structures. One primary objective is to enable users to identify and understand potential biases within the datasets, fostering transparency and accountability in machine learning processes.

Motivation

The currently available tools for assessing bias in machine learning datasets generally fall into two categories: library-based solutions and user interface (UI) counterparts. Library-based tools require users to possess a proficient understanding of machine learning models and coding skills to effectively leverage these libraries for bias detection in their datasets. On the other hand, UI counterparts of certain existing libraries, like Aequitas and Google's What-IF, aim to simplify the process by providing a graphical interface. However, these UI tools often make bias determinations autonomously, potentially limiting user control and customization. While tools like FaiVis provide a comprehensive view of data based on subgroups, they are not readily available to use on the web.

As Machine Learning models branch out into various fields, it's vital to acknowledge their usage as black boxes. Therefore, emphasizing transparency and making both the data and model as universally understandable as possible becomes key.

Target Users

1. Machine Learning Engineers: ML engineers can leverage the tool to analyze and visualize biases in datasets, enabling them to fine-tune models for improved fairness
2. Software Developers: Developers involved in building machine learning applications can use the tool to comprehend and address biases in their datasets, enhancing the ethical implications of their software
3. Beginner-Level Developers in ML: Developers who are just starting their journey in machine learning can benefit from the tool's user-friendly interface, gaining insights into biases without an extensive technical background
4. Scientists without Technical Expertise: Scientists who may not have a strong technical background but use machine learning in their work can use the tool to understand and mitigate biases in the datasets

4.1 Tool Design

In this section, we will walk through our Objective and motivation in developing the tool. We have also outlined the existing features and provided an overview of the next iteration of the tool.

The primary objective behind this tool is to assist users in pinpointing biases within their dataset both before and after model training. This tool empowers users to customize visualizations, allowing for tailored views of distribution or composition. Moreover, it allows the assessment of attribute correlations, enhancing comprehension of their interconnections.

Sometimes, merely observing data isn't sufficient to identify bias. It is essential to train our models and assess their performance to ensure fairness and avoid disadvantaging any particular group. This tool offers users the choice to train a binary classifier and partition the data based on suspected biased attribute values to get a closer look at Accuracy, Misclassification, Precision, sensitivity, and specificity.

This tool is called CrystalLens as it aims to let users see through their dataset and model. It is developed on the Observable platform. Our goal is to build this tool through multiple iterations, consolidating numerous features in one location for comprehensive bias detection.

These are the tool's existing features, organized in the sequence users will follow when using it.

Step 1: Upload your data: We're in the process of making the tool applicable to all datasets. currently, we're utilizing the UCI adult dataset for our exploration

Step2: Explore your data: Users can select their choice of features to visualize through the search and/or checkbox. Users can select multiple features and also choose how they want to view the data, i.e., distribution or composition. Data distribution will show how data values are spread or distributed across a feature. Data composition indicates the makeup of the feature in a dataset

Step3: Attribute correlation: The tool allows users to pick specific features from their dataset and assist in understanding the relationships and correlations. This functionality allows users to select various attributes or features within your data and analyze how they interact or influence one another. By identifying correlations between different features, users gain insights into potential

Tool	AuditAI	IBM AI Fairness	What-IF	Aequitas	Fairlearn	FairVis	Facets	Proposed Tool
Open-source	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Notebook Integration	Yes	Yes	Yes	Yes	Yes	No	No	No
User Interface	No	No	Yes	Yes	No	Yes	Yes	Yes
Visualization	yes	yes	yes	No	No	Yes	Yes	Yes
Disparity Index	Yes	Yes	No	Yes	Yes	No	No	Yes
Statistical Parity	No	Yes	Yes	Yes	Yes	No	No	Yes
Theil Index	No	Yes	No	No	No	No	No	No
Adverse Impact Calculations	Yes	No	No	No	No	No	No	Yes
WIP	No	Yes	No	No	Yes	No	No	Yes

Figure 4: Comparsion table of existing tools and proposed tool. It compares various tools and indicates if a particular feature is present in the tool or not.

dependencies or patterns existing within the dataset. This feature empowers users to explore the connections between variables, aiding in comprehensive data understanding

Step4: Train your model: The tool enables users to select features from their dataset to train the model. The tool also provides the user the control to decide their train and test split. As an interactive tool, users can see how alterations in the selection of training features and adjustments in the train-test split percentage impact the model evaluation. This tool's interactive nature allows users to see real-time changes in evaluation metrics based on their selections, which helps them gain a deeper understanding of the model's performance and behavior.

Step5: Select the characteristics users believe might exhibit bias: In this version, Race and sex, the features known to demonstrate bias in the UCI dataset, have been hard-coded for demonstration purposes. These are the groups users will examine for bias detection. If a user selects "sex" as the feature for analysis, the tool will partition the dataset accordingly and compute evaluation metrics for each distinct value within that feature, showcasing the results.

Step6: Make decisions with the Evaluation Matrix: This tool strongly prioritizes allowing users to assess dataset bias and ensure equitable and accurate group representation. Thus, the tool provides an evaluation matrix displaying metrics like True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN), Accuracy, Misclassification, Precision, True Positive Rate (Sensitivity), and False Positive Rate (Specificity) and not a decision on whether the dataset is biased or not.

In the next iteration, we plan to,

1. Incorporate Adverse Impact and chi-squared calculations for the dataset
2. Integrate Disparity index, Statistical Parity Difference, and Equal Opportunity Difference into the evaluation table.
3. Include the percentage change with the selection change in the evaluation matrix.

5 The current state of the art

In this section, we'll recap the features of the current tool and compare them with those of the proposed tool.

Figure 2 provides a comprehensive comparison of various tools, aiding users in making informed decisions regarding the selection of the tool that best suits their requirements.

The table measures characteristics such as

1. Open-source: Indicates software whose source code is openly accessible and modifiable by users
2. Notebook Integration: The ability of a tool to seamlessly integrate and operate within notebook environments like Jupyter Notebooks or Google Colab
3. User Interface: The graphical interface that allows users to interact with and control the tool's functionalities
4. Visualization: The representation of data or results through visual formats for easier comprehension
5. Disparity Index: A metric used to quantify differences or gaps between groups for fairness and bias
6. Statistical Parity: Ensures that outcomes are similar for different demographic or user groups, regardless of protected attributes
7. Theil Index: A measure of inequality that is used to evaluate disparities within a dataset
8. Adverse Impact Calculations: Assessment or computation of the adverse effects or impacts related to fairness and biases
9. WIP (Work in Progress): Indicates that tool or functionalities are currently being developed

The tools that are compared are AuditAI, IBM AI Fairness, What-IF, Aequitas, Fairlearn, FairVis, Facets, and the Proposed Tool in this project.

6 Limitation and Future Work

In this section, we will take a look at the limitations of the proposed tool.

Our end goal is to create an end-to-end tool for bias detection for ML and AI systems. The tool aims to be a one-stop solution for identifying and mitigating bias.

The tool currently exhibits limitations that restrict its versatility and comprehensive bias assessment. Firstly, its dependency on the UCI adult dataset for exploration potentially constrains its applicability to other datasets, limiting its scope and usability. Moreover, the hardcoded demonstration of bias in specific attributes like "Race" and "Sex" does not cover all potential bias scenarios evident in different datasets, overlooking other critical biases that require assessment. The tool does not provide a mitigation plan at this stage and the identification of the bias is entirely dependent on the user's judgment.

To address these limitations, future iterations of the tool will focus on several enhancements. Firstly, the tool will aim for dataset generalization, allowing it to effectively operate across a wide range of datasets. Additionally, we aim to provide users with the flexibility to input their chosen attributes for bias analysis rather than relying on predefined characteristics and aim to significantly improve the tool's adaptability. We also plan to integrate more sophisticated bias metrics and analysis techniques beyond basic evaluation metrics would enable a more thorough and comprehensive assessment of biases in the datasets. We also plan to implement a decision support system to aid users in interpreting bias assessment results by offering guidance or suggestions based on the evaluation matrix. Finally, we will be expanding the tool's model training options to include advanced models and algorithms to provide a more diverse and robust model training to ensure greater accuracy and effectiveness in bias detection and mitigation.

7 Conclusions

In this section, we will wrap up the research findings and current functionality of the tool.

Ensuring fairness in AI and ML models is a shared responsibility spanning from the person collecting the data to the developers that develop the models to the authorities that employ the model. It's important to meticulously address biases at every stage of the process.

This research contributes to fostering a fairer AI and ML landscape through two key contributions. Firstly, it offers a comprehensive comparison of existing bias detection tools, aiding developers in selecting the most suitable tool for their specific use cases. Secondly, it provides the development of a web-based interactive tool facilitating visualization of dataset features' distribution and composition. Users can explore attribute correlations, assess bias impact, and conduct various evaluations by manipulating training features and the train-test split within the tool's interface. This tool aims to empower users to understand and address biases within their datasets and models.

ACKNOWLEDGMENTS

The authors wish to thank Kasi Viswanath Vandanapu.

REFERENCES

[1] Ai360. https://aif360.res.ibm.com/. 1, 2
[2] Auditai. https://github.com/pymetrics/audit-ai. 1, 2
[3] Google's facet. https://pair-code.github.io/facets/. 2, 3
[4] Google's what-if tool. https://pair-code.github.io/what-if-tool/. 1, 2
[5] News medicallifescience. https://www.news-medical.net/news/20230319/Machine-learning-applications-for-the-diagnosis-treatment-and-prognosis-of-cancer.aspx. 1
[6] Propublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. 1
[7] Y. Ahn and Y.-R. Lin. Fairsight: Visual analytics for fairness in decision making. *IEEE transactions on visualization and computer graphics*, 26(1):1086–1095, 2019. 3
[8] O. Ameri Sianaki, A. Yousefi, A. R. Tabesh, and M. Mahdavi. Machine learning applications: The past and current research trend in diverse industries. *Inventions*, 4(1):8, 2019. 1
[9] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020. 2
[10] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 46–56. IEEE, 2019. 2, 3
[11] D. Dablain, B. Krawczyk, and N. Chawla. Towards a holistic view of bias in machine learning: Bridging algorithmic fairness and imbalanced learning. *arXiv preprint arXiv:2207.06084*, 2022. 1
[12] J. Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. reuters (2018), 2018. 1
[13] B. Davies and T. Douglas. Learning to discriminate: The perfect proxy problem in artificially intelligent criminal sentencing. 2022. 1
[14] A. C. B. Garcia, M. G. P. Garcia, and R. Rigobon. Algorithmic discrimination in the credit domain: what do we know about it? *AI Soc.*, May 2023. 1
[15] G. Gezici, A. Lipani, Y. Saygin, and E. Yilmaz. Evaluation metrics for measuring bias in search engine results. *Information Retrieval Journal*, 24:85–113, 2021. 5
[16] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller. Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4):e1312, 2019. 5
[17] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021. doi: 10.1145/3457607 1
[18] P. Saleiro, B. Kuester, L. Hinkson, J. London, A. Stevens, A. Anisfeld, K. T. Rodolfa, and R. Ghani. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577*, 2018. 1, 2
[19] F. Tramer, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin. Fairtest: Discovering unwarranted associations in data-driven applications. *arXiv preprint arXiv:1510.02377*, 2015. 1, 3
[20] J. J. Van Wijk. The value of visualization. In *VIS 05. IEEE Visualization, 2005.*, pp. 79–86. IEEE, 2005. 5
[21] E. A. Weaver and G. Richardson. Threshold setting and the cycling of a decision threshold. *System Dynamics Review: The Journal of the System Dynamics Society*, 22(1):1–26, 2006. 5
[22] X. Zhang, J. P. Ono, H. Song, L. Gou, K.-L. Ma, and L. Ren. Sliceteller: A data slice-driven approach for machine learning model validation. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):842–852, 2022. 2, 3