Users' needs in interactive bias auditing tools introducing a requirement checklist and evaluating existing tools

Users' Needs in Interactive Bias Auditing Tools

Introducing a Requirement Checklist and Evaluating Existing Tools

Daphne Lenders[1*] and Toon Calders[1]

[1*]Department of Computer Science, University of Antwerp, Middelheimlaan 1, Antwerp, 2020, Antwerp, Belgium.

*Corresponding author(s). E-mail(s): daphne.lenders@uantwerpen.be;
Contributing authors: toon.calders@uantwerpen.be;

Abstract

In the past, automated decision-making (ADM) models have been shown to adopt biases from the data they have been trained on and make discriminatory decisions based on individuals' gender, age, race and intersections of these. To make sure that these unwanted biases are found before models are deployed, interactive auditing tools, that do not require programming knowledge of their users, have been developed. Since discriminatory patterns are typically quite subtle and may unfold in complex ways, such tools need to offer a wide range of functionalities, to make sure that auditors can detect, understand, and contextualize all the important biases within a model. Many interviews and usability studies have been conducted to identify the functional requirements an optimal tool should satisfy. Yet, there exists no extensive checklist of these requirements, nor is it clear to which extent current auditing tools fulfil them. In this paper, we are the first to provide an overview of currently existing tools, while also encapsulating auditors' functional needs for such tools in one comprehensive checklist. More importantly, we will evaluate each of the existing tools according to this checklist and identify ways their shortcomings can be overcome. Common points of improvement we identified using our checklist, concern the tools' functionality to let users detect complex forms of bias (like intersectional bias) and let users understand the causes of this bias.

Keywords: Bias Auditing Toolkits, Interactive Tools, Functional Requirements, Tool Evaluation

1 Introduction

Over the past years, there has been a surge in the development of automated decision-making (ADM) models used for tasks such as credit scoring or job recruitment. Though powerful and potentially timesaving, these models come with the risk of mirroring discriminatory patterns recorded in the data they are based on. In other words, their decisions might unfairly impact some groups of the population, based on sex, age, race or other characteristics of the decision subjects [1]. One example of this is the infamous COMPAS case, where a model trained to make recidivism predictions, unjustifiably predicted higher risk scores for black than white defendants [2]. To address these fairness concerns, research organizations and legal institutions have emphasized the need to audit the biases of ADM systems before they are deployed. This process has, in the case of hiring and employment systems already become mandatory in New York City, in the form of Local Law 144 [3] and regulations for it are proposed in the EU, in the form of the EU AI Act [4].

The upcoming regulations are an important step towards ensuring more ethical and transparent use of ADM systems, however, beyond specifying that these systems should be audited they do only give minimal guidelines on how this audit should take place or what components it should exist of [5, 6] (for a more elaborate discussion of the regulations we refer to section 5). Thus, there are still many open questions surrounding algorithmic bias audits, such as which components of a model should be audited (e.g., one could only inspect the models' output or also evaluate the code behind the model) or when an audit can be considered to pass or fail [6]. While these are important ongoing debates, the focus of our paper will lie on the part of the audit in which the fairness/accuracy of a model's predictions and its underlying data are assessed and in which way this assessment can take place. According to a study by Constanza et al. this is the most common part of an algorithmic bias audit [6]. For simplicity's sake, we will in the remainder of this paper (without wanting to diminish the importance of other parts of the auditing process) use the term "audit" to refer to this specific part of the process. Further, we will use the term "auditor" to refer to the organization/person conducting this assessment. While the discussion on who this auditor should be is an important one, we push these concerns in our paper aside and merely assume that an auditor is a team or person (that could either be internal or external of the organization whose system is audited) who wants to detect and understand the biases of an ADM model.

Even when making these simplified assumptions, the process of conducting an audit is still complex and challenging. Many considerations need to be made; for instance, how to define bias, which subgroups might be affected by it, and which biases are the most urgent to address. Some examples of the biases that may exist within an ADM model are prediction bias when certain demographic groups are consistently favoured by a model (e.g., a loan-allocation model that hands out more loans to men than other genders) or error bias when a model makes more mistakes for some population groups (e.g., a recidivism prediction model inaccurately predicting that white people are less likely to re-offend).

To facilitate the process of finding and addressing these biases, various tools have been developed to assist auditors. These tools enable them to visually and interactively

inspect the underlying data behind an ADM model, as well as its prediction outcomes on new data (e.g., granting a loan or not) [7–11]. Unlike tools that come in the form of programming libraries these interactive tools are usable by a wide range of people, as their usage does not require technical or coding skills. Additionally, these tools can help in standardizing the auditing process, by providing clear pointers on which considerations need to be made throughout, and on which potential unfairness issues to explore. Lastly, these tools can have a broad impact because they are accessible to the public for free. They can save time and money for users who don't have to start audits from scratch but can use the tools' existing functionality. Despite their clear potential, many tools are developed in isolation of those who might use them, begging the question of how suitable they are in realistic settings. Interview studies with developers and other possible auditors can reveal an answer to this question.

Veale et al., Holstein et al. and Constanza-Chock et al., for instance, conducted interviews with practitioners and auditors to identify their current procedures in testing the fairness of ADM systems [6, 12, 13]. Though they did not directly explore how interactive tools can aid this process they still identified common obstacles that they face, that should be considered when designing auditing toolkits. For instance, they found that auditors often do not have information about decision subjects' sensitive attributes, like gender or race, complicating the process of assessing how a system might impact demographic groups differently. Hence, this reveals the requirement that interactive toolkits should enable a bias audit when sensitive/demographic information is not available.

More recently, other interview studies were conducted in which potential auditors were directly asked to list their requirements in auditing toolkits and identify points in which to improve current ones [14–17]. The studies identified essential user needs, including the requirement for scalable bias audit tools. Since ADM models may make occasional errors, auditors want to avoid wasting time on random mistakes and focus on significant issues that indicate systematic discriminatory patterns [14].

All aforementioned studies uncover important considerations that should go into the design of interactive tools. However, so far only two attempts have been made to give an extensive overview of all these requirements [16, 17]. First, Richardson et al. introduced a rubric listing both functional and non-functional criteria for interactive toolkits [17]. However, this rubric is not specifically targeted to ADM systems and some of its items, like "[tool] can detect bias" or "[tool] contextualizes fairness" remain somewhat vague, and do not provide actionable and concrete suggestions on how to implement them. Nakao et al. also introduce a list of tool requirements. However, they base this list solely on the results of their interview studies and therefore miss essential design needs identified by other research works [16].

In this paper, we conduct a literature review of interview studies with practitioners to provide a more complete list of tool requirements. Further, we give concrete and actionable insights into how these requirements can be implemented. We do so by first examining how some currently available toolkits already fulfil some design criteria so that developers of new tools can draw inspiration from their functionality. Second, if none of our examined tools satisfies a given requirement, we provide pointers to relevant literature that gives insight into how some functionality can be implemented.

In doing so, we are also, to the best of our knowledge, the first to provide a detailed overview of some of the interactive tools that are already available.

2 Overview of selected fairness tools

To give an idea of some of the interactive tools that currently exist, we introduce six tools that we will, later on, evaluate to determine whether they meet the needs of potential auditors. By describing these tools, we do not aim to give a complete overview of all the tools that are currently available but to give an initial understanding of their functionalities. This understanding will form the basis for determining how auditors' functional requirements can be met on a technical level. All the tools we review have an interactive graphical interface, meaning that we excluded tools like AIF360 [18] or FairLearn [19] that come in the form of a Python library. We also only review tools that we were able to use and test ourselves.

To better understand how each tool can be used, we show how each of them assesses bias in the prediction task associated with the "Adult Income Dataset". This dataset contains information on individuals' demographics and working life, like their type of job and their amount of working hours. The associated decision task is to predict whether an individual has a high or low income. We refer to the former as the "favourable outcome" or the "positive label". The dataset contains the attributes "sex", "race" and "age", which are known to elicit biases in ADM models trained on it. We will use the term "protected group" to refer to the group of people that are, based on their sensitive attribute values, historically at lower risk of receiving the favourable outcome than the "unprotected" group. The auditing toolkits described in this section can be used to detect and understand these patterns.

Note that some of the tools are merely prototypes that work only on this *Adult dataset*. Even though they may not be used in real bias audits yet, we discuss their most important components to see how their functionality may be useful to incorporate into future tools.

2.1 Aequitas

Aequitas is a web application that can create bias reports, showing for which groups within a dataset an ADM model satisfies some fairness definitions of choice [20][1]. A visualization of Aequitas' interface is given in Figure 1.

[1] Aequitas also comes in the form of a Python library, with more complex functionalities to detect and mitigate bias in ML algorithms. Since this library can only be used by an experienced programmer, we will only focus on the web application in this paper.

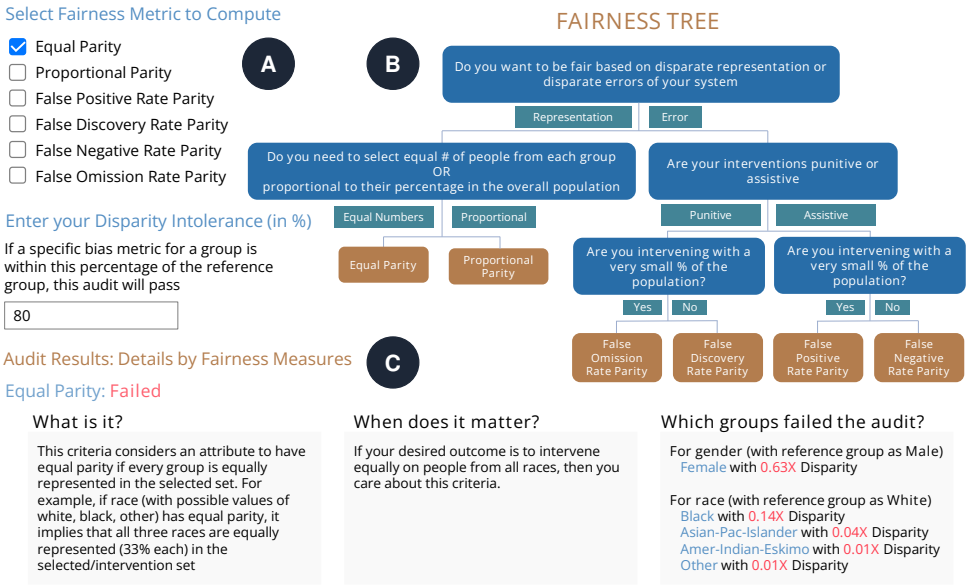


Fig. 1: Visualization of Aequitas: **(A)** After uploading their data and specifying the sensitive attributes (e.g. "gender" and "race") and the reference groups for these attributes (e.g., "men" and "white"), users can select one or more fairness goals as well as a threshold t. Aequitas will then check for each non-reference group, whether the chosen metric does not diverge more than $(1-t)$ from the reference group. **(B)** The "fairness tree" presents a flowchart that is meant to help users in the choice of the fairness metric **(C)** Extract of the bias report: for every fairness metric selected in (A) it is shown whether the model satisfies this metric or not for the given sensitive attributes. In this case we see that the model does not satisfy "Equal Parity" for race nor gender. Along the attribute "sex" it e.g. shows that there is a disparity of 0.63, meaning that the ratio of men and women predicted to have a high income is 1:0.63. An explanation on how the fairness metric is calculated and why it may matter is also provided.

2.2 FairVis

The goal of FairVis is to let users identify intersectional subgroup bias within a model [8]. To this end, users can either compare performance and fairness measures between self-generated subgroups, or explore subgroups on which a model performs unfairly, that are automatically suggested by the tool.

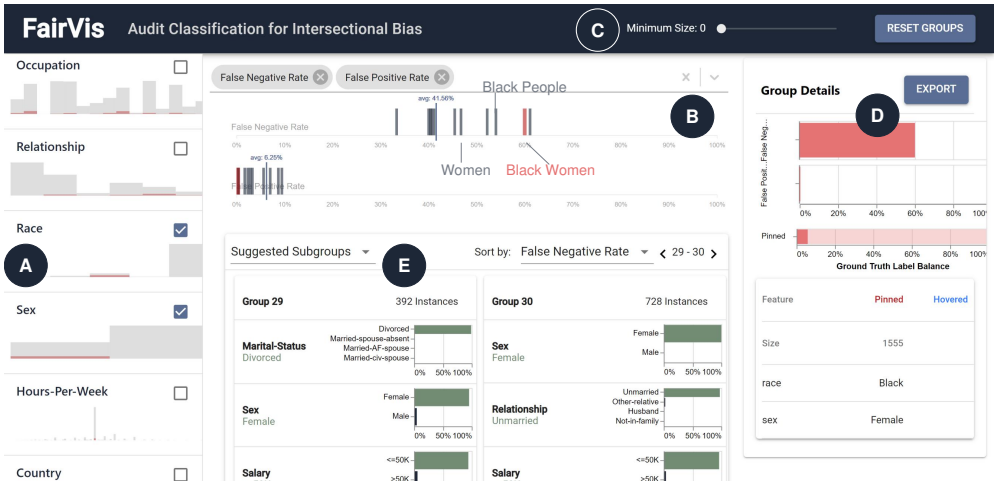


Fig. 2: Interface of FairVis **(A)** Users can select the attributes they want to generate subgroups for; in this case subgroups based on all possible value combinations of "sex" and "race". **(B)** After selecting some performance metrics (e.g., False Negative Rate), the scores of all subgroups (as generated in (A)) on this metric are visualized. Here we see (among others) the model's False Negative Rate for the population of women, black people and black women specifically, whereas the metrics is highest for black women. Using the slider in **(C)**, users can filter out subgroups smaller than a specified size. **(D)** Here users can get additional information on up to two subgroups as selected in (B), namely the number of individuals belonging to the selected subgroups and the positive decision ratio for them in the data. Here the user sees that there are a total of 1555 black women in the dataset, with a positive decision ratio of $\sim$10% in the labels **(E)** Possible subgroups of interest are suggested to the user, sorted according to their score on a performance/fairness measure of choice. Here groups with a high False Negative Rate are suggested, among others, the groups of divorced and unmarried women.

2.3 DiscriLens

The aim of DiscriLens is to visualize discriminatory itemsets, which in this tool are defined as subgroups from the data for which the fairness measure of "conditional demographic parity" is not met. To formally define discriminatory itemsets, assume that we have one sensitive attribute S, one decision variable Y and a set of resolving attributes r. The discriminatory itemsets are then all sets where the conditional demographic parity, defined as $P(Y = 1|S = 1, r) - P(Y = 1|S = 0, r)$, is higher than a threshold τ. In the case of the*Adult dataset*, education and the amount of working hours could be seen as resolving attributes: if an unprotected and protected group (e.g. men vs. women) have the same values on these attributes, but still do not receive a similar ratio of positive decision outcomes, then these group of men and women together constitute a discriminatory itemset.

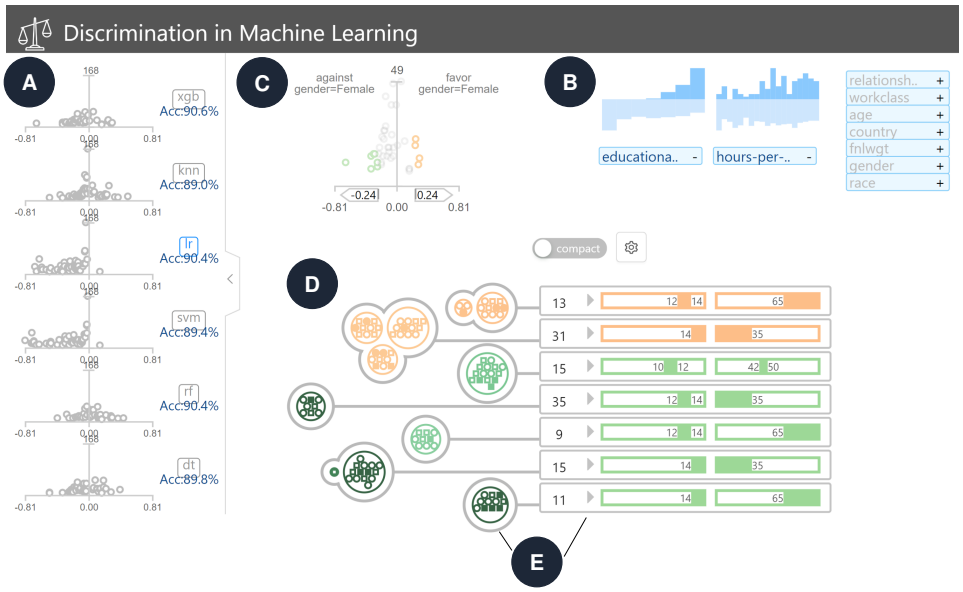


Fig. 3: Interface of DiscriLens **(A)** Users select the ML model from which they want to inspect the predictions **(B)** Users can select resolving attributes **(C)** Visualization of all discriminatory itemsets, where discriminated itemsets are visualized in green, and favoured ones in orange **(D)** All discriminatory itemsets are here visualized through a so-called 'Ripple Set', which encodes information about the direction of discrimination (either in favour or against a protected group), its severity, and its significance **(E)** One of the discriminatory itemsets; in this case the discriminatory itemset consists of people with a higher education level than 14 and more than 65 workinghours. Within this group there are 5 women (visualized by circles) and 6 men (visualized by squares). The fill colour of the shapes denotes whether the individuals received a positive or negative prediction outcome. In this case, 4 squares are filled, meaning that 4 men received a positive decision label, while none of the women received one (as none of the circles is filled). Because of this high discrepancy, the group is marked as a discriminatory itemset.

2.4 FairSight

The goal of FairSight is to let users identify bias in all three stages of the ML pipeline [7], which are defined as follows: the first stage is the "Input", i.e. the data itself and how it may be differently distributed among the protected and unprotected group. The second stage is the "Mapping", which relates to how the input is mapped to the output and whether similar input data receives similar outcomes. The third stage is the "Output", i.e. the ML model's predictions and how they may be different for protected and unprotected instances. Note that, different from the other tools examined in this paper, FairSight does not operate on binary outcomes of a decision task, but on the

rankings generated by a classifier. A classifier's ranking is obtained by ordering the instances according to their assigned prediction probabilities. Assuming that only the top k highest ranked individuals get assigned the positive decision label, bias occurs if there are more instances of the unprotected than of the protected group within this top k. In the following figures, we give a visualization of FairSight's interface. Note that FairSight is currently a prototype, that only works on the German Credit dataset (and hence it is the only tool not working on our running example of the *Adult dataset*). This data consists of information on loan applicants (incl. sensitive information about, e.g. people's age or gender) and the decision label indicates whether a person was approved for a loan or not. In Figure 4, 5 and 6 we give visualizations of the interface of FairSight.

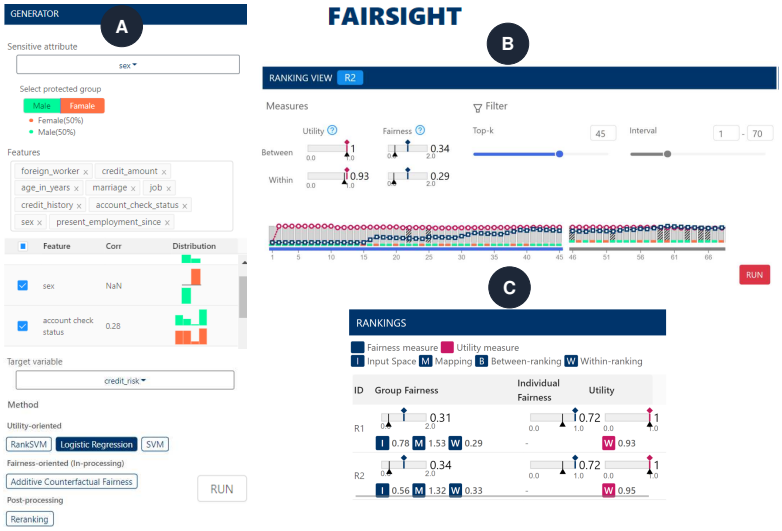


Fig. 4: **(A)** In this "Generator" tab users can select which ADM model to train on the data, as well as the features this model should be trained on. In this case, a Logistic Regression classifier is trained on the features, where "sex" is seen as the sensitive feature (and women are defined as the protected group). Some extra information per feature is given, showing through two histograms how they are distributed differently for the protected and unprotected group **(B)** This is the "Ranking View" tab. After a model has been trained, users can see a visualization of the models' generated ranking, presenting each instance within the ranking as one rectangle, and colouring it according to its protected group membership and its ground truth label (i.e. negative or positive). In this tab, users can also choose a value for k to denote which top-k individuals from this ranking will be assigned a positive label. Some performance and fairness measures are given as well, to show how accurate the ranking is and how fair it is in regard to the proportion of protected and unprotected group members represented in it. **(C)** Here a log is kept of all the ML models the user-generated in **(A)**, and their most important performance and fairness measures are summarized.

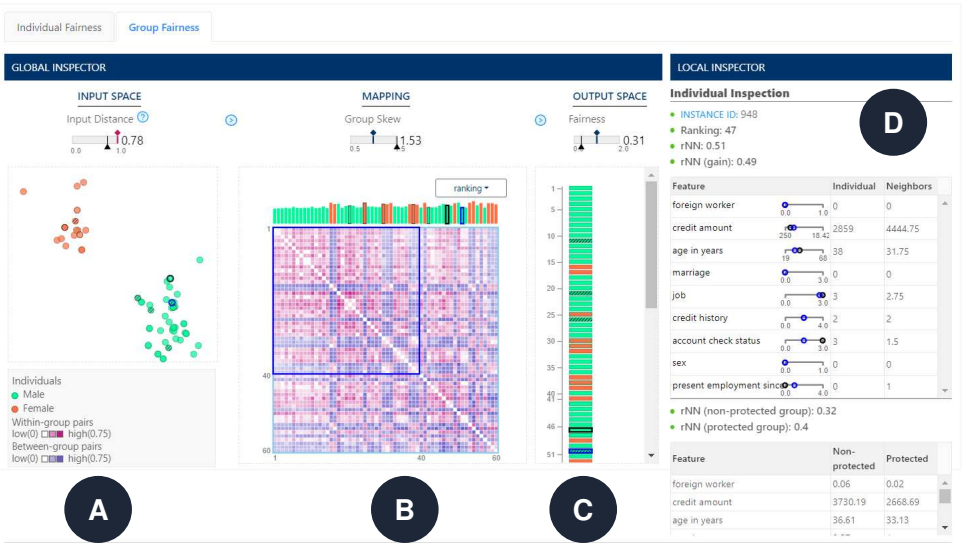


Fig. 5: **(A)** This "Input Space" inspector is meant to show any fundamental differences between the protected and unprotected group, by visualizing a dimensionality-reduced version of the input data, using different colours for both groups (green for men, red for women). In this case, we see that both groups are quite distinct in their input data, indicating that there may be many features correlated to peoples' sex **(B)** This "Mapping" graph visualizes the so-called "distortion" for each pair of input instances. We speak of high distortion when two instances have similar features on the input space, but received different outcomes. The colours encode the degree of distortion and whether two individuals differ in their sensitive attributes. A dark purple colour between two input pairs, for instance, means that the instances have different sensitive attribute values (i.e. denoted by purple rather than pink colour) and the distortion between them is high (denoted by high colour saturation). A user can inspect this graph to get a high-level overview of the distortion within an ML model, and to select individual instances to inspect more closely (see (d)) **(C)** Similarly, as in Figure 4 **(B)** the output ranking of the given ML model is visualized, colour encoding the sensitive attribute of each ranking instance and their ground truth label. We see that a lot more male than female instances are included in the ranking (denoted by colour) and that some of the male instances receiving a positive outcome by the model did not have a positive decision label in the ground truth (denoted by stripes through a block) **(D)** By clicking on one of the instances visualized in (C) users can inspect this instance more closely, and find some measures on how this features ranking position relates to that of its nearest neighbours. If an instance scores low in the ranking, while its most similar neighbours score high, this might be a sign of individual discrimination

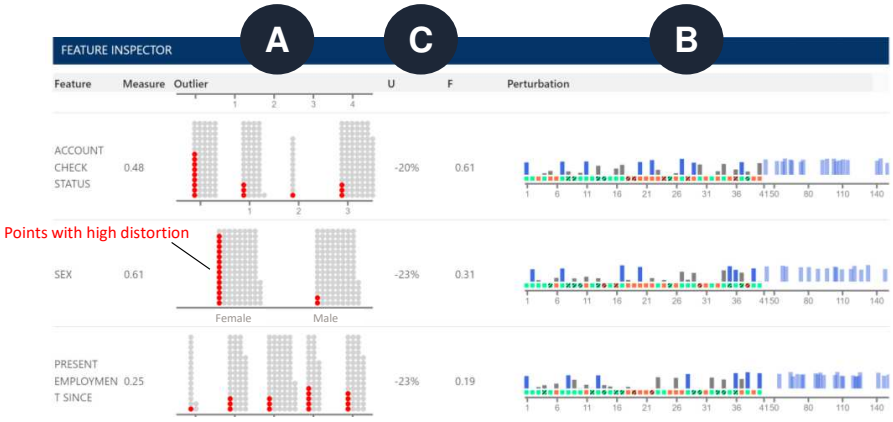


Fig. 6: Visualization of the "Feature Inspector" tab of FairSight. **(A)** For each feature a histogram is visualized. Within each histogram, data points with the highest distortion in the model (i.e. instances whose nearest neighbours receive significantly different prediction outcomes than the instance itself) are coloured red. Auditors can use this, to inspect whether specific feature values are connected with high distortion for individuals with those feature values. For instance, we see that when splitting people by their "sex", most instances with high distortion, are women, indicating that they are the group suffering most from individual discrimination. **(B)** Here the output ranking is visualized, which is obtained when training a model with a perturbed version of the given feature. Here users can check how perturbing the feature (i.e. removing all correlation between the given attribute and the decision attribute) affects the accuracy of the ranking (i.e. how many individuals who are part of the top-k ranking, also deserve a positive decision label), as well as the fairness of the ranking (i.e. how many protected and unprotected individuals are represented in it). Formal measures of this are provided in **(C)**. Here we for instance see, that when perturbing the attribute "sex" the accuracy of the ranking drops by 23%, while the fairness increases by 0.31

2.5 The What-If Tool

The What-If Tool was developed to give users a better understanding of ML models in general, but also specifically in regards to bias in these models [11]. The interface consists of two main components: the "Datapoint Editor" tab (see Figure 7) and the "Performance & Fairness" tab (see Figure 8). Auditors can use the former to visualize data and model predictions, and select data points to obtain further information or conduct individual fairness analysis on. In the latter, users can inspect a model's performance and fairness on subgroups of choice.

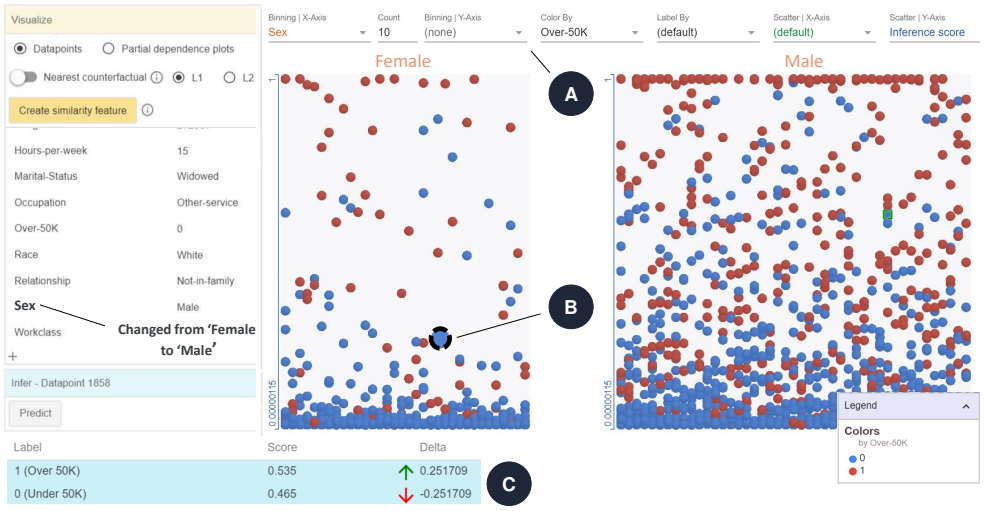


Fig. 7: Interface of the "Datapoint Editor" component of the WhatIf tool. In **(A)** users can select which attributes of the data they want to visualize, using the x- and y-axis in a 2D graph, as well as the colours and labels for each datapoint. In this case "sex" is plotted on the x-axis, and prediction probabilities on the y-axis. This results into two graphs; one for women and one for men. Each point stands for one instance, and their colour denotes their ground truth label (high income - red, low income - blue), while their point in the y-axis denotes their obtained prediction probability (points on the x-axis are randomly spread to increase readability). Here we see, that generally fewer women are represented in the dataset, and it appears that women are more likely to obtain low prediction probabilities. In the graph, users can also select individual data points to either run a counterfactual or a what-if analysis on. (B) In this case, an auditor has selected a female datapoint, to observe how her prediction probability for a high income would be different if her sex was "male" instead. In **(C)** we see that the change in sex increases the prediction probability for a high income by ~ 0.25, indicating a potential case of individual discrimination

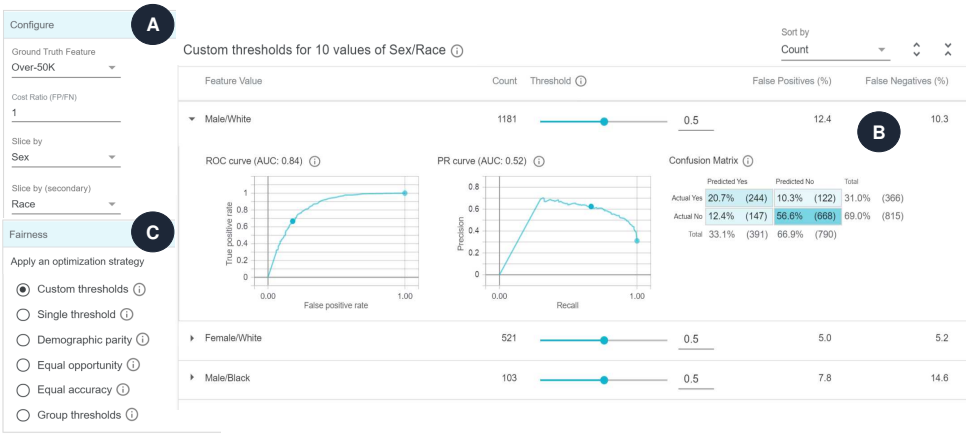


Fig. 8: Interface of the "Performance & Fairness" component of the WhatIf tool. In **(A)** users can select up to two attributes to generate subgroups for (e.g., subgroups based on all value combinations of "sex" and "race"). In **(B)** some performance and fairness measures, as well as the model's ROC curve and confusion matrices for each subgroup are displayed. Users can order these subgroups according to their size or according to one of the fairness/performance metrics. Here, groups were ordered according to size and we can see that the group of white men is most represented in the *Adult dataset*. In this tab, we can also see that they have a higher False Positive rate than all other subgroups. In **(C)** users can try to mitigate the fairness of the model, by adapting the decision threshold to translate a model's prediction probabilities into binary labels. This threshold can either be the same for all subgroups, or differ between them to optimize for a fairness goal of choice.

2.6 The Responsible AI Dashboard

The goal of the Responsible AI (RAI) dashboard is to let users understand a model's errors and behaviour, either for the dataset as a whole or for specific subgroups of the data (which can be generated by the user) [9][2]. The dashboard consists of five main features, which we will each discuss separately: in Figure 9 we show the "Error Analysis" functionality of the tool, in Figure 10 the "Model Statistics" and "Data Explorer" tabs, and in Figure 11 the "Feature Importances" and "What-If Counterfactuals" components.

[2]The RAI Dashboard is part of a larger "Responsible AI Toolbox", which also contains functionalities for bias mitigation. Since, however, these functionalities can only be used by experienced programmers we will only focus on the RAI dashboard (that can also be used by non-technical users) in this paper.

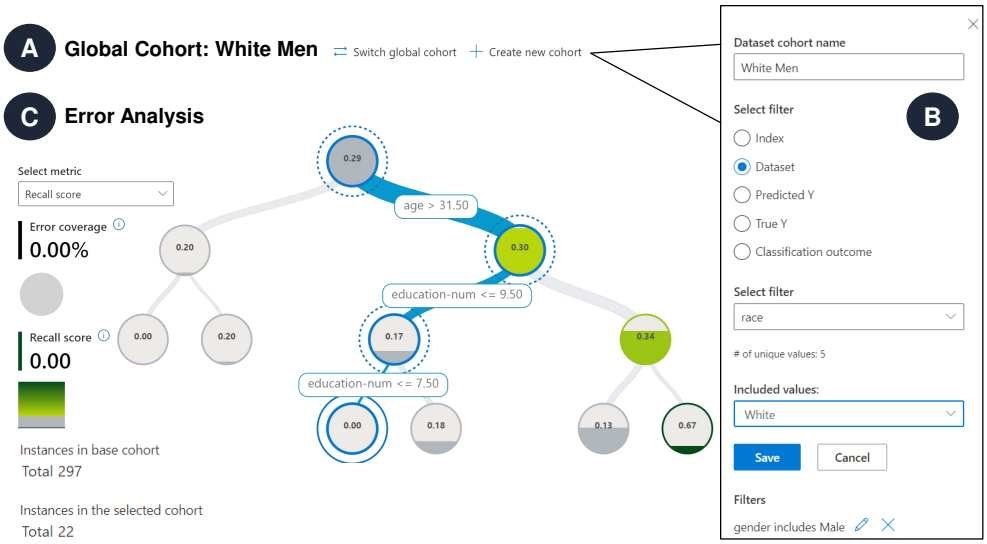


Fig. 9: Visualization of the error analysis component of the RAI dashboard **(A)** The heading shows which part of the data is currently inspected, either the "global cohort" (i.e. all of the data), or specific subgroups from the data, that a user can generate using the interface shown in **(B)**. In this case the user is inspecting the group of "White Men" from the data. As we see, this group consists of a total of 297 instances. **(C)** Visualization of the "error tree", showing how the errors of the ML models are distributed over different subgroups. Given the subgroup of the data the user wants to inspect and a performance measure of choice, it is visualized on which partitions of the subgroup the performance measure is particularly high or particularly low. In this case, we observe the "recall" performance measure, which is over the whole subgroup of white men 0.29. Following one of the paths of the tree, we see that it is with 0.00 even lower for the group of white men older than 31, with a higher education level smaller than 7 (whereas an education level of 7 indicates the completion of 11th grade in High School). In a similar fashion, a user can inspect subgroups with even higher or lower recall measures.

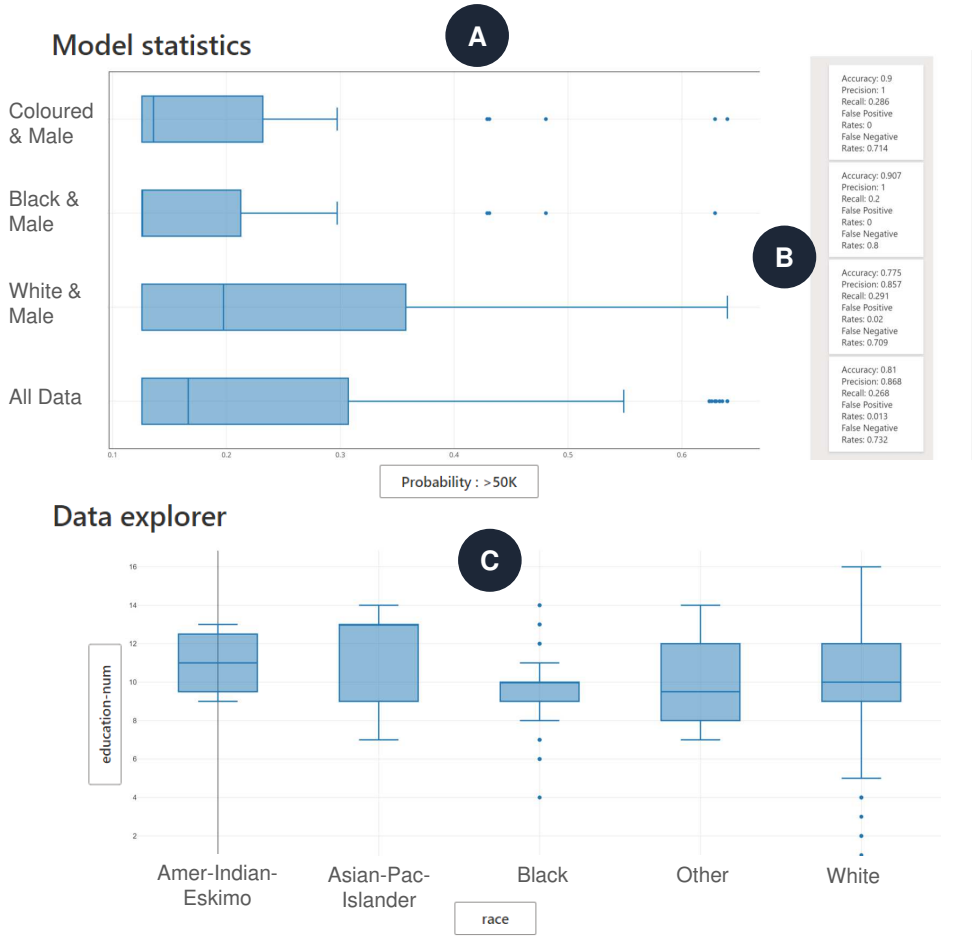


Fig. 10: **(A)** The "Model Statistics" component of the RAI dashboard, where users can explore the performance of the model on the different subgroups. They can either visualize the distribution of prediction outcomes, of prediction probabilities or of the ground truth labels. Here we e.g. see how the prediction probabilities are distributed for men of different races. The boxplot clearly shows that for white men prediction probabilities are higher than for, e.g., black men. Additionally, in **(B)** some performance measures per subgroup are shown, reflecting patterns we saw in (A). **(C)** The "Data Explorer" component: for a subgroup of choice, users can use the x- and y-axis of the graph to visualize the relation between two features for that subgroup. Here we, e.g. see, how education levels differ among races and how people from asian-pacific-islander background, have for instance higher education levels than black people. Studying these patterns, auditors can reason about inequalities present in the input data.

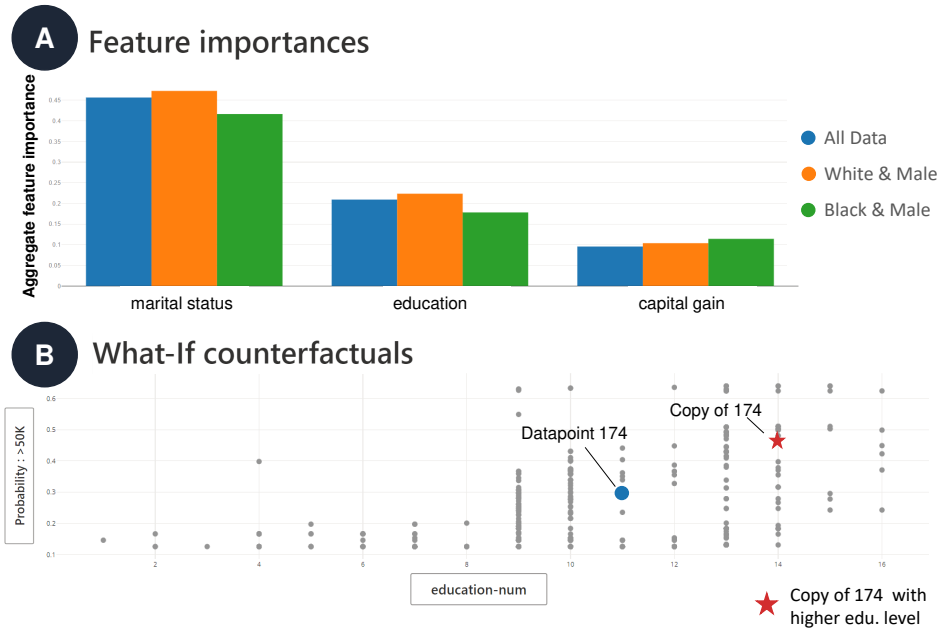


Fig. 11: **(A)** The top k globally most important features for the decision problem are visualized per user-generated subgroup. In this case, we see that for all subgroups (i.e. the complete data, the group of white men and the group of black men) the same three features are most important for being predicted a high income **(B)** This is the "What-if" analysis tab. Here users can visualize the data according to two attributes of choice and select individual instances to run a what-if analysis on. With this, they can test how changing one or multiple feature values of an instance affects the model's prediction probability for it. Here we can see, for instance, how changing the education level of a data instance positively affects its probability for a positive decision outcome

3 Design Considerations for Fairness Tools

While we've only covered a few fairness audit tools, it is evident that interactive tools are developed with different use cases and user needs in mind. However, research has shown that there is a gap between what auditors need from a tool and what functionality the tools offer [15, 17]. To identify this gap, we conducted a literature review to gain insight into auditors' practices and needs that should be accounted for in the design of toolkits. In the next section, we will use this literature review as a basis to compile a list of requirements for toolkits.

We used the earliest and widely-cited key studies by Veale et al. [13] and Holstein et al. [12] as the base for a snowball sampling literature review. Both are interview studies, where practitioners (in case of Veale et al. public sector decision-makers and in the case of Holstein et al. ML developers) are interviewed to understand what

tools, additional research and organizational reforms they need to conduct better fairness audits. Within all papers that cited either of two studies, we used the search query "interview study + fairness assessment" to extract similar papers in which practitioners are interviewed to understand their practices and need for assistance in conducting fairness audits.

Based on the results we identified two relevant lines of research. The first are interview studies, where people working with ADM and other ML systems reported their current practices and obstacles when assessing or ensuring the fairness of these [6, 12, 13]. These studies do not solely focus on the potential of interactive toolkits in addressing these problems but also explore reformations in the organizational and legal sphere.

More recent studies also directly investigated the potential of fairness toolkits, in facilitating bias audits. Here, possible auditors were interviewed or asked to test tools, in order to identify their requirements in them [14, 15, 17].

In section 3.1 and 3.2 we will summarize both lines of research and identify the design considerations (DC) for interactive auditing tools that emerge from them. In doing so we only focus on the considerations for tools that help in detecting bias in ADM systems (and not other ML applications). Further, we will only concentrate on the functionality and not the usability of such tools. As we will see there is a lot of overlap in the design considerations that have become apparent from the different interview studies.

3.1 Exploring current practices for bias detection in ADMs

The earliest significant study on algorithmic bias audits, conducted by Veale et al. [13], involved semi-structured interviews with 27 individuals from the public sector. Semi-structured interviews are a research method where participants are asked a series of predetermined open-ended questions, but the interviewer also has the flexibility to ask additional follow-up questions to explore topics in more depth. In this study, the interviewees, who utilized ADM models for decision-making in areas like taxation or policing, were asked to share their experiences with the models, express fairness concerns, and discuss obstacles they encountered. While many of the reported issues lay on an organizational level, interviewees also revealed some practices and concerns that are relevant to the design of auditing toolkits: they reported that they were aware of discriminatory effects of ADM models and that they, therefore, avoided the use of sensitive attributes when building such models. Further, they were wary of utilizing variables like "home location" in their model, as they might serve as a proxy for the sensitive attribute "race". Still, guidelines of which variables to avoid in models were more of an informal nature, as this is dependent on the decision task. Although not directly mentioned by Veale et al., the fact that practitioners do not directly use sensitive attributes in their models but have no formal way of identifying all proxy attributes poses serious considerations to the design of auditing toolkits: first, if sensitive data is available but not being used, tools should provide the functionality to detect proxy attributes, based on their correlation with sensitive attributes (**DC_identifying_proxies**) Second, if no sensitive attributes are available,

tools should still allow auditors to conduct a fairness analysis. We will refer to this design consideration as **DC_no_sensitive_attributes**.

One year after the interview study by Vaele et al., Holstein et al. released another study building on top of their results. They conducted 35 semi-structured interviews with ML developers, to find out about the obstacles they experience when assessing and improving the fairness of ML systems. After the interviews, they also conducted a survey to see whether their results were generalizable to a wider public. Similarly, as in Veale et al. many of the identified issues lay on an organizational level or were specific to ML applications that are not the focus of our paper. Still, practitioners also reported technical issues in assessing/improving the fairness of ADM models, which should be considered when designing auditing toolkits. The first issue relates to the already discussed design consideration **DC_no_sensitive_attributes**, as practitioners reported that often access to sensitive attributes is lacking. Another main issue relates to the preferred intervention stage when improving the fairness of a system. Practitioners revealed that when a model appears to be biased, they inspect the training data this model was based on, to think about ways in which collecting more data or pre-processing the data can help in mitigating the bias. Relating this to the design of interactive auditing toolkits, this means that tools should help auditors in inspecting the training data so that they can identify causes of prediction biases and resolve them. We will name this design consideration **DC_identifying_bias_causes**. Further, auditors mentioned that the closer inspection of input data is also important for assessing the quality of the test set that a model is audited on. After all, only if the test set is representative of the data that the model is applied on, the results of the fairness audit can be generalized. We will refer to this design consideration as **DC_fair_testset_design**.

Another design consideration we extracted from their work relates to practitioners' fear that a wide range of biases may creep into a model and identifying all of them is time intensive. Hence, they do not want to waste efforts on identifying occasional "one-off" mistakes from a model but want to prioritize big, systemic biases, that are unlikely due to chance. As we will see in section 4.3 there are ways in which auditing toolkits can meet this requirement (**DC_prioritize_systemic_biases**). Connected to this, practitioners still fear, that they have blind spots in analysing the fairness of a system and that they do not think of all the attributes that can serve as grounds for discrimination. Hence, creating a tool that can automatically suggest possibly discriminated subgroups or individuals, could be a way to accommodate this fear (**DC_account_for_blindspots**, described in further detail in section 4.3.2)

The final paper we are going to discuss was written by Constanza-Chock et al. [6]. They interviewed 10 different auditors of ADM systems, that were either researchers, CEOs of dedicated auditing companies, or leads of internal company teams responsible for bias audits. Their goal was to identify current auditing practices, as well as obstacles on organizational, technical and legal levels these auditors faced.

One interesting discovery was that many auditors currently favour custom-built toolkits over standardized ones. The preference for custom solutions is attributed to the fact that standardized toolkits may not always fit tailored use cases, and

some interviewees expressed concerns about the overemphasis on quantitative measures of fairness that try to express unfairness in a single number, without further consideration of its context or origins. However, despite this preference for customization, we believe there are compelling reasons to enhance standardized toolkits, as they are readily available to the public, more cost-effective, and can be swiftly implemented, unlike developing a new custom tool from scratch. While the study conducted by Constanza-Chock et al. did not directly address how to capitalize on this potential, we identify several ways to overcome the established disadvantages. First, to address the applicability to tailored use cases, toolkits should offer a wide range of functionalities that cater to various scenarios. Additionally, allowing some degree of customizability would be beneficial, enabling users to not only examine pre-defined fairness or performance measures but also define their own metrics (**DC_variability_and_customizability_of_metrics**). Related to the second concern, that tools express the fairness of a system only through quantitative measures, it is essential that tools also encourage deeper analysis of biases: for instance, by letting users inspect the training data behind a model, tools can allow users to reason about the causes of a models' unfairness (**DC_identifying_bias_causes**). Further, by letting users not just inspect demographic subgroups (e.g., based on gender or race) but also subgroups based on other attributes in the data, users are encouraged to contextualize biases better and understand their occurrences (**DC_bias_contextualization**).

Another interesting finding that reveals a design consideration for interactive tools is how auditors currently deal with intersectional bias analysis: Constanza-Chock et al. found that auditors generally have the intent to perform such analyses, but in practice could not provide many cases in which they were conducted. They hypothesised that this was likely due to the general difficulties surrounding such analyses, like dealing with a large number of small subgroups and not being able to identify all marginalized groups. Despite such difficulties, the importance of identifying and understanding intersectional biases is clear, which is why tools should support and facilitate such analysis (**DC_intersectional_analysis**).

3.2 Exploring the potential of tools

The studies discussed in the previous section address auditors' current practices and concerns when assessing the fairness of ADM systems. In this section, we will examine the studies that explore how practitioners think toolkits can help in this assessment. The first of these studies was conducted by Law & Du. They held 10 semi-structured interviews with ML practitioners of the same company, working on different projects. They introduced the practitioners to the case example of bias detection in the *Adult dataset* (the same dataset we have described in section 2 of this paper) and then asked them about their encounters with bias detection in ADM models and how they thought tools could help them. As we already identified as **DC_no_sensitive_attributes**, practitioners reported that not having access to sensitive attributes was a major obstacle for them in auditing a systems' bias and that having tools that can help with that would be highly useful. One concrete suggestion was to make tools that automatically predict the sensitive attributes of data instances, based on other features in the input data. Similarly as found by Holstein et. al, interviewees also expressed their fear of bias

audits becoming unscalable and suggested implementing functionality in tools that allow them to prioritize big, systemic biases (**DC_prioritize_systemic_biases**) and functionality to ensure they do not miss any of them **DC_account_for_blindspots**). Finally, they also mentioned the importance of having tools that allow them to assess the training data, to identify the causes of a model's biases (**DC_identifying_bias_causes**).

Richardson et al. conducted another study investigating toolkits' potential in facilitating algorithmic bias audits. In a usability study, they let 20 ML practitioners test one of two fairness toolkits (Aequitas [20] or Google Fairness Indicators [21]) and let them reflect on the usefulness of these. They used the results of this study to set up a rubric with tool requirements. While this rubric also contains points regarding the tools' usability and tools that could be used for a broad range of ML applications, we will concentrate on the functional requirements related to bias detection in ADM systems. All of these requirements are related to design considerations we already established from previous literature: **DC_variability_and_customizability_of_metrics**, **DC_intersectional_analysis**, **DC_no_sensitive_attributes**, **DC_bias_contextualization** and **DC_identifying_bias_causes**.

Another relevant paper is the work by Lee & Singh, who conducted semi-structured interviews with ML practitioners to review programming libraries like IBM Fairness 360 or Fairlearn [19, 22] that provide pre-defined metrics and algorithms to analyse and mitigate the bias of ADM systems. Based on the interviews, Lee & Singh establish how these libraries could be improved. The first design considerations concern the need to inspect the training data to identify bias causes and find possible proxies for protected attributes (**DC_identifying_bias_causes**, **DC_identifying_proxies**) Second, they were concerned about the customizability of tools to their use cases **DC_variability_and_customizability_of_metrics** and the degree to which they could handle more complex form of bias (e.g., bias based on multiple non-binary sensitive attributes, **DC_intersectional_analysis**). Third, they expressed their interest in tools that highlight the significance of biases, so that they would not waste time inspecting disparities that are due to random chance **DC_prioritize_systemic_biases**.

Even more recently a study was conducted by Nakao et al., who developed a new prototype for an interactive auditing toolkit after they conducted several workshops to identify stakeholders' needs in such tool [16]. They specifically focused on fairness audits in the context of a loan allocation system and therefore interviewed both data scientists and loan officers as potential auditors of this system (note that we do not review their developed prototype as part of our fairness tools since it is not publicly available). In their study they identified the following design considerations: **DC_variability_and_customizability_of_metrics**, **DC_intersectional_analysis**, **DC_identifying_proxies** and **DC_bias_contextualization**.

4 Functional Requirements for Tools

4.1 Functionality for detecting bias in models' predictions

The first main category of requirements regards a tool's functionality of letting users identify bias in a model's predictions. In the following sections, we will cover what forms of biases should be detectable by toolkits (section 4.1.1), how it is important that intersectional bias can be analysed (section 4.1.2) and how tools should offer bias analysis that goes beyond sensitive attributes, in case that there are proxy attributes in a model or attributes that make the treatment of groups with different sensitive attribute values justifiable (section 4.1.3). In each section, we will also discuss to which extent our reviewed tools offer the required functionality.

Table 1: Requirements related to tools' functionality to let auditors find bias in an ADM model's predictions.

Functionality for detecting bias in a model's predictions	
Different forms of bias can be detected [DC_variability_and_customizability_of_metrics]	
Some standard bias measures are supported:	
Outcome based (group)	AE DL FS (RD) (WI)
Actual vs. Outcome based (group)	AE FS FV RD WI
Probability based (group)	FS (RD) (WI)
Similarity based (individual)	FS RD WI
Causal based	/
Tool provides customizable bias metrics	/
Intersectional bias can be explored [DC_intersectional_analysis]	
Bias based on non-binary sensitive attributes	AE FV RD WI
Bias based on multiple sensitive attributes	FV RD (WI)
Prediction bias beyond sensitive attribute(s) [DC_bias_contextualization], [DC_identifying_proxies], [DC_no_sensitive_attributes]	
Tool lets user contextualize differences in outcomes	DL RD WI
Indirect Bias Analysis (**with** access to sensitive attributes)	
Functionality to find proxies	FS RD WI
Functionality to relate proxies to decision attribute	see section 4.1.2
Indirect Bias Analysis (**without** access to sensitive attributes)	
Estimate sensitive attributes from data	/
Functionality to relate (possible) proxies to decision attribute	see section 4.1.2

AE = Aequitas, DL = DiscriLens, FS = FairSight, FV = FairVis, RD = RAI Dashboard and WI = WhatIf Tool

4.1.1 Different forms of bias can be detected

This requirement pertains to the design consideration **DC_variability_and_customizability_of_metrics**, which has been established by reviewing the works of [6, 17]. To reiterate, some practitioners have avoided using toolkits because they do

not provide implementations for all the bias metrics relevant to their decision task. Therefore, an implementable solution is to create toolkits that offer a wide range of standard metrics, that can further be customized to their specific use case.

Some standard bias measures are supported

Because bias is such a complex and non-arbitrary concept, there are various, often incompatible definitions that can be used. While it may be tempting to only choose one of these definitions and assess a system's fairness accordingly, researchers have warned against such simplifications [23, 24]. Take for instance the measure of "Equal Opportunity" in the example of our loan allocation system. According to this definition, a system is free of bias if the "true positive rates" among all groups of interest (e.g. all genders) are equal. The "true positive rate" is defined as the probability that a loan applicant for which an ML model predicted a positive outcome (i.e. being approved for a loan) also has a positive label in the data. While at first sight, this definition may sound "fair" it does not account for the "unfairness" that might be present in the labels, based on which a model's errors are assessed. In the case of the *Adult dataset*, it could be argued that the inequality in high and low incomes between genders reflects the gender pay gap that is a result of direct discrimination as well as unequal (and possibly unfair) societal expectations and opportunities for different genders [25]. Thus, the fact that more men than women have a positive label (i.e. high income) in this data does not mean that this bias should be replicated by an ML model trained on it. Especially, if the labels would be used as a proxy for who deserves a loan and not, it can easily be argued that consistently giving more men than women a loan, would only increase existing gender inequalities. To account for existing biases in the labels, it is possible to choose a "bias-transforming" fairness goal [26] like "Demographic Parity". With this measure, we ensure that an equal portion of men and women are granted a loan by the system. Still, also this measure comes with disadvantages. For instance, it ignores differences in qualifications or eligibility of population groups that could justify a difference in outcome (i.e. loan vs no loan) [23]. One way to address this problem is by focusing on similarity-based fairness measures, that are based on the principle of "treating likes alike": individuals that are similar in terms of their eligibility for a loan should obtain the same outcome [23].

As has become clear from this example, there is a variety of bias measures to take into account when auditing an ML system and there is no single criterion that "makes or breaks" the fairness of a system. Hence, a tool must support a wide range of these metrics so that auditors can choose one or multiple to inspect based on the given use case. To distinguish between the different forms of fairness that should be measurable with a toolkit we will, similarly as [27], make use of the five bias categories specified by [28].

Group-Based Measures. The three definitions falling under the subcategory of *group-based bias measures* measure whether there are substantial differences in treatment between two or more groups (e.g., men vs. women vs. non-binary). This can first be measured by comparing the classifier's outcomes on the groups, second, by comparing the classifier's errors on them, and third by comparing the classifier's predicted probabilities on them. As can be seen in Table 1, most auditing tools support the bias

definitions based on classifiers' errors, with Equal Opportunity (one of the measures discussed previously) being one example of such definition. We have already pointed out how these error-based measures are not appropriate to account for the bias present in the ground-truth labels. Hence, the fact that many tools do not support outcome-based measures (that do account for this bias) poses a serious shortcoming for their applicability. Additionally, the fact that so few tools support probability-based measures, is another drawback. Most ADM models do not directly output binary decision labels for a prediction task, but instead prediction probabilities, which can then be translated to binary labels by applying some decision threshold on them. Yet, this threshold is quite flexible and may even change throughout a system's deployment. For instance, in the case of our loan approval system, this may depend on the bank's resources and the number of loans it can grant [29]. To be able to guarantee the fairness of a system, independent of a chosen decision threshold, it is thus useful to have tools that allow for probability-based bias assessments [29]. One example of such measure is the "Balance for positive class", demanding that for all instances with a positive decision label in the data, the average prediction probability is the same across groups [30] (i.e. the average prediction probability for women with a positive label is the same as the average probability for men with a positive label). Satisfying such a goal gives some guarantee that a model's prediction will still be fair once the decision threshold changes.

Currently, only three tools partly allow for the inspection of probability-based bias measures. FairSight operates on the ranking produced by a classifier, which is obtained when ordering the decision instances according to their prediction probabilities. The tool then prompts the user to specify which top-k instances of this ranking will be granted a loan and calculates various bias metrics based on the protected/unprotected individuals represented in this ranking (see Figure 4 (B)). While this gives some insights into the probability-based fairness of the corresponding model, the tool does not operate directly on prediction probabilities but only on the obtained ranking. The other two tools that partly allow the exploration of probability-based bias measures are the WhatIf tool and the RAI dashboard. Both allow users to visualize the prediction probabilities for different subgroups, as can be seen in Figure 10 (A) for the RAI dashboard and Figure 7 (A) for the WhatIf tool. However, neither of these visualizations is accompanied by formal measures (note, that the same holds for output-based bias measures: both tools allow for visualization of them but do not provide exact measures). Adding formal measures would thus be an easy way to improve the suitability of both tools.

Adding a wider variety of bias measures to a tool like DiscriLens, which has been developed with one specific bias metric in mind (in DiscriLens' case "conditional demographic parity"), will prove a bigger challenge. This highlights the need to take a broad perspective when designing auditing tools and make them flexible for different tasks and fairness notions.

Similarity based Measures. The fourth group of bias definitions, in addition to the three described above, are similarity-based ones, which define bias s on an individual level by comparing the outcome of a data instance with those of similar ones. Currently, only FairSight, the RAI dashboard and the WhatIf tool support users in

exploring these definitions. FairSight does so by enabling the user to conduct a "nearest neighbour" analysis, where the user can select an instance and compare their predicted ranking position to those of similar ones (see Figure 5 (d)). The WhatIf tool and the RAI dashboard on the other hand provide a "What-if" analysis. Here users can change feature values of an instance of choice, and observe how this affects the prediction. To illustrate, look at Figure 7 (B), showing this component of the WhatIf tool. Here a user has selected a female data instance and changed their sex to "male". The outcome of this change is that the probability of granting a loan raises from 0.283 to 0.535. There are two primary reasons why this kind of similarity-based fairness analysis should be an essential component of any auditing tool:

- First, consider our loan approval system and imagine that it satisfies the outcome-based fairness measure demographic parity in regard to gender (i.e. for all gender groups it grants the same proportion of loans). While this system may look fair on the outside, there is no guarantee that the people who get granted a loan also are eligible for one, or that the reasoning behind granting a loan is fair. It may for instance still behave in an individually discriminating way like in the example given above
- Second, in the study by Richardson et al., interviewees mentioned how they found it easier to understand global patterns of discrimination (as given by group-based fairness definitions) when being provided with individual examples of discriminated instances. This finding is also backed up by [31].

Both reasons should provide developers of new tools with motivation to include similarity-based analyses in them. Still, it should be noted that similarity-based measures also come with disadvantages, the biggest one being that it is not clear how to define similarity and how similar two instances should be to receive the same decision outcome. In the case of a loan allocation system, it is e.g. clear that a man and woman who are identical in all features except their sex, should not be treated differently. However, if a man and woman also differ on a relevant attribute (e.g., their current employment status), this is not so arbitrary. On the one hand, a difference in this attribute can justify handing out a loan to one person but not the other. On the other hand, differences in these attributes may reflect systemic and societal gender inequalities (e.g., different lengths of parental leaves, women working more part-time, etc.) that an auditor needs to account for [32]. Hence, for functionalities like the "nearest neighbour analysis" in FairSight, it can be useful if a tool lets auditors define their own similarity metric to allow for such considerations. We will further elaborate on this point in section 4.1.1.

Causal-based Measures. The fifth and last category of bias definitions are causal-based ones [28]. These definitions are the most distinct, as they do not solely define bias on the predictions of a model but also on the causal relationships that are assumed to underlie it. In other words, we use causal-based definitions to examine whether there are discriminatory causal relationships between a sensitive attribute and a decision attribute in a model's decision-making. Currently, no tool allows the user to investigate these causal notions. As causal bias definitions lay in a niche research area within the fairness literature, it may not be surprising that no tools support their exploration. Still, it should be noted that there may be great potential in incorporating them

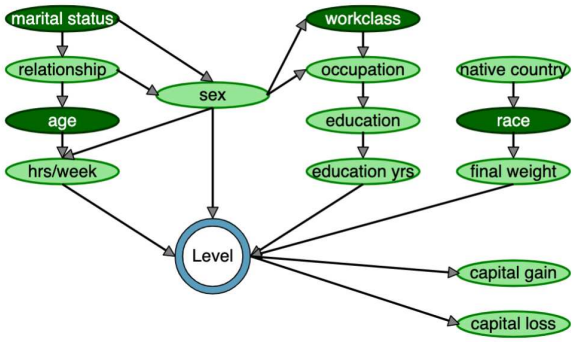


Fig. 12: A graph visualizing the causal relationships within an ADM model trained on the *Adult dataset*. This graph is visualized as a part of a auditing tool developed by [34]. Though this tool is not openly accessible, its design and its use of causal fairness definitions can still serve as an inspiration for other toolkits.

into interactive tools, especially by visualizing the causal relationships within a model through causal networks. The paper by [33] gives a good overview of how causal networks could help in the detection of bias in an ML model. Additionally, [34] and [16] present tools through which causal analyses can be conducted, and also point out the merits of adopting a causal framework. To give a more concrete example, refer to Figure 12 displaying a graph from [34], that visualizes the causal relationships within the *Adult dataset*. In their tool, [34] show this graph to let users reason about problematic relationships between attributes like "sex" or "race" and the decision attribute "Level" (short for "level of income"). In this case, we see that peoples' sex has a direct causal effect on their income, but is also linked to other attributes (e.g., the number of working hours) that may influence income levels. The cited papers give more information on how to quantify these relationships and how auditors could use visualizations like these to reason about the biases within a system. For instance in this case it is clear that utilizing the ADM model based on the causal relationships in 12 is problematic, since in this model there is a direct link between "sex" and "income". If the causal relationships within the model were different, and there was only an indirect link between people's sex and their income level (e.g., explained by the link to different working hours between different sexes), auditors could apply different reasoning as to why this link may or may not be acceptable.

Though the tools by [34] and [16] are not openly accessible, their design may still serve as an inspiration to add further functionality to existing tools.

Customizable metrics

While it is useful if a tool provides a couple of standard bias measures by default, our reviewed interview studies revealed that auditors would also like toolkits in which they can customize their own metrics (see **DC_variability_and_customizability_of_metrics** in section 3) [6, 15, 17]. This also relates to findings in other ML literature, where practitioners explain how they evaluate their products

on organization-defined and product-specific metrics, rather than standard ones [35]. While the customizability of metrics is arguably a broad requirement, we already touched upon some ways in which this requirement could be fulfilled, like allowing users to define a similarity function for similarity-based fairness measures (see section 4.1.1) or allowing them to specify a decision threshold to translate prediction probabilities to prediction labels (see section 4.1.1). Another suggestion that came forth from the interview studies discussed earlier is to let users define metrics that can be used to assess a model's fairness in non-binary prediction tasks, like multi-class problems or regression problems. In the case of a loan approval system, it might, e.g., be of interest not just whether an individual gets granted a loan but also what the height of that loan is, and whether that is equally distributed among demographic groups.

4.1.2 Intersectional bias can be explored

Another design consideration for the development of tools is their functionality to detect intersectional biases (**DC_intersectional_analysis**). "Intersectional bias" is a term that was 1989 coined by Kimberly Crenshaw, to describe the discrimination that black women faced in employment that could neither be fully explained by discrimination against sex, nor discrimination against race [36]. Since then, the term has been used to describe how people who come from a combination of marginalized groups (based on gender, race, religion, class, and other identity markers), face different levels of discrimination than cannot be explained by the "sum" of discrimination faced by each marginalized group in isolation. ADM systems may also behave in intersectionally discriminatory ways, which is why tools should assist in the detection of those.

To facilitate this, there are two functional requirements a tool should fulfil: first, it should allow the analysis of bias based on non-binary sensitive attributes, and second, it should allow the analysis based on combinations of these attributes. Both points are elaborated on in the next paragraphs.

Bias based on non-binary sensitive attribute(s)

The first consideration that needs to be made when conducting an intersectional bias analysis, or even when analysing bias from a single-axis, is which identities to include per sensitive attribute [37]. As sensitive attributes are typically non-binary, auditing toolkits must support this non-binary analysis. Out of our six tools, all do so except DiscriLens and FairSight. The risk of using such simplifications should not be underestimated. Take for instance the attribute "race"; using a tool like DiscriLens or FairSight, we are forced to discretize this feature into two groups, e.g. "white" and "non-white". Yet, for any domain expert using such tool, it is clear that this discretization does not account for all the different types and levels of bias different non-white racial groups may face [12, 37]. Looking for instance at Figure 10 (A), we see the distribution of predicted probabilities for the group of white men, coloured men, and black men. The model predicts higher probabilities of granting a loan for the group of white than coloured men. However, the difference in prediction probabilities (and also False Negative Rates) is even larger, when comparing the group of white and black men. This indicates that within the group of coloured men, black men face especially

averse effects. If an auditor would use a tool that only allows bias detection on binary-sensitive attributes, they would miss this important pattern. Fortunately, it should not be too difficult to allow for non-binary bias analysis in DiscriLens and FairSight. Both tools heavily rely on colours to encode different groups of interest in their data visualization/exploration. Adding more colour options to the tools is one possible way to allow for fairness analysis of non-binary sensitive attributes.

Finally, note that in the question of which categories to include per sensitive attribute, also broader issues need to be addressed, for instance how attributes like race or gender were recorded (i.e. are they self-reported or recorded by the data collectors?). While it is not possible for a tool to address these issues on a technical level, they can still pose serious threats to the fairness of an ADM system and therefore should not be ignored in an audit [37, 38].

Bias based on multiple sensitive attributes

Once it is clear which categories to include for each sensitive attribute, the next step for an intersectional analysis is to decide on the combinations of attributes that need to be inspected. To be able to conduct such analysis with an interactive tool, the tool must support the fairness analysis based on multiple sensitive attributes. Out of all tools, only FairVis and the RAI dashboard fully do so. Indeed, when using this functionality we see that a model trained on the *Adult dataset* also displays signs of intersectional discrimination. Looking at Figure 2 (b) we see the "False Negative Rates" for different subgroups based on people's "sex" and "race". We observe that this rate is already quite high for women, even higher for black people and highest for black women. This knowledge is crucial for a fairness auditor to decide on how to improve an ML system. In this case, an auditor could e.g. recommend that before the system can be deployed more data needs to be gathered for this subgroup. If an auditor would only analyse one sensitive attribute at a time, they might not have found this solution, and might only suggest collecting additional data for women and black people, rather than the intersection of both. Following this example, more tools must allow the analysis of intersectional discrimination. The WhatIf tool already partly supports this feature, but only for subgroup combinations based on two sensitive attributes. Still, the way this tool as well as FairVis and RAI dashboard allow for intersectional bias analysis can serve as inspiration for other tools: the functionality works by letting users generate subgroups of choice (see e.g. Figure 2 (A) for FairVis, Figure 8 (A) for the WhatIf tool and Figure 9 (B) for the RAI dashboard), and then inspect and compare all fairness metrics across all user-generated groups. Similar functionality could be added to other tools.

4.1.3 Prediction bias beyond sensitive attributes

In the previous section, we assumed that in the fairness assessment of an ADM model auditors have access to all relevant sensitive attributes and that they are only interested to observe disparate behaviour of a model based on these attributes. The interview studies revealed, however, that current auditing practices often go beyond the analysis of just sensitive attributes for two reasons: first, it is important to contextualize differences in outcomes between different demographic/sensitive

groups, since a model's decision to treat groups differently may be justifiable (**DC_contextualize_biases**, [6, 16]). Second, discriminatory biases may not always be based on attributes that are legally protected (e.g., gender or race) but on attributes that might serve as a proxy for these (e.g., zipcode for race), a phenomenon known as indirect bias. Functionality to conduct an indirect bias analysis, both in the case in which auditors have access to sensitive attributes and those in which they don't is, therefore, essential in a toolkit as earlier indicated by **DC_identifying_proxies** and **DC_no_sensitive_attributes** [12–15, 17].

Contextualize differences in outcomes

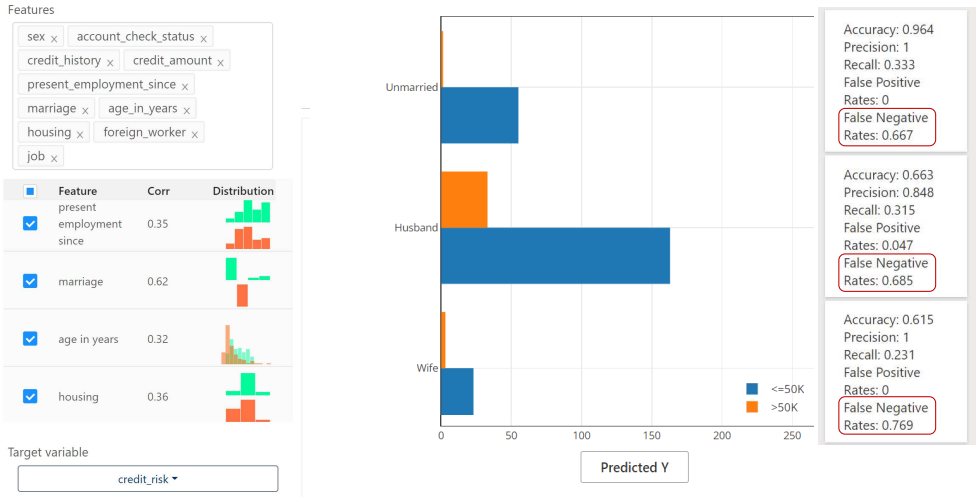
The functionality to contextualize biases is important to understand why an ADM model may make less preferable decisions for some population group over another. In some cases, a difference in treatment may be justifiable by so-called "explainable attributes" [39]. When for instance in our use case a classifier decides that more men than women should receive a loan, this is not necessarily problematic if this can, e.g., be explained by women in the data working in lower job positions than men, indicating that they have less financial means to pay back a loan. DiscriLens is a tool specially developed to contextualize biases and understand if they are explainable. Here users specify a list of explainable attributes, and the tool automatically highlights the cases where a protected and an unprotected group have a high difference in positive decision probability, conditioned on these attributes. In other words, it only displays biases that are not explainable. For instance, in Figure 3 (E), we see that when specifically filtering for people with high education levels and high amount of workinghours, still more men than women get granted a loan. Similarly, the "Model Statistics" component of the RAI dashboard allows for the analysis of non-explainable discrimination, by visualizing the prediction outcomes for the group of highly educated men and women, to see if there are fundamental differences in both (see Figure 10). Note how powerful the "subgroup generation" functionality is in the RAI dashboard, as the same mechanism can be used to study intersectional discrimination (see the previous section). Thus, adding similar functionality to other tools should make them more suitable to auditors' needs.

One final note for the contextualization of bias, is that the choice of "explainable" attributes should always be carefully considered by a domain expert. In the previously mentioned example, of women less likely to receive a loan because of having lower job positions than men, an expert should always consider the question of why this is the case and whether this is the result of historical bias (which in our example might very well be the case, given that women are known to not receive the same job opportunities as men). To make up for this already existing bias, it may not make sense to ignore "explainable" patterns of discrimination, but instead, critically question the extent to which "explainable" discrimination is legitimately explainable [26].

Indirect Bias Analysis (with access to sensitive attributes)

As explained earlier, indirect discrimination in ADMs occurs when a model does not directly make use of sensitive attribute information to derive its decisions, but when it relies on attributes that are proxies for these. One famous example is the practice

of redlining, where a model indirectly disadvantages racial groups, by using the zip code of people as a factor in its decisions. As we have found in our review, auditors are highly aware of the phenomenon of indirect discrimination, which is why they require tools that allow them to analyse it. In the case that they have access to sensitive attributes, like gender or race, an auditing tool can facilitate this analysis by first allowing them to identify proxy attributes and then letting them explore a model's behaviour on these. FairSight helps users in the first step, by providing visualizations of how attributes are differently distributed among sensitive groups, as well as giving a correlation measure between them (see Figure 4 (A) to inspect the component for this tool, and Figure 13a for a specific case example). In Figure 13a we see that the feature "marriage" is considerably differently distributed between the protected and unprotected group, caused by the fact that "Wife" is a feature value that is only applicable to women, while "Husband" is a value only applicable to men. The difference in feature distribution is also indicated by a high correlation measure between the feature "marriage" and sensitive attribute "sex". To allow for the detection of proxy attributes, also the RAI dashboard, the WhatIf tool and FairVis enable users to visualize how attributes are differently distributed for population groups (see Figure 10 (B) for the RAI dashboard, Figure 7 (A) for the WhatIf tool and Figure 2 (A) for FairVis). However, these visualizations are not accompanied by correlational measures. Concerning the second step in the identification of indirect bias, i.e. the exploration of a model's predictions on the different values of this variable, only the RAI dashboard, the WhatIf tool and FairVis allow doing so (this step essentially boils down to inspecting a models' behaviour on a non-binary sensitive attribute (see section 4.1.2)). After e.g. having found that an individual's relationship status is highly correlated to their sex, we could use the RAI dashboard to visualize a model's performance on different population groups determined by this proxy attribute. In Figure 13b we indeed see that the model performs unfairly on the group of "Wives", making more false negative errors for them than other "relationship" groups.



(a) FairSight's feature to detect proxy attributes

(b) The RAI dashboard lets users inspect differences in prediction outcomes for redlining attribute values

Fig. 13: Functionality to (a) find proxy attributes that are highly correlated to sensitive attributes and to (b) detect the relationship between these proxy attributes and a class label

Indirect Bias Analysis (without access to sensitive attributes)

Analysing the occurrence of indirect discrimination is complicated considerably when the training data of a model does not contain any "traditional" sensitive attributes but may contain proxies for these, which in turn cannot be directly identified as such. As we have seen in section 3 this is a very real concern among auditors: often sensitive information is not collected for a decision task (since it may be even illegal to do so), yet without this information, it is hard to identify possible disparate impacts of a model for different sensitive groups [6, 12–14, 17]. In the interview studies by [12, 14] auditors made some suggestions on how this concern could be addressed on a technical level, using interactive tools: they suggest making tools that can estimate sensitive attribute values for each individual based on the rest of their information. To illustrate, some ML practitioners interviewed by [12], already developed systems that use information about peoples' IP addresses (disclosing information about their home location) and names to estimate their sex and ethnicity. Still, they were wary about additional biases introduced by this process and also had concerns about storing (inferred) demographic information and the associated risk of data leakage or misusing this data for secondary purposes [12]. Currently, none of our reviewed tools supports the estimation of sensitive information based on other attributes in the data. However, given the risks of this approach, it is also questionable to which extent this feature is desirable to deal with the analysis of indirect bias.

Other literature on fairness in ADM explores alternative ways to deal with this problem. First, though of less interest in our paper, there are legal regulations that could be enforced, to only allow trusted third parties to access sensitive attributes solely for auditing purposes [40–42].

Another more technical approach for unravelling patterns of indirect discrimination is the exploratory analysis of the ADM model [42]. In a 2017 paper Veale & Binns suggest that exploratory analyses could be used to find interesting patterns in the data, that could afterwards be more closely inspected for possible correlations with sensitive/demographic groups (using additional data sources) [42]. This approach to identifying indirect discrimination was also suggested by Ruggieri et al., who extracted potentially discriminatory association patterns from the data (e.g. *IF "zipcode" = XYZ THEN "no loan"*) to then use additional databases to find whether the premises of these rules (i.e. *"zipcode" = XYZ*) relates to sensitive information of individuals [43]. While additional resources are needed to perform this second step, tools can facilitate the execution of the first step by enabling auditors to analyze performance/fairness measures based on the value/value combinations of other attributes in the data. This requirement was already established in section 4.1.2 and further explored in section 4.1.3, highlighting how important a flexible design of toolkits is and how limiting it is if tools only support the fairness analysis based on one, pre-determined sensitive attribute.

4.2 Functionality for detecting bias in models' input data

The rubric given in Table 1 focuses on a tool's functionality to find biases in the predictions of a model. However, as we have found through our literature review, auditors also find it important to inspect the input data for possible biases for two reasons: first, to understand where biases in a model's predictions may be coming from (**DC_identifying_bias_causes** [6, 12, 14, 15, 17]) and second, to ensure that the fairness audit of the model is conducted on a representative and "fair" test set (**DC_fair_test_set_design** [12]). Lastly, also note that in our paper we focus on assessing the input data as part of the audit *after* model development. Still, the assessment of the training data should be an essential step *before* a model is trained, as basing a model on highly biased data might not be desirable in the first place. Of course, the requirements listed in the upcoming section still hold for tools that would be used for this purpose.

4.2.1 Finding bias causes in training data

In discussing a tool's functionality to find bias causes we will distinguish between the different bias causes established by Suresh & Guttag [44]. Note that many bias causes (e.g., errors in how the data was collected) are not completely identifiable on a technical level and that we will merely focus on those causes whose identification can be facilitated by auditing toolkits.

Table 2: Requirements related to tools' functionality to let auditors find biases in an ADM model's input data.

Functionality for detecting bias in models' input data [DC_identifying_bias_causes], [DC_fair_test_set_design]	
For identification of bias causes in the training data, let user...	
Inspect the relation between attributes and ground truth	FV RD WI
Inspect relation between features	FS FV RD WI
Compare train and test set	/
For identification of biased pattern in test data, let user...	
Inspect subgroup sizes of interest	FV RD (WI)

AE = Aequitas, DL = DiscriLens, FS = FairSight, FV = FairVis, RD = RAI Dashboard and WI = WhatIf Tool

Inspecting the relation between attributes and ground truth

One possible source of bias in ADM models comes from the bias that may be present in the ground truth that the model is trained on, that the decisions that were made historically for the decision subjects (e.g., high vs. low income or loan vs. no loan) in the training data. The ground truth can be subject to *historical bias* or *measurement bias* citeSuresh2019AFF. The former occurs when the label has been recorded correctly, but contains patterns of historical inequalities between population groups (e.g., women being recorded to have lower income than men). The latter (i.e. measurement bias) occurs when due to errors or biases in the decision process, individuals did not get the label they were eligible for (e.g., women who are not granted a loan, even though they would have paid it back if given the opportunity). While it may not be possible to distinguish between these forms of bias in the labels, it is still crucial that tools allow auditors to inspect the labels, to understand if they are favoured more towards some groups than others. For this, tools should support all purely label-based fairness measures, as discussed in section 4.1.1 (i.e. the measures that are not based on prediction errors/prediction probabilities) on the model's ground truth: outcome-based measures, like demographic parity, similarity-based measures and causal measures. When tools provide these measures, the same requirements hold as discussed in section 4.1.2 and 4.1.3: the measures should be applicable on intersectional groups and the bias measures should go beyond sensitive attributes, to contextualize biases and understand patterns of indirect discrimination. Currently, FairVis, the RAI dashboard and the WhatIf tool allow for partial analysis of bias in the ground truth labels. In Figure 14, an example is shown where predictions of an ADM model are biased against black women (high False Negative Rates) and favoured towards white men (higher False Positive Rates). When we inspect the ground truth label balance, we find a reason for why the ADM model is more biased towards predicting positive labels for white men: they have more than double the ratio of positive labels in the ground truth than black women. Though this information is useful to understand where the model's bias comes from, it would (among others) be useful if FairVis would let users contextualize this bias, to understand whether the difference in ground truth label balance is "explainable" by other attributes. Note, how the functionality to contextualize biases

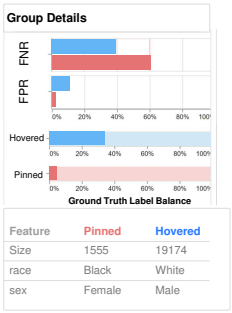


Fig. 14: After selecting different subgroups of interest, users can inspect the models' performance on these groups as well as their label balance in the ground truth. This can help in identifying possible bias causes of a model.

is lacking both in FairVis' functionality to assess fairness in predictions as well as fairness in ground truth labels. Similarly, other tools suffer from the same limitations in their functionality for ground truth label analysis as they do in their functionality for prediction analysis. For instance, the WhatIf tool does not allow intersectional analysis based on more than two sensitive attributes (see section 4.1.2) and the RAI dashboard only gives visualizations (and no quantitative measures) on the ratio of positive/negative labels in the ground truth (see section 4.1.1). Extending the tools' functionality in these regards will thus be a way to accommodate for the design considerations addressed in Table 1 as well as Table 2.

Inspect relation between features

Historical bias may not only be present in the ground truth labels but also the data itself [44]. Take for instance the *Adult dataset*, which may not only be unfair in terms of the unequal distributions of high/low income but also in terms of other features. To observe these inequalities, tools must enable auditors to inspect the relation between different features. Currently, this is already possible, using FairSight, the RAI dashboard and the WhatIf tool. Both in the RAI dashboard and the WhatIf tool, this functionality is provided by a visualization interface, where users can specify which variables should be plotted on the axes of a two-dimensional graph, along with other options for colouring or labelling data attributes according to the dataset's features (see Figure 10 (C) for the RAI dashboard and Figure 7 (A) for the WhatIf tool). Looking for instance at Figure 10 (C) we see interesting patterns in the education level of racial groups, and that people with an Asian, Asian American or Pacific Islander ancestry have higher education levels than black people in this dataset. As mentioned in section 4.1.3 this could be seen as a ground for explainable discrimination. In other words, the fact that black people have lower education levels explains their lower income and could consequentially justify a bank giving out fewer loans to them. However, as also mentioned in this section, an essential part of a bias audit is questioning where these inequalities in input data come from and whether they reveal patterns of historical bias, that lead to unfairness in an ML model [26, 45]. In our example,

the differences in education levels could partly be due to differences in educational opportunities for people of different races, caused by unequal funding for schools and overall differences in access to resources (e.g., private tutors or high-quality books) [46]. As this reflects a larger pattern of systemic racism in the US, this can hardly be seen as fair and an auditor might wish to make up for this unfairness, by choosing bias-transforming measures (as mentioned in section 4.1.1) as their fairness-goal. Thus, having tools that supports the analysis of input data, are needed to identify historical bias as a cause of bias in an ML model, as well as to help auditors make well-informed choices about the fairness requirements of a model.

As mentioned before, FairSight also supports the (visual) analysis of the input data, which is done in two ways: first, per feature two histograms are provided to show how the distribution of this feature differs on the protected and unprotected group (in this case women vs. men, see Figure 4 (A) and 13a). This functionality can be useful to detect patterns like the one mentioned above, but in its current form works only for one binary sensitive attribute, which limits its usefulness. Additionally, FairSight provides a two-dimensional graph where a dimensionality-reduced version of the input data is visualized, and the protected and unprotected datapoints are colour-encoded (see Figure 5 (A)). While this can help in understanding how distinct both groups are overall, it is hard to understand where potential differences come from and whether they might indicate problematic historical biases.

Compare train and test set

Another possible cause of bias in ADM systems is when the data a model is trained on, is not representative of the data it is applied on [35]. To give an example of this *representation bias* in the case of our loan application system: imagine a bank using relatively old data to train their ML model, where some population groups like non-male people are less represented than they are at the time of model deployment. Since the model does not see all population groups equally at training time, it will likely not perform accurately/fairly on the underrepresented groups once it is deployed. To establish representation bias as the cause of a model's unfairness, an auditing tool must let users compare the distribution of train- and test set. For this, tools must make a clear distinction between both so either can be individually inspected and then compared. Currently, none of our reviewed tools supports this functionality. All tools require the user to upload one dataset, along with its ground truth labels and the corresponding model's predictions. Users must choose whether this dataset is the same as the one the model has been trained on or is a separate test set. Since representation bias is a common cause of bias in ADM systems, this is a serious shortcoming in letting auditors identify this as a bias cause.

4.2.2 Inspecting subgroup sizes of interest

As we have touched upon in the previous section, a crucial part of the fairness audit of an ADM system, is ensuring that the audit is conducted on a representative test set [12, 35, 47]. After all, if a test set does not contain all the groups that an ADM system will be applied on, it is impossible to estimate whether the system will behave fairly on those groups. The way in which auditing toolkits can help in crafting representative

test sets is by giving clear indications of the size of different subgroups in the data. We already see an implementation of this in FairVis, where in Figure 14 we see (along with some performance measures) the number of people represented in selected subgroups. Though this is already useful, it could be even more useful if the tool would allow users to order subgroups according to their size, just like it is already possible to order them according to performance/fairness measures. Also, in the WhatIf tool and the RAI dashboard it is possible to observe the group sizes of selected subgroups. However, in the WhatIf tool this is only possible for subgroups based on two sensitive attributes (see Figure 8 (b)) and in the RAI dashboard it is only possible to observe the subgroup size for one group at a time (see Figure 9 (c)). Functionality for an easier inspection of subgroup sizes would be useful.

4.3 Functionality to make bias detection scalable

The requirements introduced in section 4.1 and 4.2 relate to the tools' functionality to audit an ADM system's predictions and input data for bias. The requirements introduced in this section focus on making sure that this audit is scalable. On the one hand this refers to design consideration **DC_prioritize_systemic_biases** [12, 14, 15], in that auditors do not want to inspect errors of an ADM system that are due to chance, rather than reflective of systemic bias issues. On the other hand, auditors fear that in focusing on only the big "obvious" biases, they might miss important blindspots; something that should be accounted for according to **DC_account_for_blindspots** [12, 14].

Table 3: Requirements to make a bias audit scalable.

Tool makes bias detection scalable	
Tool let auditors narrow down the biases they need to inspect [DC_prioritize_systemic_biases]	
Report Confidence Intervals	/
Group similar subgroups together	DL RD
Group similar individuals together	FS
Tool let auditors narrow down the biases they need to inspect [DC_account_for_blindspots]	
(Sub)group biases	FV RD
Individual biases	FS

AE = Aequitas, DL = DiscriLens, FS = FairSight, FV = FairVis, RD = RAI Dashboard and WI = WhatIf Tool

4.3.1 Tool let auditors narrow down the biases they need to inspect

The first way to make bias detection more scalable is by providing tools that can narrow down all the biases auditors need to inspect. In this section we explore how this can be accomplished.

Report confidence intervals

An ADM model is unlikely to yield the same performance and the same positive decision ratio among all groups of interest. Hence, an important question in the audit of an ADM system is which disparities are due to chance and which ones reflect systemic issues. Hence, interviewees in the studies of [12] and [15] expressed their interest in tools that let them explore the statistical significance of subgroup biases, by reporting the confidence interval of bias measures. Currently, none of our reviewed tools supports this feature but some literature on subgroup fairness in ADM gives insight into how it can be implemented: Wang et al. apply the same model on five different test sets to calculate the confidence interval of the resulting bias metrics [37]. Similarly, Friedler et al. have studied how significant biases are, by calculating and comparing the bias metrics over multiple train-test-splits of a model [48]. Likewise, an interactive fairness tool could take a model, a train and a test set as input to then automatically divide the test set into different splits and calculate the confidence interval of the model's fairness measures over them. This would require more effort from the tool developers' side, since this tool needs to access more than just a simple CSV-file of the test data, but also the model itself. Alternatively, users themselves could provide the models' results on different test sets, over which a tool could (without needing access to the model) calculate the confidence interval. This approach would require more effort from the users' side in setting up a file containing all the necessary data. In choosing which option is more viable for an auditing toolkit, it is important to consider the current workflow of ADM model builders and auditors. Hence, before the feature of reporting confidence intervals is implemented in a tool, more conversations with practitioners would be needed to understand how they currently set up their model evaluation and how an auditing toolkit could account for that.

Group similar subgroups together

Though not directly suggested by possible auditors, but already implemented in some tools, another way to reduce the number of subgroups an auditor needs to inspect, is to (automatically or manually) group similar subgroups together. The RAI dashboard already allows one to do so: instead of e.g. generating one subgroup of 50-year-old men, and another of 51-year-old men, users can generate a subgroup of men with a certain age range (e.g. 50-55) and inspect a model's performance on it. Similarly, when working with categorical features, users can group people with similar feature values. In the *Adult dataset*, there is for instance a variable "workclass" with values like "Federal Government" "State Government" or "Local Government". Instead of inspecting each subgroup individually, users of the RAI dashboard can generate one subgroup of all people working for the government (see Figure 9 (b)) for the dashboard's subgroup generation component). Similar subgroups are also automatically grouped together in DiscriLens: in Figure 3 (e), the group of people with more than 65 workinghours per week is suggested as a discriminatory itemset, allowing an auditor to inspect a bigger group of people than when only looking at the group of people with exactly 65 workinghours per week. Grouping similar subgroups together is currently not possible in the other tools supporting the analysis of subgroup biases (i.e. FairVis and the

WhatIf tool). Hence, implementing this feature could help in making the bias analysis more scalable.

Group similar individuals together

When it comes to individual biases in an ML system, it is even more unfeasible for an auditor to inspect all of them, since individual biases only affect one data instance at a time. A way in which tools can help is to group similar instances facing discrimination together. FairSight is the only tool that currently does so, in its Feature Inspector tab (see Figure 6). Here, each feature is visualized in a histogram, and data points that face high levels of individual discrimination are marked in red, to observe their value for the given feature. To illustrate, in the histogram showing the data distribution on the feature "sex", we see that on this feature most individually discriminated instances have the value "female". Auditors can use this histogram to study patterns of individual discrimination in a quick and scalable way. Another way in which the inspection of individual biases could be made more scalable is by applying a clustering algorithm on instances that score high on an individual discrimination score. The auditor could inspect the resulting clusters to find common patterns of individual bias. A similar approach was once adopted by Luong et al., who first identified individually discriminated instances, and then derived decision rules to learn what sets them apart from other instances [49]. Linking individual examples of discrimination to more general "discrimination rules", can also be a promising way to facilitate auditors' understanding of global discrimination patterns in the model [27]. Hence, this kind of functionality could be added to tools that allow the inspection of individual discrimination (i.e. RAI dashboard and the WhatIf tool), but where this inspection needs to be done "one at a time".

4.3.2 Tool automatically highlights most important biases

Auditors are concerned that in efforts to make a bias audit scalable, they will miss hidden but important patterns of bias (**DC_account_for_blindspots**) [12, 14]. Generally, they know that by involving stakeholders in the auditing process, as well as having diverse development teams they are more likely to identify the different population groups that may be subject to bias [12]. Still, they also find it useful when a tool can automatically suggest patterns of bias that would otherwise go unnoticed. In this section, we discuss ways in which both group and individual biases can be automatically highlighted by our toolkits.

Group Biases

Currently, only FairVis and the RAI dashboard automatically suggest subgroups that may be affected by bias (see Figure 2 (E) and Figure 15, for a close-up of this feature). We see that the user has previously generated subgroups based on combinations of people's sex and race, and we already see the false negative rates for these groups, including, e.g., the groups of black women and white women. In the "suggested subgroups" tab we see other potential subgroups of interest, sorted according to their False Negative Rates. In this case, the group of divorced black women from the United

Fig. 15: FairVis is one of the tools that automatically suggest potentially discriminated subgroups to its users. In this case it suggests the group of divorced black women from the US as well as the group of unmarried white women working in the private sector as two groups with high False Negative Rates.

States, as well as the group of unmarried white women working in the private sector are suggested. When clicking on these suggestions, we indeed see that they have considerably high False Negative Rates, also compared to their supergroups of black women and white women.

Similar suggestions are also given in the RAI dashboard, as part of the "Error Analysis" component (Figure 9 (C). In this case, the user has selected to specifically look at the subgroup of white men. Within this group, they want to inspect for which subgroups the recall is especially high or low. On top of the error tree, we can see that the overall recall for the group of white men is 0.29. When following one of the paths of the tree we see that this score is much lower (0.0) for the group of white men older than 31 and with an education level below 7 (where an education level of 7 describes people who have followed education until 11th grade of high school).

The suggestions of FairVis and the RAI dashboard can help auditors in detecting otherwise missed patterns of subgroup bias, but also contextualizing these biases. Thus either of these features can be a useful addition to other auditing toolkits. Still, it should be noted that their current implementation of subgroup bias suggestion may not be ideal. First, both tools only show the performance measure, without giving absolute measures about how many people are affected by unfair treatment. Consider, for instance, the group of divorced black women from the US, that is suggested by FairVis. Inspecting the corresponding data and the ADM model's predictions on it, we found that out of a total of 32k dataset instances this subgroup consists of 300 people. Out of these, only 22 have a positive label (i.e. high income) in the data. Since the model only correctly predicts this label for 3 people, the False Negative Rate is

so high for this group. While this is certainly a problematic pattern, it only concerns a very small part of the data, which the tool does not make apparent. It is then also hard to estimate whether this bias is significant for the group of black divorced women from the US, or whether it is a pattern coming forth from the general bias against black women (independent of their marital status and nationality).

The second concern about how discriminated subgroups are automatically suggested in the RAI dashboard and FairVis, is that both tools only suggest subgroups based on error-based bias measures. To improve this functionality, it is important that other forms of bias, e.g. defined by outcome-based measures (see section 4.1.1), are also automatically highlighted.

Individual Biases

As mentioned in 4.1.1, the only tools that support the detection of individual bias are FairSight, the WhatIf Tool, and the RAI dashboard. Out of all, only FairSight automatically highlights some cases of individual bias. This is done through the distortion matrix (see Figure 5 (D)). To reiterate, the distortion between two pairs of individuals is high when they are close on the input space (i.e. they are similar in terms of their features), but distinct on the output space. Instance pairs with high distortion are highlighted with different colour saturation than instance pairs with low distortion. Additionally, instance pairs that differ on the sensitive attribute "sex", are coloured differently than instance pairs with the same sex. Hence, to find potentially discriminated instances, users could look for female-male instance pairs with high distortion (as their difference in output might only be explained by sexist biases, as the instances are otherwise close in input space).

No clear cases of individual discrimination are highlighted in the WhatIf tool and the RAI dashboard. One possibility to add this functionality is by using the tool's "what if" analysis and automatically highlighting cases that experience a change in their prediction outcome if the value of a sensitive attribute or a redlining attribute is changed.

5 Conclusion & Future Research

In this paper, we have presented an overview of the functional requirements that users have for auditing toolkits. We evaluated six available toolkits according to these requirements and identified realistic ways to overcome their shortcomings. One of the most common shortcomings is their lack of flexibility: many tools assume that there is only one binary sensitive attribute that auditors need to assess for possible biases, and that information on this attribute is always available. To address this issue, tools must be developed that make less rigid assumptions about the availability and number of sensitive attributes. Such toolkits allow for audits where discriminatory bias occurs solely on the basis of proxy attributes and where bias may be of intersectional nature. Other important design requirements include the tools' functionality to assess the training data for possible bias causes and the extent to which they make audits scalable.

While our requirement checklist and the tool evaluation can already guide developers in creating better and more suitable tools, some aspects still should be studied to unlock their full potential.

Integration in workflow

The tools that we reviewed differ in the way they need to be set up. Some are webtools that take CSV files of the data and models' predictions as input [8, 20] others are evoked through python libraries [9, 11]. Whether practitioners choose to use toolkits in practice will depend on the ease with which they can be integrated into their workflow. The ADM developers that were interviewed by Lee & Singh, for instance, preferred tools that can be evoked in Python and that integrate well with other python libraries like pandas or sklearn [15]. Additionally, they had privacy concerns about using web-based tools, that require them to upload sensitive data on external sites [15]. The preferred way of setting up tools likely also varies, depending on the technical skills of an auditor and whether they are involved in system development or not [50]. Hence, developers should spend serious effort on designing tools that easily integrate into different types of workflows and that can run on local machines to minimize privacy concerns.

The usability of the tools

Auditing toolkits should offer the right functionality, but at the same time, this functionality should be easy to use. To determine the current usability of tools, more studies are needed. Some tool developers, already conducted (small) usability studies as part of their research papers [7, 10, 11]. However, all these studies suffer from some disadvantages like only testing the tools on university students with Computer Science backgrounds and only giving the participants tasks that are specific to the exact purpose of each tool. To illustrate, one of the tasks of the usability study of FairSight was *"Can you quantify the degree of fairness in the ranking outcome?"* [7], which requires participants to locate one specific fairness metric reported in the tool. While it makes sense to study the usability of specific tool's components, it is worthwhile to give users more general tasks, that provide a better understanding of how each tool would be used "in the wild". Inspiration for task set-ups could be taken from literature on the interpretability of explainable AI (xAI). For instance, Kaur et al. purposefully manipulated the predictions of a Machine Learning model and studied how well explanation methods could help participants in identifying these undesirable patterns. A similar approach could be taken for testing the usability of bias auditing tools, to see whether the tools help in finding unfair patterns in a model (as well as contextualizing these patterns and identifying their causes) [51].

Need for clear legislation

We have already mentioned how auditing toolkits are highly relevant in light of upcoming legislation like Local Law 144 and the EU AI Act. To shortly summarize: the EU AI Act calls for human oversight of decision-making systems to evaluate risks concerning health, safety, and fundamental rights. Though non-discrimination is considered a fundamental right, the act does not provide specific guidelines on how to ensure that

Sex Categories				
	# of Applicants	# of Selected	Selection Rate	Impact Ratio
Male	1390	667	48%	1.00
Female	1181	555	47%	0.979

Table 4: An example of the type of report that is mandated by the Local Law 144.

a system does not violate it [4]. New York's Local Law 144 is a bit more precise in the requirements it imposes: it mandates that any algorithm used for hiring decisions must undergo an audit by an independent third party, with the results made publicly available. The audit should assess the algorithm's output for potential discrimination based on sex, race and their intersection and must report some basic measurements regarding the corresponding population groups. To specify, it needs to report the number of job applicants for each group, their selection rate (i.e., the percentage of applicants that are selected to move forward in the hiring process) and their impact ratio. The impact ratio is calculated as the selection rate for the group, divided by the selection rate of the highest selected group. An example of such a report as mandated by the law is given in Table, where the different measures among sexes are given 4 [3].

While Local Law 144 is more specific than the EU AI Act, it still lacks (similarly to the EU AI act) specific standards for how the success or failure of an audit is determined. Additionally, neither of the new legislation gives further information on how possible signs of discrimination should be further inspected, e.g. by contextualizing differences in selection rates or by trying to find their cause in the training data. On the one hand, not having clear definitions of when an audit passes or fails helps in accommodating many use cases, where the most sensible bias measure differs depending on the decision task (see section 4.1). On the other hand, researchers and auditors have warned that the lack of more elaborate standards creates a risk of companies conducting minimal, superficial audits, that merely fulfil the regulatory requirement without genuinely addressing bias issues [6]. As hopefully, more rigid legislation will arise, the requirements in auditing toolkits will evolve accordingly. Until then, we also believe that the currently available toolkits and the requirements auditors have in them can shape the new rules that should be set into place. Some of the requirements highlighted in our paper, such as tools' functionality to inspect the training data for different forms of bias and the ability to contextualize biases, are related to essential components of the overall auditing process. Hence, our study and other relevant research contributions can serve as a basis for determining the best practices for audits, which could in turn be incorporated into new laws.

Acknowledgments. This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme.

Appendix A Overview Tools

A.1 Aequitas

A.1.1 Advantages

- supports a variety of "outcome" and "outcome vs. prediction" based fairness measures
- supports non-binary sensitive attributes

A.1.2 Disadvantages

- lacks functionality for detecting probability-, individual- and causal-based fairness measures
- does not provide options for intersectional analysis
- assumes that auditors have access to the sensitive attributes (does not provide functionality to detect proxies)
- no functionality to inspect the training data (e.g., training features and their relation to the input labels)
- no functionality to filter out insignificant biases
- does not automatically suggest possibly discriminated subgroups

A.2 DiscriLens

A.2.1 Advantages

- supports outcome-based measures of fairness
- lets users contextualize the differences in outcomes between groups (i.e.. let them differentiate between "explainable" and "non-explainable" discrimination)

A.2.2 Disadvantages

- does neither provide other group-based fairness measures (i.e., "outcome vs. actual" and "probability" based measures), nor similarity- or causal-based measures
- only supports binary-sensitive attributes
- does not provide options for intersectional analysis
- assumes that auditors have access to the sensitive attributes (does not provide functionality to detect proxies)
- no functionality to inspect the training data (e.g., training features and their relation to the input labels)
- no functionality to filter out insignificant biases
- does not automatically suggest possibly discriminated subgroups

A.3 FairSight

A.3.1 Advantages

- supports analysis of all group-based fairness measures ("outcome", "outcome vs. prediction" and "probability" based)
- supports similarity-based fairness measures
- has functionality to find proxy variables (if sensitive attributes are available)
- lets users visualize relations between sensitive attributes and other features, as well as dimensionality-reduced version of training data (both can help for finding bias in features)
- makes individual bias detection more scalable by: automatically highlighting strong cases of individual bias & grouping similar individually discriminated instances together

A.3.2 Disadvantages

- only supports binary sensitive attributes
- does not provide options for intersectional analysis
- does not let user contextualize differences in outcome and differentiate betweeen "explainable" and "non-explainable" discrimination
- does not let the user inspect all aspects of the training data (e.g. sample group sizes, relations between features and ground truth)
- does not let the user differentiate between train- and test set
- does not provide functionality to filter out (or automatically detect) significant group biases

A.4 FairVis

A.4.1 Advantages

- provides a variety of "actual vs. outcome" based fairness measures
- fully supports intersectional fairness analysis
- lets users inspect different aspects of the training data (e.g., subgroup sample sizes, positive label ratios per subgroup)
- automatically suggest possibly discriminated subgroups

A.4.2 Disadvantages

- does not support any other group-based measures of fairness (i.e. outcome- and probability based)
- does not support similarity- and causal-based analysis
- does not let users contextualize differences in outcome to differentiate between "explainable" and "non-explainable" discrimination
- does not report significance of biases
- does not let the user differentiate between train- and test set

A.5 Responsible AI Dashboard

A.5.1 Advantages

- supports analysis of all group-based fairness measures ("outcome", "outcome vs. prediction" and "probability" based)
- supports similarity-based fairness measures
- fully supports intersectional fairness analysis
- lets users contextualize the differences in outcomes between groups (i.e.. let them differentiate between "explainable" and "non-explainable" discrimination)
- has functionality to find proxy variables (if sensitive attributes are available)
- lets user visualize different aspects of the training data: relation between features and relation between features and ground truth label,
- lets users inspect subgroup sizes within training data
- makes it easier to focus on systematic patterns of subgroup biases, by grouping similar subgroups together
- automatically suggests subgroups that may be subject to discrimination

A.5.2 Disadvantages

- some of the group-based fairness measures can only be conducted visually (and not numerically)
- does not support causal-based analysis
- does not let the user differentiate between train- and test set
- no functionality to make detection of individual biases more scalable

A.6 WhatIf Tool

A.6.1 Advantages

- supports analysis of all group-based fairness measures ("outcome", "outcome vs. prediction" and "probability" based)
- supports similarity-based fairness measures
- supports intersectional fairness analysis for up to two features
- lets users contextualize the differences in outcomes between groups (i.e.. let them differentiate between "explainable" and "non-explainable" discrimination)
- has functionality to find proxy variables (if sensitive attributes are available)
- lets user visualize different aspects of the training data: relation between features and relation between features and ground truth label,
- lets users (visually) inspect subgroup sizes within training data

A.6.2 Disadvantages

- some of the group-based fairness measures can only be conducted visually (and not numerically)
- does not support causal-based analysis
- does not support intersectional fairness analysis for subgroups based on more than two sensitive attributes

- does not let the user differentiate between train- and test set
- no functionality to focus only on significant patterns of subgroup/individual bias
- no functionality to automatically highlight patterns of subgroup/individual bias

References

[1] Ntoutsi, E., Fafalios, P., Gadiraju, U., Iosifidis, V., Nejdl, W., Vidal, M.-E., Ruggieri, S., Turini, F., Papadopoulos, S., Krasanakis, E., *et al.*: Bias in data-driven artificial intelligence systems—an introductory survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **10**(3), 1356 (2020)

[2] Larson, J., Mattu, S., Kirchner, L., Angwin, J.: How We Analyzed the COMPAS Recidivism Algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm (2016)

[3] Automated Employment Decision Tools: Automated Employment Decision Tools. https://rules.cityofnewyork.us/rule/automated-employment-decision-tools-updated/ (2023)

[4] European Commission: Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1623335154975&uri=CELEX%3A52021PC0206 (2021)

[5] Rood, J.: NYC Local Law 144- Brief Overview. https://proceptual.com/2023/01/23/quick-takeaways-from-the-dcwp-rules-hearing-on-aedts-nyc-local-law-144/ (2023)

[6] Costanza-Chock, S., Raji, I.D., Buolamwini, J.: Who audits the auditors? recommendations from a field scan of the algorithmic auditing ecosystem. In: 2022 ACM Conference on Fairness, Accountability, and Transparency, pp. 1571–1583 (2022)

[7] Ahn, Y., Lin, Y.-R.: Fairsight: Visual analytics for fairness in decision making. IEEE transactions on visualization and computer graphics **26**(1), 1086–1095 (2019)

[8] Cabrera, Á.A., Epperson, W., Hohman, F., Kahng, M., Morgenstern, J., Chau, D.H.: Fairvis: Visual analytics for discovering intersectional bias in machine learning. In: 2019 IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 46–56 (2019). IEEE

[9] Sameki, M.: Responsible AI dashboard: A one-stop shop for operationalizing Responsible AI in practice. https://techcommunity.microsoft.com/t5/azure-ai-blog/responsible-ai-dashboard-a-one-stop-shop-for-operationalizing/ba-p/3030944 (2021)

[10] Wang, Q., Xu, Z., Chen, Z., Wang, Y., Liu, S., Qu, H.: Visual analysis of discrimination in machine learning. IEEE Transactions on Visualization and Computer Graphics **27**(2), 1470–1480 (2020)

[11] Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F., Wilson, J.: The what-if tool: Interactive probing of machine learning models. IEEE transactions on visualization and computer graphics **26**(1), 56–65 (2019)

[12] Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudik, M., Wallach, H.: Improving fairness in machine learning systems: What do industry practitioners need? In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp. 1–16 (2019)

[13] Veale, M., Van Kleek, M., Binns, R.: Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In: Proceedings of the 2018 Chi Conference on Human Factors in Computing Systems, pp. 1–14 (2018)

[14] Law, P.-M., Malik, S., Du, F., Sinha, M.: Designing tools for semi-automated detection of machine learning biases: An interview study. arXiv preprint arXiv:2003.07680 (2020)

[15] Lee, M.S.A., Singh, J.: The landscape and gaps in open source fairness toolkits. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2021)

[16] Nakao, Y., Strappelli, L., Stumpf, S., Naseer, A., Regoli, D., Gamba, G.D.: Towards responsible ai: A design space exploration of human-centered artificial intelligence user interfaces to investigate fairness. International Journal of Human–Computer Interaction, 1–27 (2022)

[17] Richardson, B., Gilbert, J.E.: A framework for fairness: A systematic review of existing fair ai solutions. arXiv preprint arXiv:2112.05700 (2021)

[18] Bellamy, R.K., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilović, A., *et al.*: Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. IBM Journal of Research and Development **63**(4/5), 4–1 (2019)

[19] Microsoft and Contributors: Fairlearn. https://fairlearn.org/ (2019)

[20] Saleiro, P., Kuester, B., Hinkson, L., London, J., Stevens, A., Anisfeld, A., Rodolfa, K.T., Ghani, R.: Aequitas: A bias and fairness audit toolkit. arXiv preprint arXiv:1811.05577 (2018)

[21] Tensorflow: Tensorflow Fairness Indicators. https://github.com/tensorflow/fairness-indicators (2020)

[22] Bellamy, R.K., Dey, K., Hind, M., Hoffman, S.C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilović, A., *et al.*: Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. IBM Journal of Research and Development **63**(4/5), 4–1 (2019)

[23] Barocas, S., Hardt, M., Narayanan, A.: Fairness in machine learning. Nips tutorial **1**, 2 (2017)

[24] Selbst, A.D., Boyd, D., Friedler, S.A., Venkatasubramanian, S., Vertesi, J.: Fairness and abstraction in sociotechnical systems. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 59–68 (2019)

[25] Chamberlain, A.: Demystifying the Gender Pay Gap: Evidence from Glassdoor Salary Data. https://www.classlawgroup.com/wp-content/uploads/2016/11/glassdoor-gender-pay-gap-study.pdf (2016)

[26] Wachter, S., Mittelstadt, B., Russell, C.: Bias preservation in machine learning: the legality of fairness metrics under eu non-discrimination law. W. Va. L. Rev. **123**, 735 (2020)

[27] Richardson, B., Garcia-Gathright, J., Way, S.F., Thom, J., Cramer, H.: Towards fairness in practice: A practitioner-oriented rubric for evaluating fair ml toolkits. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2021)

[28] Verma, S., Rubin, J.: Fairness definitions explained. In: 2018 Ieee/acm International Workshop on Software Fairness (fairware), pp. 1–7 (2018). IEEE

[29] Kallus, N., Zhou, A.: The fairness of risk scores beyond classification: Bipartite ranking and the xauc metric. Advances in neural information processing systems **32** (2019)

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

[31] Dodge, J., Liao, Q.V., Zhang, Y., Bellamy, R.K., Dugan, C.: Explaining models: an empirical study of how explanations impact fairness judgment. In: Proceedings of the 24th International Conference on Intelligent User Interfaces, pp. 275–285 (2019)

[32] Hu, L., Kohler-Hausmann, I.: What's sex got to do with fair machine learning? arXiv preprint arXiv:2006.01770 (2020)

[33] Chiappa, S., Isaac, W.S.: A causal bayesian networks viewpoint on fairness. In: IFIP International Summer School on Privacy and Identity Management, pp. 3–20 (2018). Springer

[34] Yan, J.N., Gu, Z., Lin, H., Rzeszotarski, J.M.: Silva: Interactively assessing machine learning fairness using causality. In: Proceedings of the 2020 Chi Conference on Human Factors in Computing Systems, pp. 1–13 (2020)

[35] Shankar, S., Garcia, R., Hellerstein, J.M., Parameswaran, A.G.: Operationalizing machine learning: An interview study. arXiv preprint arXiv:2209.09125 (2022)

[36] Crenshaw, K.: Demarginalizing the intersection of race and sex: A black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics. In: University of Chicago Legal Forum: Vol. 1989, (1989)

[37] Wang, A., Ramaswamy, V.V., Russakovsky, O.: Towards intersectionality in machine learning: Including more identities, handling underrepresentation, and performing evaluation. In: 2022 ACM Conference on Fairness, Accountability, and Transparency, pp. 336–349 (2022)

[38] Ding, F., Hardt, M., Miller, J., Schmidt, L.: Retiring adult: New datasets for fair machine learning. Advances in neural information processing systems **34**, 6478–6490 (2021)

[39] Kamiran, F., Žliobaitė, I., Calders, T.: Quantifying explainable discrimination and removing illegal discrimination in automated decision making. Knowledge and information systems **35**(3), 613–644 (2013)

[40] Kilbertus, N., Gascón, A., Kusner, M., Veale, M., Gummadi, K., Weller, A.: Blind justice: Fairness with encrypted sensitive attributes. In: International Conference on Machine Learning, pp. 2630–2639 (2018). PMLR

[41] Bekkum, M., Borgesius, F.Z.: Using sensitive data to prevent discrimination by artificial intelligence: Does the gdpr need a new exception? Computer Law & Security Review **48**, 105770 (2023)

[42] Veale, M., Binns, R.: Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. Big Data & Society **4**(2), 2053951717743530 (2017)

[43] Ruggieri, S., Pedreschi, D., Turini, F.: Data mining for discrimination discovery. ACM Transactions on Knowledge Discovery from Data (TKDD) **4**(2), 1–40 (2010)

[44] Suresh, H., Guttag, J.V.: A framework for understanding unintended consequences of machine learning. ArXiv **abs/1901.10002** (2019)

[45] Hoffmann, A.L.: Where fairness fails: data, algorithms, and the limits of antidiscrimination discourse. Information, Communication & Society **22**(7), 900–915 (2019)

[46] Darling-Hammond, L.: Unequal opportunity: Race and education. The Brookings Review **16**(2), 28–32 (1998)

[47] Barocas, S., Guo, A., Kamar, E., Krones, J., Morris, M.R., Vaughan, J.W., Wadsworth, W.D., Wallach, H.: Designing disaggregated evaluations of ai systems: Choices, considerations, and tradeoffs. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pp. 368–378 (2021)

[48] Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E.P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 329–338 (2019)

[49] Luong, B.T., Ruggieri, S., Turini, F.: k-nn as an implementation of situation testing for discrimination discovery and prevention. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 502–510 (2011)

[50] Madaio, M.A., Stark, L., Wortman Vaughan, J., Wallach, H.: Co-designing checklists to understand organizational challenges and opportunities around fairness in ai. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–14 (2020)

[51] Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., Wortman Vaughan, J.: Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–14 (2020)