Building and Auditing Fair Algorithms: A Case Study in Candidate Screening

Christo Wilson
Northeastern University
cbw@ccs.neu.edu

Avijit Ghosh
Northeastern University
avijit@ccs.neu.edu

Shan Jiang
Northeastern University
sjiang@ccs.neu.edu

Alan Mislove
Northeastern University
amislove@ccs.neu.edu

Lewis Baker
pymetrics, inc.
lewis@pymetrics.com

Janelle Szary
pymetrics, inc.
janelle@pymetrics.com

Kelly Trindel
pymetrics, inc.
kelly@pymetrics.com

ABSTRACT

Academics, activists, and regulators are increasingly urging companies to develop and deploy sociotechnical systems that are fair and unbiased. Achieving this goal, however, is complex: the developer must (1) deeply engage with social and legal facets of "fairness" in a given context, (2) develop software that concretizes these values, and (3) undergo an independent algorithm audit to ensure technical correctness and social accountability of their algorithms. To date, there are few examples of companies that have transparently undertaken all three steps.

In this paper, we outline a framework for algorithmic auditing by way of a case-study of PyMetrics,a startup that uses machine learning to recommend job candidates to their clients. We discuss how PyMetrics approaches the question of fairness given the constraints of ethical, regulatory, and client demands, and how PyMetrics's software implements adverse impact testing. We also present the results of an independent audit of PyMetrics's candidate screening tool.

We conclude with recommendations on how to structure audits to be practical, independent, and constructive, so that companies have better incentive to participate in third party audits, and that watchdog groups can be better prepared to investigate companies.

1 INTRODUCTION

Increasing concern over bias in automated systems has led to an outcry for companies incorporate fairer and more transparent systems [20]. However, automated systems are complex, requiring that developers (1) deeply engage with social and legal facets of "fairness" in a given context, (2) develop software that concretizes these values, and (3) undergo an independent algorithm audit to ensure technical correctness and social accountability of their algorithms. To date, there are few examples of companies that have transparently undertaken all three steps.

PyMetrics is a startup that offers a candidate screening service (a.k.a. pre-employment assessment) to employers based on data and applied machine learning (ML). One of the core assertions PyMetrics makes about their service is that they pro-actively debias ML models before deployment to comply with the U.S. Uniform Guidelines on Employee Selection Procedures (UGESP) [16]. PyMetrics claims to use an outcome-based model de-biasing process [50] where candidate models are assessed for compliance with the UGESP "four-fifths" rule using minimum bias ratio as a metric and then retrained as necessary to ensure compliance [42, 45].[1]

In this paper, we outline our process of auditing PyMetrics as a case-study for creating transparent and accountable systems. We have two goals: (1) to present the process we used to audit PyMetrics's candidate screening product as a replicable framework, and (2) to present the results of this specific audit.

With regards to process, we introduce the *cooperative audit* as a framework for external algorithm auditors to audit the systems of willing private companies. Cooperative audits are unlike existing proposals for audits that involve insider employees [51] or outsider audits where the target company is unaware of the testing (e.g., [6, 53, 60]). Given the unique challenges of the cooperative format, we needed to develop careful protocols to ensure the independence and transparency of the auditors and the audit itself.

With respect to the audit of PyMetrics, we scoped the audit to five specific questions all related to the fairness guarantees that PyMetrics claims their system implements. We leveraged source code analysis, consideration of human behavior, and statistical analysis of data to investigate these questions. We found that PyMetrics's candidate screening service did faithfully implement the stated fairness guarantees, and that their systems included sufficient safeguards against human error and intentional malicious behavior to reasonably ensure compliance with the four-fifths rule.

FAccT '21, March 2021, Toronto, Ontario, Canada

ACM ISBN XXX-XXX-XXX…$15.00
https://doi.org/10.XXXX/XXXXXX.XXXXXX

[1] Or, if no compliant model can be found, PyMetrics abandons the training process and no model is deployed.

We conclude with recommendations on how to structure cooperative audits to be practical, independent, and constructive. By developing the framework for cooperative audits and demonstrating the outcome of one such audit, we hope to incentivize companies to participate in more third-party algorithm audits. Furthermore, we aim to provide guidance for watchdog groups that may wish to become third-party auditors, and regulators interested in how to structure mandatory third-party audits of systemically important sociotechnical systems.

2 BACKGROUND

We begin by presenting context for our audit, starting with concerns about the use of ML in hiring and followed with a brief introduction to the practice of algorithm auditing.

2.1 Fairness in Algorithmic Hiring

Since at least the mid-1990s, critical scholars have been raising alarms about the potential for computer systems to embed, entrench, and compound social biases [24]. These voices have grown louder and the concerns more acute as data-driven ML systems have seen increased adoption [5].

The adoption of ML techniques in the domain of hiring is particularly contentious. On one hand, discrimination in hiring driven by human biases is a long standing and widespread problem [49]. From this perspective, using ML to evaluate job seekers has the potential to remove human biases from the hiring process, potentially leading to more equitable outcomes. On the other hand, there is no reason to assume a priori that ML systems in the hiring domain will automatically be "objective," "neutral," or "bias-free." Indeed, algorithm audits of gig-economy marketplaces and traditional resume boards have uncovered race and gender biases in these systems [8, 27].

As startups have emerged that apply ML to the hiring process, scholars have begun to investigate the legal, conceptual, and practical space in which they operate. Raghavan et al. surveyed publicly available information about 18 startups offering *pre-employment assessment* systems to map their practices into the law and policy space, especially claims about compliance with the UGESP's four-fifths rule [50]. Ajunwa and Kim both present extensive taxonomies of the ways that bias may emerge in ML-based hiring systems and map these to U.S. legal doctrine [1, 33]. Whereas these studies reported aggregated, publicly available information, the current study investigate a single company to an unprecedented level of access.

Title VII of the U.S. Civil Rights Act of 1964 distinguishes between two forms of discrimination that may impact hiring processes: *disparate treatment* and *disparate impact* [59]. The former refers to cases where people are directly discriminated against based on legally protected attributes, such as race and gender. In the ML context, avoiding disparate treatment is often operationalized as a prohibition against the use of protected attributes as input features for models [1, 8]. Disparate impact refers to cases where a facially neutral process still produces substantially different outcomes for people that are correlated with legally protected attributes. With respect to ML, there are many de-biasing techniques that aim to ensure that models do not produce disparate impact [23, 38, 48, 63], although scholars have found that not all of these fairness objectives are mathematically compatible [22, 34]. We examine both disparate treatment and disparate impact in our audit of PyMetrics.

2.2 Algorithm Auditing

Raji et al. write that "audits are tools for interrogating complex processes" [51]. With respect to modern sociotechnical systems, Sandvig et al. motivate the need for algorithm audits as a means "to investigate normatively significant instances of discrimination involving computer algorithms operated by Internet platforms" [56]. While some have likened algorithm auditing to reverse engineering by outsiders, in that the goal is to make "black-box" systems more transparent regardless of the system creator's intent [13], this conceptualization has since been expanded to include audits carried out by ethically and morally-conscious insiders [51].

There is a growing body of algorithm audits assessing a variety of systems for a diverse set of harms. This includes examining broad classes of systems like search [14, 14, 25, 28, 32, 35, 36, 41, 53–55], e-commerce [10, 26], news recommendation [4, 31], online advertising [58, 60], maps [57], ridesharing [9, 30], online reviews and ratings [17, 18], natural language processing [7], and recommendation [29]. Some audits have specifically focused on algorithms in high-stakes contexts like facial recognition [6], predictive policing [2, 15, 39], housing [3], and child protective services [11].

In the absence of regulation or accepted best-practices, recent work has attempted to define a process for algorithm auditing. Raji et al. developed a six step process for auditing that we expand upon in our audit of PyMetrics [51]. This process includes scoping (see § 4.1), mapping (which involves interviewing stakeholders, see § 4.3.3), artifact collection (again, see § 4.3.3), testing (§ 5), and reflection (of which a large part is generating reports like this one).

Raji et al. draw a distinction between *internal* and *external audits* [51]. In Raji et al.'s parlance, an internal audit of (e.g.,) PyMetrics would be conducted by PyMetrics employees, while an external audit would be conducted by experts with no association to PyMetrics and no privileged access to PyMetrics's systems. The audit we present in this study does not fall into these paradigms: we were not employees of PyMetrics, yet we were given privileged access to PyMetrics source code and documentation (see § 4.3.3). Thus, we refer to this endeavor as a *cooperative audit* as it involves cooperation between internal and external actors.

Sandvig et al. introduced five designs for conducting algorithm audits [56]. Our study of PyMetrics corresponds most closely with a *code audit* in this taxonomy since we directly examined PyMetrics's source code and datasets. However, Sandvig et al. assume that source code will be publicly available so that a key precept of classic audit study design can be maintained: that the audited party not be aware of the audit. This assumption is not true in our case, since our audit was cooperative. As we discuss in § 4.3.4, we undertook other steps to insure our independence from PyMetrics.

Several tools have been developed by academics to facilitate auditing of black-box ML models, including LIME [52] and SHAP [40]. These advanced statistical tools were not necessary for this audit since (1) training data and source code were available to us and (2) PyMetrics uses interpretable models (see § 3.2).

3 ABOUT PYMETRICS

In this audit we focus on the candidate screening (a.k.a. pre-employment assessment [50]) product offered by PyMetrics. PyMetrics is a startup that offers a number of services in the context of employment. Unlike job board services like Monster.com or Indeed, PyMetrics is not a marketplace where employers post jobs or job seekers post resumes. Rather, PyMetrics uses gamified psychological measurement and applied ML to evaluate the cognitive and behavioral characteristics that differentiate a role's high-performing incumbents to make predictions about job seekers applying to that role.

PyMetrics' candidate screening service is designed to surface the applicants with the greatest potential and pass them on to the interview stage while simultaneously seeking to avoid disparate impact by abiding by the UGESP's four-fifths rule with respect to protected demographic groups. At a high-level, PyMetrics' candidate screening service can be summarized as follows [45]:

(1) An employer contracts with PyMetrics to develop and deploy a predictive model for candidate screening. We refer to these employers as *clients*.
(2) A *job analyst* from PyMetrics surveys the client to understand the *target role* (e.g., the job description, seniority-level, etc.) and the metrics that the client uses to assess job performance in that role [44].
(3) The client has incumbent employees in the targeted role play PyMetrics's suite of games (described further in § 3.1). The client also gives existing job performance data about these incumbents to PyMetrics. This performance and gameplay data are used as the training input for a predictive model.
(4) A PyMetrics data scientist uses a proprietary PyMetrics tool to develop a predictive model for the client. These models are evaluated for predictive performance and compliance with the UGESP's four-fifths rule using a separate held-out testing set with demographic information.
(5) PyMetrics deploys the best-performing predictive model that meets the fairness criteria. Job seekers who apply for the targeted role are asked to play PyMetrics' suite of games. Based on this gameplay data, the model predicts which candidates have similar attributes to the clients' high-performing incumbent employees. Information about high-scoring job seekers are sent to the client, who may then apply additional filters (e.g., resume screening) and interview candidates.
(6) PyMetrics performs longitudinal analysis of the predictive model. This includes *back-testing* to re-evaluate whether fairness criteria are being met with respect to the pool of job seekers that have applied for this role, and studying the job performance of candidates who were hired.

3.1 Data Sources

PyMetrics' candidate screening service relies on a variety of data sources to train and evaluate ML models. The primary data source is a core set of twelve games that are derived from peer-reviewed psychological studies. These games are purported to assess intrinsic mental qualities of individuals — the games are not meant to be won or lost, but rather to surface information based on how people play. Each game produces a number of features per player. These games are available in internet browsers or in a mobile app, are translated into over 20 languages, and have built-in accommodations for players with color-blindness and/or dyslexia. At present, PyMetrics maintains a database with gameplay from over 2 million users from across the world and a variety of industries (see § 5.4).

After players complete the PyMetrics games they are asked to take an optional demographic survey about their gender and ethnicity/race. The available categories correspond to those delineated by the EEOC for adverse impact testing. While PyMetrics allows for other responses, the categories considered as protected by the EEOC include male/female for gender, and Asian/Black/Hispanic/White/two-or-more groups for ethnicity. PyMetrics reported to us that over 75% of players complete the demographic survey [47]. This data is used to construct the held-out sets of data that are used for adverse impact testing.

3.2 Model Training

The goal of PyMetrics' models is to identify features that characterize clients' high-performers so that predictive models can accurately and fairly identify potential high-performers from pools of applicants. To support this goal, PyMetrics constructs three datasets: an *in group*, a baseline for comparison called an *out group*, and a held-out set for testing adverse impact called a *bias group*.

The *in group* is composed of the gameplay data from high-performing incumbent employees at the client company, as identified through the job analysis process. The *in group* dataset typically contains data on 50–100 players. The *out group* is used as a point of comparison in the training process, and is sampled from the PyMetrics database to approximate the potential applicant pool. This allows for the predictive model to isolate the behaviors that highlight potential hires from a candidate set. Finally, the *bias group* is the set of gameplay data from users in the PyMetrics database who voluntarily provided demographic labels. The *bias group* dataset typically contains over $10k$ users and is engineered to include an equal proportion of players from each of the EEOC's protected groups. The *bias group* is the dataset used for testing adverse impact.

Following construction of these datasets, the next step in PyMetrics's process is to clean the data. This involves:

(1) Correct for platform differences. Data across multiple platforms may be scaled to account for population-level differences due to platform effects, e.g., some games elicit different behaviors given the affordances of web and mobile app-based platforms.
(2) Remove players with missing values. Recall that PyMetrics asks players to complete twelve games. It is possible that a player may skip games, or start games but abandon them midway. These actions result in missing feature values. By default, gameplay from users with more than two missing games is considered incomplete, and those players are removed from analyses.
(3) Clean the feature values and remove outliers. All features are checked to make sure they fall within acceptable numerical ranges, corresponding to the bounds of each game. Acceptable bounds are set for each game through psychometric and statistical testing such that typical boundaries are

within several standard deviations from the mean. Values outside the acceptable range are rounded to the minimum or maximum of the range, respectively.

(4) Impute missing feature values. A player may have an empty value for a feature for a number of reasons. As mentioned above, the player may intentionally or unintentionally skip a game or part of a game, or have experienced a technical or connectivity error. PyMetrics replaces missing feature values with the median values for that feature.

(5) Scale the feature values. Each feature distribution is re-centered around zero and scaled to unit variance.

PyMetrics then uses a proprietary implementation of a Support Vector Machine (SVM) algorithm [12] to train predictive models. The SVM class of models is a reasonable choice for scenarios where (1) the feature space is known and small (64 features in the code we audited), and (2) the amount of labeled training data is small in absolute terms and in relation to the amount of unlabeled training data. The *in group* and *out group* are the only inputs for the training and hyperparameter optimization stages of the SVM model.

To test whether the predictive model meets PyMetrics' fairness criteria, PyMetrics conducts a search for the most predictive, leased biased permutation of features. Fairness is measured by applying the predictive model to the *bias group* data, and comparing performance of the demographic subgroups as described in § 3.3.

If no models are found that meet PyMetrics standards for both performance and fairness, the job analyst may continue to work with the client to reconstruct appropriate incumbent selection criteria. Otherwise, the data scientist will deploy the most performant, fair model for the client to use as part of their selection process.

3.3 Adverse Impact Testing

According to PyMetrics [45], all deployed models comply with the UGESP's four-fifths rule. In practice, this means that the pass rate, or impact ratio (IR), of the lowest-passing group over the pass rate of the highest-passing group, must always be greater than 0.8. While the source code for PyMetrics model building is proprietary and was shared with us only for the purposes of this audit, PyMetrics has provided their adverse impact testing framework as an open source tool [46]. In addition to calculating IR, the library also utilizes Bayesian estimation and statistical tests such as the z-test, Analysis of Variance (ANOVA), χ^2 test, and Fisher's exact test.

In the PyMetrics use-case, the trained model scores job seekers who are then sorted into three tiers: "Highly Recommended," "Recommended," and "Not Recommended". The tiers are based on percentile thresholds that can be customized for each client, but are typically set at the 50^{th} and 70^{th} score percentiles (so that applicants with scores falling in the 70^{th} percentile or greater are in the "Highly Recommended" tier, those with scores between the 50^{th} and 70^{th} percentiles are in the "Recommended" tier, and those with scores below the 50^{th} percentile are "Not Recommended"). This categorization raises a complication with respect to measuring compliance with the four-fifths rule: the pass rate for a given demographic group may vary based on the chosen threshold used to operationalize success. According to PyMetrics' documentation [45], the search for fair feature sets considers the IR at the 70^{th} percentile, but the final models are tested for fairness at both the 50^{th} and 70^{th} percentiles. PyMetrics claims that a model that does not pass all fairness and checks using the *bias group* will not be deployed. After a model has been deployed and used in a client's application process, PyMetrics continuously performs both practical and statistical adverse impact testing using real applicant data. This audit focuses only on the pre-deployment testing claims of PyMetrics that all deployed models will have $IR \geq 0.8$ at the 50^{th} and 70^{th} percentiles.

4 DESIGNING THE AUDIT

In this section we discuss the design of our audit, including what we examined, what we did not examine, the baseline requirements for conducting the audit, and how we managed our relationship with PyMetrics. We provide our scope in more general terms as a template for future auditors, and provide our results as they specifically pertain to PyMetrics.

4.1 In Scope

The focus of our audit was PyMetrics's claim that their model training process produces models that abide by PyMetrics's interpretation of the UGESP's four-fifths rule [16]. To this end, we examined documentation and source code that implements PyMetrics's candidate screening product, as described in § 3. We conducted our audit in summer 2020.

During the audit we focused on the following specific questions:

(1) **Correctness.** PyMetrics's documentation describes their process for performing adverse impact testing on trained models before they are deployed. *Does the model training source code correctly implement adverse impact testing as the four-fifths rule using the minimum bias ratio (a.k.a. impact ratio) metric as described in the documentation? Is fairness assessed for the seven demographic categories defined by the EEOC (five racial and ethnic, two gender)?*

(2) **Direct Discrimination.** Using demographic data as training features for models can be construed as a form of direct discrimination. This motivates us to ask *do trained models use demographic data directly as input, or is demographic data only used for post-training adverse impact testing?*

(3) **De-biasing Circumvention.** There are numerous examples of deployed ML-based systems that had their safety systems subverted by clever and malicious users [10, 27, 37, 61, 62]. These experiences motivate us to ask *is there any way for training data that is erroneously corrupted or intentionally biased to somehow avoid the adverse impact tests, thus resulting in an unfair model being released?*

(4) **Sociotechnical Safeguards.** PyMetrics's process for producing models involves human intervention, which raises the issue that human errors may subvert fairness guarantees. *Does PyMetrics have checks in place to ensure that human errors (either benign or malicious) do not result in an unfair model being released?*

(5) **Sound Assumptions.** Using ML is never as simple as loading data and inputting it into a training algorithm. Data must be preprocessed and transformed to prepare it for analysis. This process concretizes assumptions about the data that may influence the adverse impact assessment. *Are there*

assumptions about data and data preprocessing baked into PyMetrics's model training process that could cause the adverse impact assessment to fail or mislead?

4.2 Out of Scope

Just as important as defining what we **were** auditing is understanding what we **were not** auditing. This point is critical for properly contextualizing any audit, so as to focus on specific criteria for success or failure. In particular, our audit did not cover the following aspects of PyMetrics's products and business.

- Prior to conducting the audit, we agreed with PyMetrics that we would not question their choice of fairness objective (the UGESP four-fifths rule) or fairness metric (minimum bias ratio). Although there are many other potential fairness objectives and metrics [23], including others designed to prevent disparate impact [38, 48, 63], PyMetrics chose their existing objective and metric based on what they felt was most appropriate in the context of their business, i.e., candidate screening. This objective and metric were proposed by the relevant U.S. regulators themselves.
- Similarly, we agreed not to question PyMetrics's choice of race, ethnicity, and gender categories that they evaluate for fairness since these categories are delineated as protected by the EEOC. Further, we agreed to not evaluate fairness for intersectional groups (i.e., combinations of demographic categories like Black males or Asian females) since they are not considered protected by the EEOC.
- We only audited PyMetrics's game-based candidate screening product. We did not audit other products and services.
- We did not investigate the ability of PyMetrics's games to measure human capabilities, whether those capabilities map to job performance, or whether other assessment methods would be superior in some respect (e.g., fairness or accuracy). As computer scientists, evaluating these aspects of the PyMetrics system were beyond our capabilities. Additionally, we do not comment on the rationality and ethics of using these measures to evaluate a candidate's suitability for employment.
- PyMetrics recently started offering an additional suite of numerical and logical reasoning games. We did not have access to datasets that included data from these games, so we cannot comment on their impact to fairness. That said, as we discuss in § 5.1, the control flow in the PyMetrics source code ensures that all models eventually must pass fairness checks, regardless of whether the model includes or does not include data from these additional games.
- PyMetrics performs post-training adverse impact testing on models using a held-out set of data [42, 50]. Prior to conducting the audit, we agreed with PyMetrics that we would not question their choice to use post-training testing. While pre-training [19] and during-training model debiasing methods [63] exist, they require that training data include complete demographic information, which is not always available in employment contexts.[2]
- During our audit, we did not focus on evaluating or maximizing the predictive performance of PyMetrics's models — fairness was our main concern. That said, during our testing, we did obey the minimum baseline predictive performance requirements that PyMetrics demands of all their models.
- We did not audit PyMetrics's process for performing annual adverse impact back-testing on deployed models [45].
- We did not examine PyMetrics's cybersecurity posture, e.g., we did not perform penetration tests. We did not attempt to become a PyMetrics client, play their games while posing as an employer or a job seeker, have any contact with PyMetrics employees outside the narrow confines of this audit, or attempt to conduct insider attacks given our privileged access to PyMetrics systems, data, and employees.
- We did not examine PyMetrics's posture with respect to data privacy or compliance with laws like Europe's General Data Protection Regulation, the California Consumer Privacy Act, the U.S. Children's Online Privacy Protection Act, etc. However, PyMetrics has developed an information security program compliant with the internationally recognized ISO/IEC 27001 information security standard [21] and undergoes semiannual security audits by an internationally accredited certification body.

4.3 Requirements

To fully and completely answer the questions we posed in § 4.1 in a way that the public can trust necessitated that we establish a number of requirements for our audit. Major University[3] and PyMetrics signed a contract in March 2020 agreeing to the scope of the audit and these requirements before we commenced the audit. Here, we outline the requirements of our audit.

4.3.1 Transparency. One of the foremost issues we identified heading into the audit was how to preserve our objectivity and trustworthiness. These properties are essential, both from the audit design perspective (i.e., are we asking the right and tough questions) and from the public reporting perspective (i.e., will people believe the results of the audit).

To promote these properties, we adopted a stance of transparency. The contract between Major University and PyMetrics, the audit and data sharing protocol agreed to by the the audit team and PyMetrics, the non-compete signed by the lead auditor, and the project budget are publicly available online for anyone to inspect.[4] As stipulated in the contract, the only facets of the project covered by non-disclosure rules include source code, data, and internal manuals from PyMetrics. Otherwise, the audit team retained the right to speak and publish about the audit, up to and including specific results that might reflect negatively on PyMetrics.

As specified in the contract, we adopted a policy of *responsible disclosure*: the audit team agreed not to publicly discuss problems

[2] For example, an employer may not be willing to divulge demographic information about incumbent employees due to privacy and consent concerns.
[3] The auditor's institution, anonymized for blind review.
[4] To preserve anonymity, all documents will be publicly released after paper acceptance.

or issues uncovered during the audit until PyMetrics was given 30 days to privately remediate the issues. This policy was modeled on industry standard disclosure practices from the realm of cybersecurity. Likewise, while we gave PyMetrics an opportunity to review documents generated from the audit, we retained final editorial discretion.

4.3.2 Remuneration. Another fraught issue that we identified was remuneration: should PyMetrics pay for the audit, and if so, how? On one hand, we as auditors wanted to avoid real and perceived conflicts of interest. Accepting payment for the audit would immediately raise legitimate concerns about our objectivity.

On the other hand, performing a pro-bono audit raises the issue of setting an unrealistic, exclusionary precedent. To our knowledge, there have been very few cooperative algorithm audits between companies and independent investigators. It is our sincere hope that our audit of PyMetrics will serve as model for more cooperative audits in the future. With this framing in mind, if we established a precedent that only pro-bono audits are sufficiently objective, this could severely limit who is able to serve as an auditor in the future. In the absence of funding from neutral sources (e.g., grants from government, foundations, or endowments), many academics, investigative journalists, etc. may not have the financial means to support themselves while conducting audits.

Further, completing the audit pro-bono raises issues of exploitation. Auditing is challenging work that requires a high-level of expertise. Especially when graduate student labor is involved, as it was during this audit, setting a precedent of work without compensation exacerbates pre-existing power imbalances in academia.

We asked several people in the academic community who are well versed in the study of online platforms for advice on how to approach the issue of remuneration. Based on their helpful guidance, we decided to accept payment for the audit subject to a number of constraints. *First*, the payment was structured as a grant to Major University, not as direct payment to the auditors as independent contractors. This added a layer of institutional oversight to the project. *Second*, all payment was received before we delivered the findings of the audit to PyMetrics, thus mitigating concerns that payment could be conditioned on positive audit results. *Third*, PyMetrics provided computational processing power for the audit at their own expense. This was done so that the material expense of the audit was not put on the auditors.

4.3.3 Access and Materials. To complete this audit, we were given extraordinary access to PyMetrics. At the outset, we spent a day with PyMetrics employees learning about the company, their data science pipeline, and how they approach fairness issues, as well as demos of PyMetrics data scientists using their internal tools to train and evaluate models. During this "onboarding" we were also given copies of internal PyMetrics documents that present, among other things, a technical overview of their candidate screening product [45] and specific details about their fairness testing procedure [42]. PyMetrics makes these documents available to prospective clients under a non-disclosure agreement.

PyMetrics gave us access to source code for their candidate screening product. At a high-level, the source code encompasses a "template" Jupyter notebook that is used by PyMetrics data scientists, along with associated custom Python modules. The notebook implements the data scientist-facing process of producing a predictive model for a specific client, including presenting the results of adverse impact testing. The Python modules implement specific algorithms that are generally constant across all client engagements, such as model search and the minimum bias ratio metric.

We were given eight notebooks in total:

- **Blank Template.** One notebook was "blank," in the sense that it had not been filled out by a data scientist. In other words, this is how the template notebook appears to a PyMetrics data scientists that is beginning a new client engagement from scratch — it contains scaffolding code and processes but no specifics.
- **Representative Samples.** Six notebooks were sampled from recent completed client engagements. These notebooks had each been filled out by a PyMetrics data scientist and produced a model that ultimately went live into production. Five of these notebooks were selected uniformly at random by PyMetrics from client engagements that had occurred within the six months preceding the audit. The sixth notebook was chosen because it came from a recent engagement where the client requested extensive changes to the adverse impact testing process.
- **Complete Engagement.** The final notebook also came from a completed client engagement, and included the associated datasets that were used to train and evaluate the models. The bulk of our attention during this audit was on this "complete" notebook and its associated data.

The seven completed notebooks were anonymized to remove specific references to the client companies. The data from the complete engagement was pseudonymous: it contained no personally identifiable information (PII), but gameplay and demographic data was associated with individual game players.

All of these notebooks and data were uploaded to a virtual machine that was provisioned by PyMetrics and hosted on Amazon Web Services. The audit team performed all analysis within this virtual machine. We agreed to confine our activities to within this virtual environment to obviate PyMetrics's concerns about their proprietary code and data being leaked.

Finally, after the completion of our analysis, we observed a live demonstration of PyMetrics data scientist training, testing, and deploying a model into production. This demonstration allowed us to confirm that the notebooks we analyzed were representative of what PyMetrics uses in production.

4.3.4 Independent Testing. An essential principle of auditing is that, to the greatest extent possible, the subject should not know the manner in which they are being evaluated or be allowed to dictate the tests that will be conducted. In keeping with this principle, we did not inform PyMetrics of the tests or testing methods we planned to use before commencing the audit. The extent of PyMetrics's knowledge prior to conducting the audit was (1) that the audit was taking place, (2) that we would be evaluating the model training and testing portion of their game-based candidate screening product, and (3) that we might employ "fake" or synthetic data in our testing.

We maintained this posture of secrecy throughout the course of the audit. During our initial onboarding with PyMetrics staff

we made sure to ask only general questions that would not reveal the focus of our testing or our methods. Similarly, at several points during the audit we required technical assistance to run PyMetrics code, as well as additional datasets that were not provided initially. During these interactions were also opaque, and did not elaborate on why we wanted things or our specific motivations. The PyMetrics team honored our requests and did not attempt to extract information about the status of our testing.

4.3.5 Deliverables. As outlined in our contract with PyMetrics, the required deliverable from this audit was a report that they planned to distribute to their clients. After the conclusion of the audit, the auditors and PyMetrics mutually agreed to prepare this manuscript in the interest of widely disseminating the methods and results of the audit.

4.4 Limitations

As with any scientific study, it is critical to be forthright about the limitations of our audit study.

First, we operated under an assumption of good faith on the part of PyMetrics. We assumed that documentation, source code, and data that we received were representative of the actual, deployed systems and data used by PyMetrics. Given that PyMetrics agreed to full transparency of this audit, we have no reason to doubt their sincerity. Additionally, we observed a live demonstration of a model being trained, tested, and deployed by a PyMetrics data scientist. This demonstration allowed us to confirm that the "production" source code and process matched the one we audited.

Second, our audit examined PyMetrics's fairness claims and source code in summer of 2020. We cannot make any claims about PyMetrics's practices and guarantees before we ran our audit, or about new and modified products they release in the future.

Third, PyMetrics may customize their model training and adverse impact assessment process for specific clients. We cannot make claims about these customized products. Our audit results are only representative with respect to PyMetrics's standard, non-customized model training and adverse impact assessment process.

5 RESULTS

In this section we present the results of our audit. We organize our discussion around the five questions given in § 4.1 using the source code and data introduced in § 4.3.3.

5.1 Overall Implementation

We began by addressing three questions:

(1) **Correctness.** Does PyMetrics's source code faithfully implement the four-fifths rule via the minimum bias ratio metric, using the methods described in their documentation?

(2) **Direct Discrimination.** Do models trained using the PyMetrics source code directly incorporate demographic features, and/or do the models take demographic features as direct input?

(3) **De-biasing Circumvention.** Is there a way to manipulate the input data to the PyMetrics source code in such a way that the fairness checks are circumvented?

To answer these questions, we manually examined the source code provided by PyMetrics that we introduced in § 4.3.3. With the exception of one notebook that was heavily modified to suit a particular client, the remaining six notebooks had consistent source code and use of custom modules.

With respect to correctness, we found that PyMetrics's source code did implement the four-fifths rule using the minimum bias ratio metric, with the seven considered groups being the EEOC-defined categories of male, female, White, Black, Hispanic, Asian, and people who identify with ≥ 2 racial or ethnic groups. The code for calculating these metrics was inside a custom Python module and we found no issues with these algorithms. We also found that the adverse impact metrics were prominently reported to the overseeing data scientist multiple times in the notebook.

With respect to direct discrimination, we confirmed that players' demographic characteristics were not used as features for model training. The *in group* and *out group* datasets used for model training did not contain demographic information, nor did it contain any overt proxies for demographic information (e.g., no zip codes or biometric information). Only players in the *bias group* had corresponding demographic information, and the *bias group* was only used for feature and model evaluation.

With respect to circumventing fairness checks, we were unable to find a way to produce a biased model that was not flagged as such by the code. For the purpose of this testing we assume the following *threat model*: a client may arbitrarily manipulate the *in group* dataset that they supply to PyMetrics, e.g., by controlling its size or the gameplay data of players.[5] By manipulating the *in group* data, the malicious client's goal is to get PyMetrics to unwittingly deploy a biased model.

Our threat model is intentionally abstract to cover a wide range of potential malicious behaviors. In practice, a PyMetrics client could, hypothetically, engage in several specific types of malicious behavior. One possibility is that a client could supply an *in group* dataset that contains information from a demographically homogeneous group of employees. Another possibility is that a client could lie by (1) having a single employee play PyMetrics's games 50 times and then (2) supplying PyMetrics with fabricated performance data for 50 imaginary employees.

In our testing, we were unable to circumvent the fairness checks in PyMetrics's source code by manipulating *in group* data. All control flow paths in the Jupyter notebook eventually arrived at the adverse impact tests. We could generate *in group* data that would cause the model search to fail the adverse impact tests, but the model deployment process could not continue unless a compliant model was produced. Alternatively, in some cases the model building process was able to successfully de-bias our malicious *in group*, which also meant that our attack had failed.

[5]Note that this threat model is unrealistically strong. In practice, it would be very difficult for a client to produce arbitrary gameplay data, since they would either need to train human beings to play the PyMetrics games in very specific ways, or write software to emulate a human and play the games. Further, PyMetrics's clients work closely with a human job analyst from PyMetrics to select incumbent employees and fairly evaluate their performance. A malicious client would need to lie to the job analyst in addition to producing manipulated data.

5.2 Human Oversight

The next question we examined concerned **sociotechnical safeguards**. Rather than being a fully automated process, models at PyMetrics are crafted by hand. Involving data scientists has benefits: they can notice and correct issues during model building, and tweak models to better support the unique needs of clients.

However, human involvement also raises concerns, like whether a negligent or malicious data scientist could release a biased model into production. There were no programmatic constraints in the Jupyter notebook that prevented a data scientist from training a model, failing to check it for bias, and uploading it to PyMetrics's back-end production system. Further, building such programmatic constraints would be extremely difficult and perhaps impossible — this is a fundamental tradeoff that comes with empowering human data scientists to participate in the model training process.

To mitigate these human risks, PyMetrics relies on a system of manual review that is similar to the code review practices that are common in serious software development projects. After a data scientist has finished training, evaluating, and packaging a model for given client they must complete a checklist that includes over 100 questions. These questions review all of the key aspects of the model building process. The data scientist not only needs to "check the boxes," but also copy salient numerical data (e.g., accuracy and pass rates) into the sheet and document in writing any significant deviations from the standard "templated" model training process. While much of this process could, in theory, be automated, PyMetrics deliberately adopted a manual process that forces the data scientist to document and justify their work.

Before a trained model can be released into production, the corresponding notebook and checklist are reviewed by a second data scientist from PyMetrics. The second data scientist provides extra bulwark against unintentionally erroneous models being accidentally put into production. Further, it would now take collusion between two data scientists to maliciously release a biased model into production.

In our opinion, this manual review process offers a reasonable level of assurance against malicious insiders and negligence.[6] The design of this review process forces self-reflection by data scientists and is so detail-oriented that it would be difficult to unintentionally miss substantive problems with a trained model. The addition of a second reviewer guards against gross negligence and raises the bar against intentional malfeasance.

5.3 Cleaning and Imputation

Next, we investigate the **soundness of assumptions** underlying the process PyMetrics uses to prepare data for model training and evaluation. As described in § 3.2, data preparation is a complex task that involves many choices: what data to use, how to clean it to filter outliers, how to impute any missing values in the data, and how to normalize and scale the numerical data. These choices may impact model performance and fairness guarantees, so they are worth critically interrogating.

[6] We note that our opinion is based on assumptions about the threat model PyMetrics faces, e.g., a malicious company that might want to get PyMetrics to bless their biased hiring practices as fair. See § 5.1 for more discussion of this assumed threat model.

Missing	Demographic 1	Demographic 2	MW U	KW H
Games	Asian	Black	2864400.0	11.0^{*}
	Black	White	2894400.0^{*}	9.0^{*}
Traits	Female	Male	14912935.0^{*}	9.3^{*}
	Asian	Black	2654601.0	37.0^{***}
	Black	Hispanic	3043934.5^{***}	19.0^{***}
	Black	White	3069502.0^{***}	26.0^{***}
	Black	Two-or-More	1583887.0^{***}	50.0^{***}
	Hispanic	Two-or-More	1503170.0^{**}	14.0^{**}
	White	Two-or-More	1486928.5^{*}	9.1^{*}

Table 1: Cases where missing game and missing feature distributions were significantly correlated with demographics. $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$.

	count(MW U $p < 0.05$)	%	count(KW H $p < 0.05$)	%
Gender	19	29.7	30	46.9
Race/Ethnicity	112	17.5	214	33.4

Table 2: Count and percentage of cases where feature distributions were significantly different between demographic groups.

During this audit, we focused on imputation of missing values as the area of concern. We focus on imputation because it is not optional, it will impact players even if filtering is applied, and there is a possibility that it may not impact all players equally (unlike scaling, which does effect all players equally).

5.3.1 The Differential Impact of Imputation. To motivate our investigation of imputation, we started by delving into two questions: (1) did some groups of players require more imputation than others, and (2) were the distributions of gameplay data for different groups statistically different? The first question sought to understand whether some groups are more impacted by imputation than others. The second question was driven by PyMetrics's choice of median imputation as their default algorithm — if groups exhibited different gameplay characteristics, then setting missing values to the population median might not reflect the gameplay characteristics of each group.

As shown in Table 1 and Table 2, the answer to both questions is yes. Using the non-parametric Mann-Whitney U and Kruskal–Wallis H tests,[7] we found that the distributions of missing data and gameplay data were significantly different across groups in many cases.[8]

5.3.2 Adverse Impact Testing and Model Performance. The results in the previous section suggested that imputation might have a differential impact on different groups of players. This motivated our second series of tests: evaluating models using different imputation algorithms to determine if the choice of imputer has a substantive impact on the fairness guarantees and performance of trained models.

To test this hypothesis, we re-evaluated the minimum bias ratios for the model that PyMetrics's data scientists selected as best for

[7] Using the Anderson-Darling test, we found that almost none of the data we analyze in this section is normally distributed, thus we rely on non-parametric tests.
[8] All p values are corrected for multiple hypothesis testing.

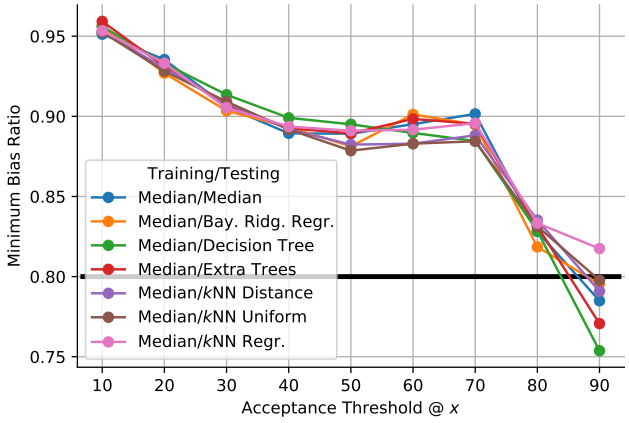

Figure 1: Minimum bias ratio at different acceptance thresholds for models trained using a dataset with median imputation, but adverse impact tested using datasets imputed with varying algorithms.

our *in group*, *out group*, and *bias group* datasets. Figure 1 shows the results when we re-evaluated the original model (which we denote as "Median" since it was trained on a median imputed dataset) using *bias group* datasets that were imputed with varying algorithms from `scikit-learn` version 0.23. "Median/Median" presents the original fairness metrics computed by PyMetrics. The x-axis denotes the acceptance threshold for employment candidates, with higher numbers corresponding to more stringent thresholds. The y-axis denotes the minimum bias ratio for the given model/evaluation data at a given acceptance threshold.

Figure 1 demonstrates that the choice of imputation algorithm did not substantively alter the adverse impact assessment of the original PyMetrics model. Despite using "better" evaluation datasets (i.e., produced by more accurate imputation algorithms), this model still passed the four-fifths test up to acceptance thresholds of 80. We also observed little variability at acceptance threshold less than 90, indicating that the fairness assessment of the model was stable. Additionally, we trained new models using data imputed with algorithms other than median, but were unable to find an alternate model that outperformed PyMetrics's original model in terms of fairness or predictive performance.[9]

Based on the analysis in this section, we conclude that the median imputer used by PyMetrics does produce models that have sound fairness guarantees. Our testing demonstrated that the potential issues we uncovered in § 5.3.1 did not have a significant impact on compliance with the four-fifths rule.

5.4 Representativeness of the Bias Group

One aspect of PyMetrics that we were unable to directly audit concerns the composition of the *bias group* dataset. Although the *bias group* for a given engagement is psuedo-randomly selected from a large population (matched for language, platform and occasionally region from a pool of 600,000+ people, according to PyMetrics [47]), this does not necessarily mean that it is representative of potential job seekers. Since the *bias group* serves as the baseline for adverse impact testing, if it is not representative then the bias assessments may be flawed. We identify two issues:

(1) Only players who opt-in to the demographic survey are eligible for inclusion in the *bias group*. Although PyMetrics claims that the overall response rate to the survey is high (over 75% [47]), that does not necessarily mean that the response rate is independent of demographics.

(2) The *bias group* is drawn from prior job applicants, which is itself conditioned on the types of companies, job roles, and job locations that PyMetrics has solicited gameplay data for in the past. It is unclear what kinds of companies, job roles, and geolocations PyMetrics has developed models for in the past, or how this may impact the composition of people in the *bias group* datasets.

PyMetrics identified the first issue themselves and performed an internal study to investigate. PyMetrics provided a copy of this study to us, updated to reflect their dataset as of summer 2020 [47]. The study analysed data from four clients that agreed to disclose the demographic data of job applicants — PyMetrics was then able to compare the data collected by the clients to the demographic data PyMetrics collected during the corresponding gameplay sessions. In total this dataset covered around 40,000 people, with roughly 5,600 people confirmed to appear in both the client provided and PyMetrics datasets.

PyMetrics's study found that there was effectively no relationship between demographics and disclosure rates. Overall, PyMetrics observed that (1) people were more likely to reveal their demographics to PyMetrics than to clients, (2) that this behavior was consistent across clients and demographic groups, and (3) the proportion of people who revealed their demographics was consistent across the PyMetrics and client datasets. Taken together, these results provide some reassurance that biased survey response rates are not undermining the *bias group*. That said, the only way to revolve this issue definitely would be through a deep ethnographic study of job seekers.

With respect to the second issue, PyMetrics provided us with two datasets to shed light on the diversity of the *bias group* dataset. *First*, they provided summary data on the geographic distribution of players in the *bias group*, aggregated by country. PyMetrics had received data from players in 191 countries, with ~40% coming from the US, but also with significant numbers around North and South America, Europe, Southeast and East Asia, South Africa, and Australia. *Second*, they provided a dataset that mapped their 600+ active client engagements from January to October 2020 to O*NET occupations.[10] The data showed that PyMetrics had developed models for jobs that cover 16 of the 23 major O*NET groups (70% coverage) and 35 of the 98 minor groups (36%).[11]

These datasets provide some reassurance that PyMetrics's *bias group* dataset is relatively diverse along geographic and job category

[9] We omit these results for brevity.

[10] O*NET is a hierarchical taxonomy of employment areas, broad occupations, and detailed occupations developed and maintained by the U.S. Office of Management and Budget and the U.S. Department of Labor.

[11] We do not expect PyMetrics to cover all of these groups, such as Group 45: Farming, Fishing, and Forestry Occupations and Group 55: Military Specific Operations, since employers in these groups are unlikely to rely on predictive analytics for recruitment.

lines, although there is no guarantee that it is sufficiently diverse to cover all potential recruiting use cases.

Ultimately, our concerns about the representativeness of the *bias group*, and its potential impact on models' fairness guarantees, are only valid up to a point. PyMetrics claims to perform adverse impact back-testing on models after they are deployed, based on the data of players who applied to the corresponding jobs. If these tests reveal that a model is not meeting fairness guarantees, then PyMetrics decommissions the model and trains a replacement that is fair using the updated data. Given that there are an enormous number of factors that might cause the applicant pool for a particular job to diverge from a given reference population used for pre-deployment adverse impact testing, back-testing is a reasonable mitigation for identifying instances where modeling and testing assumptions diverge from reality.

6 CONCLUSION AND DISCUSSION

In this study, we present the process by which we audited PyMetrics's candidate screening tool and the results of our audit. The focus of our audit was on PyMetrics's claim that their trained ML models conform to the UGESP four-fifths rule using the minimum bias ratio metric. We conducted our audit in summer 2020 based on documentation, source code, and representative datasets that PyMetrics provided to us.

With respect to the results of our audit, we are comfortable stating that PyMetrics passed the audit, subject to the qualifications and limitations we state in § 4.2 and § 4.4.

This work was also intended as a case-study for practitioners on both sides of future audits. In the remainder of this section, we discuss lingering questions from the audit, opportunities for future work, and lessons for the practice of cooperative auditing.

6.1 Ethics

Throughout the audit process, we took great care to ensure that our audit was conducted in a manner consistent with community ethical norms. *First*, we minimized harm to end users (players) by not being given any access to personally identifiable information; we were only given access to the gameplay and demographic data that is used as input to PyMetrics's models. *Second*, we minimized harm to PyMetrics's clients (who were not direct participants in the audit) by similarly not being given any identifying information about them. *Third*, as described in § 4.3, we ensured the results of our audit would ultimately benefit the research community by agreeing with PyMetrics up front that (a) the audit would be as transparent as possible, and (b) we were free to speak about the results of the audit, and that the non-disclosure agreement only covered PyMetrics's source code, data, and internal manuals.

6.2 Future Work

The context surrounding PyMetrics afforded us the privilege of narrowly scoping our audit. Employment selection is a tightly regulated area, with clear compliance metrics that are relatively straightforward to operationalize, and that are supported a significant amount of case law [1, 33, 50]. That said, there are two ways that we could have expanded our audit, and these can be considered for future audits in this sector.

Our first open question concerns PyMetrics's choice to focus exclusively on Title VII of the Civil Rights Act, and more formally on disparate impact and disparate treatment in their fairness testing. The choice to operationalize the four-fifths rule potentially privileges regulatory compliance above other concepts of fairness such as differential validity [50]. PyMetrics does offer to customize their adverse impact testing to suit clients' needs, but this is not the same as fundamentally re-evaluating the default codebase and the guarantees it offers. It remains to be seen whether multiple fairness guarantees can be met in the context of MLHireInc's business while still preserving MLHireInc's careful compliance with existing U.S. federal regulations.

Our second open question that could be audited is the efficacy of PyMetrics's games at assessing the "fit" of job seekers. PyMetrics's games are based on peer-reviewed and replicated psychological studies, but drawing a direct line from laboratory experiments to real-world job performance is challenging. PyMetrics provided us with a confidential presentation containing results from game validation studies [43] — at a minimum we encourage PyMetrics to make these results public. That said, larger-scale, observational studies based on the longitudinal data PyMetrics collects from clients to evaluate model performance could be invaluable for assessing the efficacy of the games.

6.3 Cooperative Audits

We present this work as a case-study of a cooperative audit between industry stakeholders who wish to be transparent and academic researchers who want to improve the overall application of ML. As we have documented, cooperative audits require that industry partners engage transparently and give the auditors broad freedoms. For the company, this may be stress-inducing.

Likewise, every company operates within its own set of legal, regulatory, ethical, and proprietary limitations, and auditors operating in the cooperative mode should prepare to be flexible within those constraints. For example, PyMetrics expressed an interest in exploring intersectionality of gender and race demographic categories; however, intersectionality is not recognized by the relevant regulatory agencies. According to the methods in § 3, the ML objective function is to find the most performant, least biased model. A less performant model may be selected to meet the standard of fairness set by regulations. However, selecting a less performant model to meet standards *outside of regulation*, such as intersectional fairness, could create grounds for legal dispute. This is why all parties agreed that definitions of fairness and fairness for non-EEOC groups were outside the scope of this audit (see § 4.1). That said, in a different context, such as medicine or advertising, intersectionality could be crucial for model performance as well as fairness, and thus be fair-game auditors.

The transparency of ML service providers is also contextual. Companies must balance sharing enough proprietary information to be transparent with their users, clients, and watchdog groups, but not enough information as to be replicated by a competitor. A cooperative audit grants a compromise, so that independent experts can get full transparency without the company fearing a loss of IP, or jeopardizing the privacy of data subjects. Even then, the amount of transparency between PyMetrics and the auditors in this study

cannot be expected for all companies undergoing audits, which emphasizes the importance of agreeing upon scope in advance.

Many of the issues we discuss in this section come down to defining the scope of an audit. Auditors need to insist that industry partners make their criteria for substantive issues like fairness clear ahead-of-time. Without these benchmarks, it is difficult to define what the objectives or outcomes of a collaborative audit are. We argue that this audit was successful in no small part because PyMetrics had already adopted and documented business practices that could be objectively evaluated.

We have presented a case-study that, to the best of our knowledge, may be the first publicly-documented audit of algorithmic fairness between a willing private company and an external investigative team. As such, we had to navigate challenging questions around how to structure our audit with respect to scoping our research questions, accepting remuneration, maintaining the security of confidential source code and data, and preserving investigative secrecy, while simultaneously maintaining an arms-length, objective relationship with PyMetrics. We hope that this audit sets a new precedent for cooperative algorithm audits, and that this leads to more companies engaging independent experts to audit their sociotechnical systems in the future.

REFERENCES

[1] Ifeoma Ajunwa. 2020. The Paradox of Automation as Anti-Bias Intervention. *Cardozo, L. Rev.* 41 (2020).

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine Bias. ProPublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[3] Joshua Asplund, Motahhare Eslami, Hari Sundaram, Christian Sandvig, and Karrie Karahalios. 2020. Auditing Race and Gender Discrimination in Online Housing Markets. In *Proc of ICWSM*.

[4] Jack Bandy and Nicholas Diakopoulos. 2020. Auditing News Curation Systems: A Case Study Examining Algorithmic and Editorial Logic in Apple News. In *Proc of ICWSM*.

[5] Solon Barocas and Andrew D. Selbst. 2016. Big Data's Disparate Impact. *104 California Law Review* 671 (2016).

[6] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proc. of FAT**.

[7] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (April 2017), 183–186.

[8] Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. 2018. Investigating the Impact of Gender on Rank in Resume Search Engines. In *Proc. of CHI*.

[9] Le Chen, Alan Mislove, and Christo Wilson. 2015. Peeking Beneath the Hood of Uber. In *Proc. of IMC*.

[10] Le Chen, Alan Mislove, and Christo Wilson. 2016. An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace. In *Proc. of WWW*.

[11] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. 2018. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Proc. of FAT**.

[12] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20 (1995).

[13] Nicholas Diakopoulos. 2014. Algorithmic Accountability Reporting: on the Investigation of Black Boxes. Tow Center for Digital Journalism Brief.

[14] Nicholas Diakopoulos, Daniel Trielli, Jennifer Stark, and Sean Mussenden. 2018. I Vote For—How Search Informs Our Choice of Candidate. In *Digital Dominance: The Power of Google, Amazon, Facebook, and Apple*, M. Moore and D. Tambini (Eds.). 22.

[15] Danielle Ensign, Sorelle A. Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. 2018. Runaway Feedback Loops in Predictive Policing. In *Proc. of FAT**.

[16] Equal Employment Opportunity Commission, Civil Service Commission, et al. 1978. Uniform guidelines on employee selection procedures. *Federal Register* 43, 166 (1978), 38290–38315.

[17] Motahhare Eslami, Kristen Vaccaro, Karrie Karahalios, and Kevin Hamilton. 2017. "Be careful; things can be worse than they appear": Understanding Biased Algorithms and Users' Behavior around Them in Rating Platforms. In *Proc of ICWSM*.

[18] Motahhare Eslami, Kristen Vaccaro, Min Kyung Lee, Amit Elazari Bar On, Eric Gilbert, and Karrie Karahalios. 2019. User Attitudes towards Algorithmic Opacity and Transparency in Online Reviewing Platforms. In *Proc. of CHI*.

[19] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *Proc. of KDD*.

[20] Jessica Fjeld, Nele Achten, Hannah Hilligoss, Adam Nagy, and Madhulika Srikumar. 2020. Principled Artificial Intelligence: Mapping Consensus in Ethical and Rights-Based Approaches to Principles for AI. *Berkman Klein Center Research Publication* 2020, 1 (2020). https://ssrn.com/abstract=3518482

[21] International Organization for Standardization. 2012. ISO/IEC 27001 Information Security Management. http://iso.org/isoiec-27001-information-security.html.

[22] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2016. On the (im)possibility of fairness. *CoRR* abs/1609.07236 (2016).

[23] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth. 2019. A comparative study of fairness-enhancing interventions in machine learning. In *Proc. of FAT**.

[24] Batya Friedman and Helen Nissenbaum. 1996. Bias in Computer Systems. *ACM Trans. Inf. Syst.* 14, 3 (July 1996), 330–347.

[25] Aniko Hannak, Piotr Sapieżyński, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. 2013. Measuring Personalization of Web Search. In *Proc. of WWW*.

[26] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. 2014. Measuring Price Discrimination and Steering on E-commerce Web Sites. In *Proc. of IMC*.

[27] Anikó Hannák, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. Bias in Online Freelance Marketplaces: Evidence from TaskRabbit and Fiverr. In *Proc of CSCW*.

[28] Desheng Hu, Shan Jiang, Ronald E. Robertson, and Christo Wilson. 2019. Auditing the Partisanship of Google Search Snippets. In *Proc. of WWW*.

[29] Eslam Hussein, Prerna Juneja, and Tanushree Mitra. 2020. Measuring Misinformation in Video Search Platforms: An Audit Study on YouTube. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1 (May 2020).

[30] Shan Jiang, Le Chen, Alan Mislove, and Christo Wilson. 2018. On Ridesharing Competition and Accessibility: Evidence from Uber, Lyft, and Taxi. In *Proc. of WWW*.

[31] Anna Kawakami, Khonzoda Umarova, Dongchen Huang, and Eni Mustafaraj. 2020. The 'Fairness Doctrine' Lives on? Theorizing about the Algorithmic News Curation of Google's Top Stories. In *Proc. of HT*.

[32] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. 2015. Unequal Representation and Gender Stereotypes in Image Search Results for Occupations. In *Proc. of CHI*.

[33] Pauline T. Kim. 2017. Data-Driven Discrimination at Work. *William & Mary Law Review* 58 (2017).

[34] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent Trade-Offs in the Fair Determination of Risk Scores. *CoRR* abs/1609.05807 (2016).

[35] Chloe Kliman-Silver, Aniko Hannak, David Lazer, Christo Wilson, and Alan Mislove. 2015. Location, Location, Location: The Impact of Geolocation on Web Search Personalization. In *Proc. of IMC*.

[36] Juhi Kulshrestha, Motahhare Eslami, Johnnatan Messias, Muhammad Bilal Zafar, Saptarshi Ghosh, Krishna P. Gummadi, and Karrie Karahalios. 2017. Quantifying Search Bias: Investigating Sources of Bias for Political Searches in Social Media. In *Proc of CSCW*.

[37] Peter Lee. 2016. Learning from Tay's Introduction. Official Microsoft Blog. https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/.

[38] Zachary Lipton, Julian McAuley, and Alexandra Chouldechova. 2018. Does mitigating ML's impact disparity require treatment disparity?. In *Proc. of NeurIPS*.

[39] Kristian Lum and William Isaac. 2016. To predict and serve? *Significance* 13, 5 (2016).

[40] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proc. of NIPS*.

[41] Emma Lurie and Eni Mustafaraj. 2018. Investigating the Effects of Google's Search Engine Result Page in Evaluating the Credibility of Online News Sources. In *Proc. of WebSci*.

[42] MLHireInc. 2019. [Confidential] Fairness Testing Procedures.

[43] MLHireInc. 2019. [Confidential] Games, Measures and Factors: Measurement Validity.

[44] MLHireInc. 2019. [Confidential] Job Analysis Methods & Process.

[45] MLHireInc. 2019. [Confidential] Technical Brief for MLHireInc.

[46] MLHireInc. 2020. Anonymous Bias Testing Framework. Open Source Repository.

[47] MLHireInc. 2020. [Confidential] Demographic Disclosure Study.

[48] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware Data Mining. In *Proc. of KDD*.

[49] Lincoln Quillian, Devah Pager, Ole Hexel, and Arnfinn H. Midtbøen. 2017. Meta-analysis of field experiments shows no change in racial discrimination in hiring over time. *Proceedings of the National Academy of Sciences* 114, 41 (2017), 10870–10875.

[50] Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. 2020. Mitigating Bias in Algorithmic Hiring: Evaluating Claims and Practices. In *Proc. of FAT**.

[51] Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. 2020. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In *Proc. of FAT**.

[52] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. of KDD*.

[53] Ronald E Robertson, Shan Jiang, Kenneth Joseph, Lisa Friedland, David Lazer, and Christo Wilson. 2018. Auditing Partisan Audience Bias within Google Search. *Proceedings of the ACM: Human-Computer Interaction* 2, CSCW (November 2018).

[54] Ronald E. Robertson, Shan Jiang, David Lazer, and Christo Wilson. 2019. Auditing Autocomplete: Recursive Algorithm Interrogation and Suggestion Networks. In *Proc. of WebSci*.

[55] Ronald E Robertson, David Lazer, and Christo Wilson. 2018. Auditing the Personalization and Composition of Politically-Related Search Engine Results Pages. In *Proc. of WWW*.

[56] Christian Sandvig, Kevin Hamilton, Karrie Karahalios, and Cedric Langbort. 2014. Auditing Algorithms: Research Methods for Detecting Discrimination on Internet Platforms. In *In Prov. of Data and Discrimination: Converting Critical Concerns into Productive Inquiry, a preconference at the Annual Meeting of the International Communication Association*.

[57] Gary Soeller, Karrie Karahalios, Christian Sandvig, and Christo Wilson. 2016. MapWatch: Detecting and Monitoring International Border Personalization on Online Maps. In *Proc. of WWW*.

[58] Latanya Sweeney. 2013. Discrimination in Online Ad Delivery. *ACM Queue* 11, 3 (April 2013).

[59] U.S. Congress. 1964. Civil Rights Act.

[60] Giridhari Venkatadri, Yabing Liu, Athanasios Andreou, Oana Goga, Patrick Loiseau, Alan Mislove, and Krishna P. Gummadi. 2018. Privacy Risks with Facebook's PII-based Targeting: Auditing a Data Broker's Advertising Interface. In *Proc. of IEEE Symposium on Security and Privacy*.

[61] Giridhari Venkatadri, Elena Lucherini, Piotr Sapiezyński, and Alan Mislove. 2019. Investigating sources of PII used in Facebook's targeted advertising. In *Proc. of PETS*.

[62] James Vincent. 2018. These stickers make computer vision software hallucinate things that aren't there. The Verge. https://www.theverge.com/2018/1/3/16844842/ai-computer-vision-trick-adversarial-patches-google.

[63] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *Proc. of International Conference on Artificial Intelligence and Statistics*.