Biased Programmers? Or Biased Data?
A Field Experiment in Operationalizing AI Ethics[*]

Bo Cowgill and Fabrizio Dell'Acqua
Graduate School of Business

Samuel Deng, Daniel Hsu, Nakul Verma and Augustin Chaintreau
Department of Computer Science

Columbia University

December 13, 2020

Abstract

Why do biased predictions arise? What interventions can prevent them? We evaluate 8.2 million algorithmic predictions of math performance from $\approx$ 400 AI engineers, each of whom developed an algorithm under a randomly assigned experimental condition. Our treatment arms modified programmers' incentives, training data, awareness, and/or technical knowledge of AI ethics. We then assess out-of-sample predictions from their algorithms using randomized audit manipulations of algorithm inputs and ground-truth math performance for 20K subjects. We find that biased predictions are mostly caused by biased training data. However, one-third of the benefit of better training data comes through a novel economic mechanism: Engineers exert greater effort and are more responsive to incentives when given better training data. We also assess how performance varies with programmers' demographic characteristics and their performance on a psychological test of implicit bias (IAT) concerning gender and careers. We find no evidence that female, minority and low-IAT engineers exhibit lower bias or discrimination in their code. However, we do find that prediction errors are correlated within demographic groups, which creates performance improvements through cross-demographic averaging. Finally, we quantify the benefits and tradeoffs of practical managerial or policy interventions such as technical advice, simple reminders, and improved incentives for decreasing algorithmic bias.

[*] The authors thank seminar participants at the NBER *Economics of AI* Conference, Harvard Business School, New York University Stern School of Business, the MIT Conference on Digital Experimentation (CODE), American Economic Association Annual meeting session on *Algorithmic Fairness and Bias*, EC'20 (ACM's *Conference on Economics and Computation*, and the Chicago Conference on Advances with Field Experiments. We also thank the Kauffman Foundation, the National Bureau for Economic Research, the Sloan Foundation, and the Richman Center for Business, Law, and Public Policy for supporting this research. In addition we thank Erica Bailey, Erik Brynjolfsson, Elizabeth Campbell, James Carter, Matthew Diabes, Jorge Guzman, Nan Jia, Hyunjin Kim, Rem Koning, Robert Seamans, Catherine Tucker and Mike Yeomans for helpful discussions.

1 Introduction

Why do biased predictions arise? Across a wide variety of theoretical models, biased predictions are responsible for segregation and outcome disparities in settings that include labor markets, criminal justice, and advertising. Although many classic theoretical models of behavior feature decision-makers with accurate (if discriminating) statistical predictors (Phelps, 1972; Arrow, 1973), empirical evidence often shows that predictions are systematically inaccurate in practice (Bohren et al., 2019). How do these biased predictions arise? What theoretical mechanisms produce them, and what practical interventions can reduce prediction bias?

In this paper, we address these questions through a field experiment applying machine learning to predict workers' performance. Automated hiring is a leading concern of policymakers questioning the ethics and fairness of AI systems. Research and public discourse on this topic have grown enormously in the past five years along with programs to raise awareness or introduce ethical or technical training about bias. However, few studies have attempted to evaluate, audit, or learn from these interventions or connect them back to theory. This paper aims to step in that direction.

We examine the formation of biased predictions in a field experiment in the development of AI technology. Our subjects are approximately 400 machine learning engineers. Our experiment gives us a direct view of prediction technology while it is being assembled. This setting creates measurement opportunities that would be impossible for learning processes in other economic settings, allowing us to study the mechanisms behind biased learning more directly. Through randomized treatments, we show how policy or managerial interventions can change the formation of biased predictions.

We find that biased predictions are mostly caused by biased training data. However, simple reminders about bias are almost half as effective as fully de-biasing training data. We find mixed results on technical education and programmer demographics. Programmers who understand technical guidance successfully reduce bias. However, many do not follow the advice, resulting in algorithms that are *worse* than programmers given a simple reminder.

Algorithmic predictions by female and minority AI programmers do not exhibit less algorithmic bias or discrimination. However prediction errors are correlated within engineering demographics, creating bias reductions from cross-demographic averaging of predictions. We also find no effects of our incentive treatments and no evidence that programmers' implicit associations between gender and math (measured through an IAT) are correlated with bias in code.

The remainder of this paper proceeds as follows. We review related literature in Section 2. Section 3 presents a decision-making model that clarifies our framework. Section 4 describes the empirical setting and subjects in our field experiments. Section 5 describes the data and programming task given to all engineers. Section 6 describes our pre-registered experimental design and randomized interventions.[1] Section 8 covers results, and Section 9 offers concluding thoughts.

[1] The study was pre-registered in the AEA RCT Registry, RCT ID:AEARCTR-0003574

2 Related Literature

Our paper is related to the formation of biased predictions. Most research in this area examines belief formation inside a single person's mind (Bordalo et al., 2016; Bohren et al., 2019). However, predictions arise in many ways throughout the economy. Predictions arise through group processes such as prediction markets (Wolfers and Zitzewitz, 2004) or polls of expert opinions (Tetlock, 2017). Predictions are also often the output of statistical technology (Kleinberg et al., 2018). Inaccurate predictions about the value of mortgage-backed securities were reportedly the product of mathematical models.

Group- or statistical- forecasting methods can never be entirely separated from human minds. Statistical algorithms are tools developed by human beings who may bring their own agenda and context to the task. Activists have expressed widespread concern that machine learning simply exists to apply the veneer of science to glorify pre-existing "opinions embedded in code" (O'Neil, 2017). Famously, US News and World Report rejected an early statistical approach to college rankings because the results did not conform to the public's images of great universities. Development proceeded until a new statistical model placed Ivy League Schools at the top (Thompson, 2000).

The interplay between a statistical forecaster's motives, background, and prediction task is a motivating aspect of our study. It is also the source of extensive public debate around the solution to biased algorithms. "Biased programmers" is a leading hypothesis for algorithmic bias mentioned in two literature reviews in economics (Parkes et al., 2019; Cowgill and Tucker, 2019). A policy report by the AI Now Institute writes that "[T]ackling the challenges of bias within technical systems requires addressing workforce diversity."

2.1 Empirical Methods for Studying Discrimination

Our paper is also related to empirical methodology for measuring discrimination, and particularly experimental methods Bertrand and Duflo (2017); Neumark (2018). As a growing number of decisions are made through algorithms, this is an important topic of study in its own right. However, our methodological approach addresses several limitations that arise in measuring discrimination in any context.

Efforts to measure discrimination broadly fall into two strategies, each with strengths and weaknesses. The first approach, sometimes called "outcomes tests," examines data about individual subjects' performance metrics. Examples of relevant outcomes include workers' job performance or the criminal behavior of suspects. Researchers compare choices by a potential discriminator[2] with performance measures. If data suggest that some subjects receive adverse treatment correlated with sensitive demographics, but *not* justified by his/her performance, this is interpreted as evidence of discrimination (particularly on the margin, Ayres, 2002).

The "outcome tests" approach suffers from two weaknesses. First, objective outcome measures may be difficult to obtain.[3] Second, even when objective data is broadly available, sensitive demo-

[2] For example, choices about hiring and wages in an employment context; or choices about sentencing and bail in a criminal justice context.

[3] Many organizations do not rigorously collect outcome data. In addition, subjective evaluation is used in many

graphic variables are often correlated with other characteristics, some of which may be correlated with performance metrics. For example, educational attainment could be correlated with both job performance and race. These confounds raise the possibility that demographic differences are driven at least partly by performance optimization.

To address confounds, researchers developed a second broad approach. In *audit and correspondence studies*, researchers ask potential discriminators to evaluate fake identities, often without revealing the pretense (Gaddis, 2018).[4] The approach avoids confounds by randomly modifying applicant demographics and holding other characteristics constant. Demographic characteristics are often suggested via distinctive names.

However, audit methods exhibit many shortcomings as well. The use of fake identities eliminates the possibility of validating decisions with real-world outcome data. Without real outcome data, inferences about performance metrics rely on the assumption that someone with a different name would perform equally. However, as Bertrand and Duflo (2017) noted, characteristics "beyond those intended by the researcher" may be inferred from audit manipulations. In other words, names may carry performance-related information.[5]

Audit studies lack *additional* crucial data. Employer preferences are measured by a crude "callback" decision. As other researchers have noted (Cowgill and Perkowski, 2019; Kessler et al., 2019), callback decisions may be affected not only by employer preferences, but also about employer's second-order beliefs about *candidate* preferences. Because callbacks are costly, employers may be reluctant to contact candidates believed to be unlikely to accept a job offer. Even as measures of employer preferences, callbacks are coarse variables that pixelate the magnitude of preferences into single bits. Lastly, research ethicists often critique audit methods for wasting the attention of subjects under false pretenses (Pager, 2007).

Both the "outcomes" and "audit" approaches feature at least one additional shortcoming. Researchers are interested not only in measuring existing discrimination. They are also interested in measuring *changes* in discrimination induced by policy interventions. Evaluating policies that affect discrimination is also a causal inference problem, requiring clean variation in the new policy. Such variation is typically scarce, because the firms who adopt new policies are non-randomly self-selected.[6]

settings to measure performance. In criminal justice settings, "re-arrest" data could be tainted by police's subjective decisions to surveil some citizens more than others. That is, in some settings, outcomes (re-arrest) may be affected by the initial presence of discrimination upstream (Cowgill, 2017). These problems could taint any objective outcome metrics. Team production also complicates individual performance metrics. Many important jobs require collaborations in which individual contributions may not be easily recorded. Even if they were recorded, the contribution of an individual to a team outcome is a counterfactual question requiring clean variation that is typically missing. Even when objective evaluations are possible for teams or individuals, it is typically available only for a non-representative subset of the population, for example, hired workers. Missing data about the productivity of unhired workers frustrates the study of hiring bias.

[4] The earliest audit studies used actors posing as in-person applicants for jobs or housing (Daniel, 1968). More modern audit studies typically send fake resumes or inquiries to potential discriminators (Bertrand and Mullainathan, 2004).

[5] For example, Bertrand and Duflo (2017) wrote that a distinctive name may be "a political statement by the parent, accompanied by a different attitude towards schooling and obedience."

[6] Some researchers have timed audit experiments to overlap with state-level policy changes (Agan and Starr, 2017).

2.2 Our methodological approach

Our experimental framework addresses many of the above shortcomings. Our paper features high-quality performance metrics about two groups of subjects. The first is the programmers, and the second is the OECD test subjects. Performance for both groups can be measured objectively, avoiding the typical pitfalls of "outcome tests" such as subjective performance criteria.

For the math subjects, outcomes are available for a professionally-weighted sample of the entire OECD population, and *not* a limited, non-random subset. Regarding engineers, predictive outcomes are available for every engineer in our sample.[7] For our engineers, we also have natural ways to aggregate individual performance to estimate team contributions. We can measure how correlated each engineer's model is with potential teammates, and how well a simple average (or more complicated aggregation) of two engineers perform as an ensemble.

Our design also features an audit-like manipulation of algorithmic inputs. As part of our test data, we ask engineers to evaluate candidates whose characteristics are identical, except that a single covariate (gender) has changed. We can then compare the predicted outcomes of identical candidates, who differ only on their gender. Because the gender of the OECD subjects is accessible to the engineers' programs, we can avoid the issues raised by manipulating first names.

Our outcomes are far richer than a typical audit study. In place of binary callback decisions, we obtain a continuous measure of predicted math performance that spans the entire spectrum. Additionally, our methods avoid the ethical issues introduced by fake resumes. Although our design did involve some misdirection, our experimental subjects' time was not wasted without compensation.[8]

Because our design features randomization both on screeners (programmers) and on candidates (subjects evaluated by the resume), our design resembles the two-sided audit (Cowgill and Perkowski, 2019; Agan et al., 2019). In these experiments, researchers hire professional recruiters to select resumes under experimental conditions. The recruiters are then asked to evaluate job applications with audit-like manipulations of names and characteristics. Our design is essentially identical, but we use *software engineers* to build algorithms for decision-making, rather than human screeners.

Our study relates to existing empirical methods through one final mechanism. Existing audit papers may, in fact, be collecting data from algorithmic resume screeners rather than from human recruiters. The mechanisms in our study may directly relate to pre-existing audit studies that (on the surface) have nothing to do with algorithms.

As resume audit studies have exploded in popularity in the last two decades, employers' use of algorithms to evaluate job candidates has simultaneously exploded (Cowgill, 2019; Hoffman et al., 2015; Bogen and Rieke, 2018), especially through the rise of outsourced algorithmic screening. The most common use of algorithmic job assessments is to shortlist candidates for interviews; i.e., the "callback" behavior monitored by audit studies. Audit researchers have no way of knowing

[7] Although we do not have a representative sample of the entire global population of engineers, we do at least have outcomes for all participants in our study. We discuss the strengths and weaknesses of our sample in Section 4

[8] Our freelancer subjects were paid in dollars. Our bootcamp subjects participated as part of a program teaching technical skills and providing certification and credentialing.

whether a human is responsible for employer callback decisions in their study – or if these decisions are made by hiring algorithms like those in this paper.

3 Theoretical Framework

This paper is about biased predictions. However, there is more to decision-making than prediction alone. Below is a simple stylized model of decision-making which we use to clarify how machine learning in our context relates to decision-making outcomes. After showing a generic version, we adapt the particulars to the setting of human capital and hiring (the specific context of our experiment).

A decision-maker chooses an action $a \in A$ to maximize expected utility (u) subject to a probability distribution $\hat{p}(x)$ of the states of the world $x \in X$:

$$a^{\star} = \arg\max_{a \in A} \int_{x \in X} u(a|x)\hat{p}(x)dx \tag{1}$$

This framework for decision-making appears in other papers (DellaVigna, 2009; Agrawal et al., 2019). Notice the clean separation in the above expression between i) "u," the highly subjective preferences about which there are no correct answers, and ii) "$\hat{p}$," or beliefs about the objective state of the world which could in principle be verified.

Machine learning could appear in any part of Equation 1. The machine could entirely automate decisions. Alternatively, it could instead provide only one part of Equation 1. For example, machine learning could provide the technology for searching the action space A for the optimal action ($a^{\star}$) (possibly using inputs from other sources).

Alternatively, machine learning could provide the predictions ($\hat{p}$) part of Equation 1 (possibly leaving other parts to other agents). Our paper and experiment are motivated by this particular application. Agrawal et al. (2019) similarly and explicitly portrays AI advances as improvements specifically in prediction technology. As we show below, less biased predictions are useful for a variety of preferences (u).

In the context of human capital, each decision is about a candidate's job application. A recruiter's action space A is binary – either hire or not. In this setup, the "possible states of the world" (X) refer to the candidate's hidden or hard-to-measure characteristics, such as their on-the-job performance ability (our paper focuses on math skills). $\hat{p}(x)$ refers to the decisionmaker's best guess about those characteristics.

Of course, most employers do not aim to strictly maximize one dimension of performance such as math – or even overall performance. Employers may care about a bundle of skills and may also care about diversity and team performance. They may also care about the probability the candidate accepts the offer. The utility function u is where the employer would express tradeoffs between math skills, other abilities, diversity, and other hiring goals.

Note that more accurate prediction technology is useful to maximizing Equation 1 for a wide variety of preferences u. Even if an employer exhibited explicit taste-based preferences for (or against) one group, accurately judging performance helps the decision-maker navigate tradeoffs between performance and taste-based prejudicial payoffs. One frequent concern is that predictions may be biased against minorities and women; beliefs about minority and female skills are lower than their true levels. If this were the case, then correcting these biased predictions would lead to greater minority and female representation for a wide variety of utilities u (i.e., even if they cared about more than maximizing math ability). For this reason, de-biasing predictions is useful for improving representation.

4 Empirical Setting

We conducted our experiment in two empirical settings. Approximately 80% of our subjects were participants in a machine learning bootcamp at a large research university. The bootcamp taught machine learning programming techniques at a CS masters or advanced undergraduate level. Approximately half were undergraduates, and half were MS or Ph.D. students. We conducted our experiment in this setting twice; once in the Spring and once in the Fall.

These are attractive research subjects for our topic. Students from this program are frequently hired to work at large Internet companies such as Facebook and Google; the algorithms they will develop in the future may plausibly affect billions of Internet users. At the time of the experiment, 31% of the bootcamp participants had already been employed by a company that is a household name. The programming task we studied was a graded homework assignment, and performance was highly incentivized and competitive. Although this setting has many attractive qualities for our experiment, it required some design adaptations to address the possibility of cheating and contamination between treatment groups (discussed in Section 5).

Over half the bootcamp participants already graduated. The average participant had 1.2 years of work experience (median 0.67 years) at the time of the experiment. To complement these relatively inexperienced subjects, we sought a population of programmers to complement the students: Freelance machine learning engineers. Using an online platform, we recruited 60 machine learning engineers (20% of total subjects).[9] The average contractor in our study was more experienced than our bootcamp programmers, worked 3.89 years before our experiment (median 3.16 years).

Table 1 summarizes the characteristics of both sets of engineers in our experiment. Our engineers are 71% male, 29% female, 28% White, 52% East Asian, 15% South Asian, and only about 5% Black or Latino/a/x. Like the broader population of engineers, our subject population lacks diversity in key characteristics. However, our sample is slightly more diverse than the US software engineering population as a whole.

[9] We created a private job listing on the platform, then randomly invited subjects from the machine learning section of the platform. We restricted our invitations to engineers who had at least 1.5 years of work experience, were based inside the United States, and whose rates were between $20 and $80.

5 Training Data and Prediction Task

All subjects in our experiment were assigned the same task: Develop an algorithm for predicting individual math performance from biographical features and apply it to 20,000 new individuals who do not appear in the training data. This task allows us to study a fundamental mechanism behind biased performance predictions.[10]

Math literacy is an attractiveideal topic for studying algorithmic bias. Compared with other skills taught in educational settings, math is relatively useful outside of classrooms. Math ability is correlated with income within and across countries and is an important gatekeeper for many high-wage jobs (finance, management, and engineering). Importantly, math questions have objectively correct answers that span cultural backgrounds.

Math performance is also the topic of common stereotyping suggesting that women are worse at math than men. Additionally, women across world regions pursue math-related careers less than men. For example, in 2015, women were only 28.8% of scientific researchers in the world.[11] This stereotype introduces the possibility of algorithmic bias if historical data is used carelessly.

PIAAC Data. Math is also an attractive topic because of the availability of high-quality training data. High-quality data about math literacy is collected by the OECD, an intergovernmental economic organization of mostly high-income countries.

Numeracy is measured internationally on a common scale by experts in OECD's *Programme for the International Assessment of Adult Competencies* (PIAAC, Schleicher, 2008). The PIAAC data contains quantitative measures of math and other skills for a representative sample of the working-age population in 24 countries. It is the canonical dataset for cross-country and within-country comparisons of numeracy and skills. We use the PIAAC data as training data for our engineering subjects.

Within the PIAAC dataset, math ability is coded on a scale of roughly (0,5000), with most values falling between 1640 and 3200. Higher values indicate better math skills. Our engineers' instructions stated these represented ground truth performance data about the candidates.

To our knowledge, this is the first paper to utilize the PIAAC dataset in computer science research about algorithms. To facilitate other researchers utilizing this dataset, we have made a cleaned and merged copy with documentation available on dataverse.org.[12] The PIAAC data contains measures of reading, critical thinking, and other skills that could be used for training, comparing, or validating algorithms; we have included these measures as well in our cleaned/merged copy.

The PIAAC data exhibits characteristics that solve several critical research challenges for algorithmic bias researchers. The data is large and extremely rich; it contains over 200,000 citizen observations and over 500 covariates per citizen, including the citizens' demographic, educational, employment, and other background characteristics. Most of these 500 features are categorical vari-

[10] There might be other biases in algorithms besides prediction bias which we do not discuss in this paper.

[11] UNESCO Institute for Statistics, Fact Sheet No. 51: Women in Science (2018). http://uis.unesco.org/sites/default/files/documents/fs51-women-in-science-2018-en.pdf, accessed August 20, 2019.

[12] https://doi.org/10.7910/DVN/JAJ3CP.

ables; when these are converted into binary variables, there are over 5,000 features per individual. The data is extensively documented by the OECD; our repository contains some additional documentation helpful for researchers in our field.

Importantly, PIAAC measures math ability (and other skills) *for people not employed in math-related jobs*. It achieves this by administering a short math exam to all participants. The participants recruited for participation are carefully selected by professional statisticians to be representative of the broader adult population in each OECD country.[13]

Because of the careful attention to representation, researchers using PIAAC can significantly avoid the problems of sample selection behind many instances of algorithmic bias. This is a very rare characteristic of any dataset. We believe the PIAAC dataset represents is an upper bound of data quality. We are aware of no private company that has collected training data with as much attention to representativeness as shown by PIAAC.

By contrast, many datasets used in machine learning are "datasets of convenience" or "digital exhaust" from a separate, unrepresentative process. It has typically *not* been curated for research. For example, the hiring algorithm in Hoffman et al. (2017) was based on training data on employees only, but was applied to make assessments about all job applicants, including ones very different than the firms' historical set of employees.[14]

Although many researchers have highlighted this problem, it has been impossible to quantify how much "datasets of convenience" contribute to algorithmic bias. The counterfactual algorithms – machine learning trained on representative data – typically are not created or observed, because the representative data has not been collected or utilized. Researchers therefore cannot measure the size of this effect.

By using the PIAAC data, our paper can measure the "datasets of convenience" effect by experimentally altering whether engineers receive high-quality, representative training datasets, or a digital exhaust convenience sample. Each dataset contains the same features, but a different sample.

Machine Learning Task, Evaluation and Incentives. The engineers in our experiment are given a sample of PIAAC data to use for training data in their algorithms. All engineers were asked to predict math performance using biographical features. Evaluation was performed on a test set of 20,000 observations randomly selected from the PIAAC data, but left out of their training data. The observations used for evaluation were drawn from a representative distribution of the entire dataset. All subjects in all treatment groups were asked to make predictions about the same set of 20K observations. Programmers could use the programming environment of their choice, and any open-source library.

The engineers were asked to minimize the mean squared error (MSE). We did not ask for any additional diversity considerations. Minimizing MSE contained implicit fairness considerations insofar as it discouraged engineers to avoid overstating the abilities of any particular group, but

[13] This attention to representativeness contrasts with other datasets often used in studies of algorithmic bias. For example, in the widely-used COMPAS dataset (Larson et al., 2016), arrest outcomes are available *not* for the population at large, but only for citizens who had been arrested within 2013-2014 in Broward County. COMPAS has about 80 features per individual (vs 5K features for PIAAC).

[14] Cowgill (2019) contains a theory model about when and why this may (or may not) create problems.

nothing beyond maximizing predictive accuracy. The instructions made clear they were evaluated on mean squared error alone (and explained in detail how this is calculated).

To incentivize performance, the engineers were given MSE benchmarks. A lower benchmark was labeled a "minimum acceptable" performance. A second, higher MSE benchmark was labeled a "target." Students were told that meeting these thresholds would improve their grades. Contractors were told this performance would trigger a monetary bonus.

Programmers were also given incentives through a "slope." For engineers who reached the target MSE, every one unit decrease would trigger additional rewards in the amount of the slope. For students, these rewards were extra credit. For contract programmers, these rewards were additional dollars.

Cheating and Obfuscation. Because the PIAAC dataset is publicly available, programmers in our experiment could have (in theory) cheated by seeking and sending in the correct answers. We took two measures to avoid this. First, programmers were led to believe that the data originated from an employer whose goal was to develop an algorithm to help screen and select workers. The programmers were told that the jobs in question required numeracy skills and, crucially, that performance in the job tasks was quantifiable. To improve the realism of this cover story, we removed about 100 features from the PIAAC data that are typically not observable at the time of hiring.

In addition, we substantially obfuscated the PIAAC variable names and descriptions so that they could not be easily queried. We are aware of no instances of cheating, and our results suggest it was unlikely: Only about 25% of participants met even the minimal performance thresholds we sought, and none achieved their targets MSEs. In addition, we administered a survey at the end of the experiment, offering confidentiality to respondents. We asked each programmer if he/she could guess the name of our research partner providing the data. 83% responded "No idea." Of the remaining 16%, none guessed the OECD or PIAAC.

Contamination. An additional concern is that subjects may contaminate each other. This may be particularly problematic in the classroom setting, where students regularly interact. For example, the students in the "Technical Education" arm of the experiment could share techniques with students in the control group.

We took several measures to avoid such cooperation. For our contracted engineers, the platform blocks programmers from identifying other freelancers working on the same job. As a result, they would have no way to know how to share information with other participants. We suggested to each contract programmer that he/she was the only person working on the problem. In addition, their contract was covered by an NDA that would forbid discussion with a third party.

We took additional steps to prevent collusion among the students. Each student was told that he/she had a slightly different version of the company's hiring problem and that the answers were not supposed to be identical. Although all subjects were making predictions about the same 20,000 new observations, the test set was labeled with subject-specific row identifiers; this discouraged subjects from merging or cross-referencing data. The training and test sets were also placed in differing random orders so that students browsing the data casually would see different data. We added noise and/or constants to some variables so that the raw data was different in superficial ways.

For students, we also created graphically different instruction documents for each student featuring differentiated fonts and page breaks. Like our other modifications, this creates the superficial appearance of different assignments and frustrates efforts to compare assignments. The goal of these modifications is to deter efforts to collaborate by obstructing the collaboration process.

Next, we took additional steps through our distribution strategies. Each student was sent a customized link to his/her materials that was accessible only through his/her password-protected university account. The instructions – including the "technical education" white paper available to one treatment arm – was visible only online and was blocked from printing or downloading. We were able to monitor who attempted to download another student's materials through their own account. We observed no instances of students accessing each others' links, including the "technical education" white paper.[15]

Finally, the instructors of the bootcamp explicitly forbade collaboration or discussion about the assignment. Of course, it is possible that students collaborated despite these efforts. Through the instructors, we have access to data about the study groups formed by students for group assignments. These study groups were self-organized by the students and were highly segregated according to gender, race, age, and college major. Study-group peers would be natural counterpoints for discussion and collaboration. However, we find little correlation among the key outcomes for students within the same study groups.

There may have been additional forms of collaboration that we have not observed. Importantly, *any contamination or collaboration should attenuate treatment differences*. Collaboration should push the control group and treatment arms towards each other, resulting in zero differences. Despite this tendency, we document several differences across treatment arms in this paper. Because of the attenuating effect of unobserved collaboration, our results may in fact be lower bounds for true effects. The "true" effects (absent collusion) may be much stronger.

6 Experimental Conditions

Control Condition. Engineers in this condition received a dataset containing realistic sample-selection bias. For training data, many companies turn to "datasets of convenience" which are not highly representative of the target population. For the purposes of a hiring algorithm, companies enjoy convenient access to datasets of existing workers or job applicants in their own company. Many companies actually use this data to create job screening algorithms (see Cowgill (2019); Hoffman et al. (2015); Bogen and Rieke (2018)), even though the workers in their own firms are not representative of the job applicants they are screening.

Our aim with this control group is to mimic this process using the PIAAC data. We, therefore, gave our control group a "dataset of convenience"' consisting of workers who are *already employed in math-intensive jobs*. The PIAAC dataset contains identifiers for these workers.[16] Throughout this

[15] Such efforts would have been blocked by the account privacy settings of the homework website anyway.

[16] To define a math-intensive job, PIAAC combined answers to six questions about numeracy skill use at work and created a "Use of Numeracy skills at work" index. This index is bucketed into quintiles. Our control group received

paper, we will refer to these subjects in math-intensive jobs as "math workers."

Programmers in the control condition received 20,000 training observations randomly selected from PIAAC's math worker subsample, thus, we label this group as receiving "biased training data.". This simulates a real-world situation in which a firm has performance outcomes *only* for workers it hired historically, possibly in a biased or unrepresentative way. The math performance of this group was measured identically to everyone else in the PIAAC dataset (through PIAAC's brief exam).

Our engineers in this condition were explicitly informed that their data was a biased sample. They were specifically notified that their training data came from historically hired workers (and not from a more representative sample). However, this disclosure was made in a clinical way. We did not elaborate on the implications for misclassification, discrimination, or social outcomes.

In theory, enlightened engineers could have realized this issue and taken efforts to address it. Even if they had not, a sufficiently sophisticated model may be able to predict out of sample successfully. The control group partly helps us assess whether this outcome was is realistic in our setting.

How unrepresentative was the control group's training data? Table 2 compares the full PIAAC population with the population of PIAAC math workers. These populations differ along several important dimensions. For example, while the full OECD population is almost evenly divided between men and women, the percentage of men in math jobs is 62.5%. Math workers also have a distinct age and educational profile.

The population of math workers is very different from the full population. By receiving this sample of math workers, the control group received a dataset with relevant, and realistic, sample-selection bias. Table 2 shows that the unrepresentative data is truly, highly unrepresentative. This raises questions about whether it is realistic at all to forecast about the larger population. To assess this, we build an empirical model of which workers are labeled math workers, and we apply it to all workers in both the biased and unbiased training data.

We visualize the results in Figure 1. Although the math sample is much more likely to go into math, many workers enter math jobs despite having a low ex-ante probability. In fact, for any predicted probability of being labeled math workers, both biased and unbiased training data provide some observations. That is, although math workers are a highly unrepresentative group, at any point, there is some overlap in the characteristics between the two samples. Subjects who received the biased sample could make full use of the data available to them, and exploit this overlap to produce their predictions about the full sample.

Unbiased Data. In this condition, participants received a similar dataset as above. However, this dataset was free of sample-selection bias. Programmers received 20,000 observations randomly selected from the entire population. This training data is unbiased, as it is a completely representative subset of the PIAAC data, constructed to resemble the OECD population as a whole. That

random samples from the top quintile. The questions making up the index are: "How often do you calculate costs or budgets?", "How often do you use or calculate fractions or percentages?", "How often do you use a calculator?", How often do you prepare charts or graphs or tables?", "How often do you use simple algebra or formulas?", "How often do you use advanced math or statistics?"

is, the programmers have example outcomes for training on a truly representative group. This allowed programmers to utilize unbiased training data in their algorithms. Importantly, engineers were told their data was unbiased.

Reminder. The third condition received the same unrepresentative training data as the control group, and exactly the same information, but with the addition of a reminder that the training data is unrepresentative. We report the reminder below,

> *Note: As a reminder, our recruiters' expectations about who to hire may have been systematically wrong. Performance data is available only for workers like the ones we've hired in the past. This may reflect our recruiters' stereotypes or biases about who is good at math.*
>
> *As you write your algorithm, please be mindful that your training dataset may originate in a* biased social system*. Adjusting your algorithm to account for discrimination in hiring, self-sorting of applicants, or other sources of such bias could improve your accuracy on the test set. You will be evaluated only on the accuracy of your predictions on the test set.*
>
> *Please note: For the workers we've hired in the past, we are confident their performance outcomes in your training dataset have been labeled accurately. Performance is based on math and there are objective measures of quality. However, you only have this data about the candidates we previously hired.*

Technical Education. The fourth condition received the same training data and the same reminder that the training data is unrepresentative as the third condition, *as well as* some technical training about avoiding algorithmic bias from unrepresentative training datasets.

Programmers in this condition received an additional line in their instructions. This appeared after the message in the reminder saying, "Adjusting your algorithm to account for discrimination [...] could improve your accuracy on the test set."

> *Here [link] is a white paper containing an overview, some citations and instructions about this kind of adjustment. If the white paper doesn't help, feel free to ignore it. You will be evaluated only in terms of the accuracy of your predictions on the test set.*

The white paper contained simple, jargon-free information about sample selection correction. It contained a brief theoretical explanation at the level of the bootcamp, followed by simple instructions for implementing three sample correction techniques from Zadrozny et al. (2003) and Chawla et al. (2002). The white paper contained links to further reading, in particular to Zadrozny (2004), Krautenbacher et al. (2017) and Cortes et al. (2008) in addition to the papers above.

Although we refer to this intervention as "technical education," it clearly falls short of a full degree-granting academic experience. Our goal with this treatment is to mimic "continuing education" materials that professional associations often develop in response to new developments in their field.

Randomization Procedure and Balance Tests Engineers were randomly assigned to one of four different conditions, using stratified random assignment. The stratification balanced ethnicity, major, school, degree, and gender of engineers across treatment groups.

Sub-Treatment: Incentives. As mentioned in Section 5, engineers were given benchmarks and a bonus slope. Harder thresholds (or a higher bonus slope) increases the benefit of seeking out and reducing more inaccuracy.

We randomized these accuracy thresholds and slopes to experimentally test the effects of different incentives. Each programmer's thresholds were 50/50 randomized to be higher or low.[17] In theory, those with a threshold that was more difficult to reach should have exerted extra effort to reach it, and thus get maximum credit for their task completion.

We also randomized each programmer's slope. This randomization affected how easy it was to obtain extra bonuses, once the more accurate threshold was passed. Subjects were randomly assigned to one of two slopes: one higher and one lower. The group with the easier-to-reach bonuses needed less improvement in their accuracy level to obtain the same amount of extra bonus.

The slope was randomized independently of the thresholds, and both incentive sub-treatments were drawn independently from other treatment conditions. Within each one of the four main treatment conditions, there were four sub-groups: some programmers had lower thresholds to reach, and an easier time obtaining bonuses; some had a lower threshold, and more difficult bonuses to obtain; some had a higher threshold and easy bonuses; some had a higher threshold and difficult bonuses.

Audit Manipulation: In the Fall semester instance of the bootcamp, we asked programmers to score an additional set of 4,500 observations, in addition to the earlier 20K. These were a subset of the original 20K, but contain manipulations. To compose these 4,500 we first drew a random sample of 1,500 from the 20K. We then flipped the genders in these observations, leaving everything else constant. Subjects were not told that the additional observations were artificial and were created this way. Subjects were then required to score both the original 20K (including the original 1,500) as well as the modified version.

This manipulation of inputs is a digital equivalent of resume-audit style research designs (), in which researchers send fictitious resumes to employers with randomized content. These designs hold other characteristics on the resume constant while randomizing the race or gender only (using racial- or gender-specific names). This exposes how much employer decisions were explicitly conditioned on gender or race.

Our digital "audit manipulations" similarly hold the other biographical characteristics constant in the PIAAC data, while manipulating only the gender variable. The comparison reveals how much predictions explicitly used the gender variable in algorithmic forecasts.

The 1,500 "gender-flipped" observations comprised one-third of the additional 4,500 predictions. We then took the original 1,500 and left gender (and all other variables) the same, but manipulated two variables that were highly correlated with gender. The variables we manipulated were a) the amount of time the subject spent looking after children, and b) the amount of time he/she performed unpaid household work. These were the two variables in the PIAAC data whose bivariate correlation with gender was highest. These 1,500 manipulated variables comprised the second third of the additional 4,500. Our goal with these manipulations was to examine the use of proxy

[17] For the two benchmarks, we no distinction between the two thresholds: each engineer either had both thresholds higher or both lower.

variables; i.e., to examine how much each programmer was indirectly utilizing gender through other data. This is a common concern in policymaking around algorithmic bias. For the final 1,500, we manipulated *both* the gender and the proxy variables.

7 Data and Specifications

To facilitate our analysis, we merge our data into a single dataset in which each programmer is represented as a row. Because each programmer makes 20K predictions, we summarize predictions by programmer level as described below. For all programmer-level outcome variables, we estimate the following equation report the coefficients in Section 8 (Results).

$$\begin{aligned} Outcome_i =& \beta[UnbiasedData_i + Reminder_i + TechAdvice_i \\ &+ IncentivesSlope_i + IncentivesBaseline_i \\ &+ FemaleProgrammer_i + IAT_i] + \epsilon_i \end{aligned} \tag{2}$$

Standard errors are clustered at the programmer level. Our analysis pools the bootcamp and contractor observations into a single dataset, although we are able to provide additional separate analyses by request. Before we examine the results, we briefly explain all terms in the equation above.

7.1 Outcome Variables

Our outcome variables come from two main sources: First, the 20K predictions submitted by each candidate. Second, from a confidential survey administered to participants about their approach to the programming problem.

Prediction Outcomes. We focus on two summary statistics of each engineer's predictions. The first is the mean absolute error (MAE) of the engineers' predictions. Lower MAEs represent more accuracy, although not necessarily less bias. For interpretability, we divided each programmer's MAE by the standard deviation of the entire PIAAC population's math ability. This allows us to express outcomes and effect sizes in units of standard deviations of math performance, rather than PIAAC's (0,5000) math literacy scale which may be unfamiliar to readers. We denote this outcome "$Abs(error)/\sigma$" in our statistical tables. For each engineer, we calculate overall MAE as well as MAE for men and women in the PIAAC predictions.

The second summary statistic is mean error (ME), calculated as simply *Predicted Math Ability* (estimated by the engineer's model *minus Actual Math Ability* (measured in the PIAAC data). This allows us to assess the average direction of the errors. When an engineer predicted a candidate was more skilled than he/she truly was, this number was positive. If an engineer predicted someone less skilled he/she was, this value is negative.

Like MAE, we also scaled this overestimation by the standard deviation of all math ability scores to aid interpretability ("On average, subjects overrated math ability for Group X by 0.3 standard deviations.") As we shall see, on average our engineers tended to overestimate math ability for everyone. As a result, we denote this as $Overestimate/\sigma$ in our regression tables.

Like our MAE estimates, we calculate overall ME for each as well as gender-specific MEs for men and women. This allows us to measure whether each engineer is as generous (or stingy) towards men and women, compared to their actual level of math skills. We summarize this notion of equality by subtracting the "Overestimation" of women from the for of men. In our statistical tables, this difference is labeled "Female - Male Overestimation." Negative values indicate bias favoring men. Positive values indicate bias favoring women.

For the MAE and ME summarized above, we truncated results at the 99th and 1st percentile of all predictions. This constrained the effects of a few very large outlier predictions. These mostly affected the precision of our estimates, and not the magnitude or directions. Because of how extreme these predictions are, we think that they would probably be discarded in practice. Without this truncation, all of our results are much less statistically significant.

Survey Outcomes. In order to complement our analysis of algorithmic output, we developed a survey, which engineers received after they submitted their work. All subjects were told that the survey was required, but the content of their responses would not affect their grade (for students) or payment or future engagements with us (for contractors).

All subjects were told up-front that there would be a survey. Both students and contract programmers typically document and explain their code upon submission; we explained that our survey replaced the need for paragraphs of explanation. However, they were not shown the questions survey until after submitting their code and predictions. This allowed us to ask questions about their approach to algorithmic bias ex-post without fear that the question would influence their submission.

We used the survey to collect information on how engineers built their algorithms (e.g., their functional form, how they selected their features, what libraries they used), as well as some behavioral dimension (e.g., what accuracy they expected on the test set, how many hours they worked on the task). Most survey questions were multiple-choice or structured (to facilitate our quantitative study).

A complete list of survey questions is available in our experimental materials appendix. In this manuscript, we focus on 12 survey responses in particular. We summarize these by theme below:[18]

Basic Technical Approach

1. *What programming language did you use?*
2. *What was the functional form of your machine learning algorithm? (ie, neural net, random forest, ridge regression, etc)?*
3. *How did you select which variables appear in your model*?

Questions about Awareness of Fairness Issues

[18] The descriptions are not ordered in the way they were below. Also, the summaries below capture the topics of the questions, for the wording of the question see the experimental materials appendix.

4. *Was your training data from a representative sample of the broader population? Or from a historical dataset of hires?* Note: We had an experimental condition that directly affects the correct answer to this question, so this is partly a check of the subject's awareness of the instructions.

5. *In programming your algorithm, did you think about helping any particular groups?*

Sample Selection Strategy

6. *Did you consider adopting sample selection correction?*

7. *Did you implement sample selection correction?*

8. *Did you read any Technical White Papers assigned in your instructions? Note: This may not have been necessary depending on your task.*

9. *Why didn't you implement sample selection correction? Check all that apply: "No Time," "Didn't think it was necessary," "We Weren't Taught How," etc.*

Willingness to Work on Problem

10. *Do you agree that performance in math jobs is objective and verifiable?*

11. For our bootcamp participants: *Are you comfortable if the instructors use the code you developed for the assignment for commercial purposes?*

12. *What company do you think provided the data?*

Answers to the questions above are mostly represented as binary variables in our analysis.

7.2 Predictor Variables

The above section outlined the "$Outcome_i$" variables in Equation 2. We now turn to the predictor or explanatory variables. These fall into two categories: The first set is our experimental interventions. Because these were randomized, we can interpret these results causally. The second explanatory variables are a set of programmer features that cannot be randomized but may help identify (non-causally) how programmers' outcomes varied.

Experimental Treatments. Our main treatments, described in Section 6, are simple to represent and interpret econometrically. Our data includes three binary variables, one for the "Unbiased Training Data" condition, one for our "Reminder" condition, and one for the "Technical Education" condition. The control condition is omitted so that effects are measured relative to the control. Importantly, the "Reminder" variable is 1 in the "Technical Advice" condition – this is because these instructions included both a link to a technical white paper *as well as* the reminder mentioned in Section 6. As a result, we should interpret the "Tech. Advice" coefficients as adding/subtracting to the Reminder condition.

Our incentive subtreatments were also randomly assigned. Each programmer was randomly assigned a "benchmark" and "slope," each in units of MSE, that dictate rewards. These were also standardized across programmers so that coefficients can be interpreted as "a one standard deviation increase in slopes/benchmark causes [...]."

Programmer Characteristics This research is motivated partly by questions about the machine learning workforce. As a result, our specifications will include characteristics of each programmer, which help assess the relationship between programmer characteristics and algorithmic performance.

Female: We used a names coding tool to help coding subjects as males or females. We then checked every subject's name and picture manually and assigned a 1 if the subject was male, and 0 if female.

The women in our sample differed from the men not only in their gender, but also along other covariates. Table 3 summarizes how other covariates change with the engineers' gender. Women engineers in our sample are less white, more Asian, and less experienced than their male counterparts. Among the bootcamp, their homework grades were higher.

We do *not* control for these other factors in our regressions, as this would change the interpretation of the results. If employers shifted hiring towards selecting more women, the shift would *not* produce candidates identical to men in all respects except their gender. Instead, it would produce candidates who were not only female, but also potentially different in other dimensions. Table 3 documents these differences in our sample.

Implicit Association Test: Finally, we explored psychological dimensions related to bias. Individuals have different levels of bias, and for the "biased programmers" hypothesis it is central that we understand whether programmers are biased against women, and by how much. To get at this dimension, we administered an Implicit Association Test ("IAT") from a standardized package (Carpenter, 2018). The IAT is a measure in social psychology aimed at detecting how strong are automatic associations (Greenwald, 1998): it measures people's associations, rather than their beliefs.

The test asks subjects to label pairs of associations quickly. To assess bias in these associations, the test measures how quickly subjects can label various pairs of ideas. In our implementation we asked subjects to label randomly generated pairs drawn from {Men,Women} and {Math,English}. As a group, our subjects were able to more quickly label {Men,Math} and {Women,English} than the other way around.

We collected this data in order to measure whether how much these reaction time measurements were correlated with the predictions from each subject's algorithm. We administered the Implicit Association Test at the end of the survey (the very last project in the task). By the time they took the test, subjects had been treated, had developed their algorithms, and answered all previous survey questions (among them, questions about their own assignment, and about biases) before taking the IAT.

In theory, the Implicit Association Test measures stable personal characteristics that are unaffected by the treatments or situations However, situation-specific changes have been shown in some cases to significantly affect IAT scores (Dasgupta and Greenwald, 2001). Therefore, we control for treatment assignment and report IAT scores as deviations from the mean IAT score within treatment.[19]

8 Results

Engineering Approach: Table 4 presents summary statistics about the engineering approaches. The overwhelming favorite programming language was Python, used by 93% of subjects and 97% of bootcamp students. A small majority (52%) of subjects built a random forest model, and about 22% built a neural network (30% of contractors). 32% used multiple methods in an ensemble method.[20]

Prediction Performance: Our engineer subjects predictions widely varied. Figure 3 shows a 2D kernel density plot of predicted and actual math performance. As the Figure shows, there is a slightly upward sloping relationship, but the predictions are extremely noisy. Approximately 55% of all predictions are half a standard deviation (or more) away from their true values. A linear regression predicting actual math skills from predicted math skills (on the test set) yields an R^2 of 0.17 and a slope of 0.61.[21]

Although the programmers may have been wrong about the *level* of math skills, they were more correct about the rank-ordering of PIAAC participants. A linear regression predicting percentile of actual math skills from percentile of predicted math skills (on the test set) yields an improved R^2 of 0.51 and a slope of 0.71. Figure 4 presents predictions in percentiles of the PIAAC sample. As it shows, the programmer predictions were similarly noisy. Approximately 55% of predictions were off by 15 percentiles or over. The average prediction was off by 23 percentiles – almost a quarter of the entire sample.

In Table 5, we present average performance statistics the overall sample of programmers. In the first row, we see that MAE for the average programmer was 0.77 of a standard deviation in math skills. On the second and third rows, we see that predictions of male ability featured less absolute error than for female math ability (0.75 vs 0.78). However, the rates similar in magnitude.

In the final three rows, we examine the average direction of the errors. On average, engineers' predictions overstated the math skills of the PIAAC population. However, the algorithms were particularly overgenerous towards women. Women's' math performance was on average overrated by 0.23 standard deviations (compared to PIAAC's measurement). For men, performance was overstated only by 0.11 standard deviations.

[19] Following standard procedures, we excluded subjects that had more than 10 percent of their IAT answers given in less than 300 milliseconds (Carpenter, 2018). This response time is judged by the literature as too low to be able to read the proposed associations effectively, thus we exclude subjects who were answering questions randomly or thoughtlessly.

[20] Perhaps unsurprisingly, these high-level engineering choices about algorithms and programming languages were broadly unaffected by our experimental treatments or the key demographic variables. We therefore present averages across all programmer types in Table 4.

[21] As discussed in Section 7, this analysis drops observations above the 99th percentile and below the 1st percentile. Without this adjustment, R^2 and the slope are both effectively zero.

On average, the algorithms produced by the engineer subjects displayed systematic overrating of a particular gender. The systematic overrating was towards women in the PIAAC sample, despite the fact that 75% of the engineers were given training data in which men were oversampled. In Figure 2, we show the distribution of all programmers tendency to overrate male vs female math ability. The figure shows that the systematic overrating of women was *not* driven by a single outlier programmer, but appears throughout the distribution. 93% of our programmers overrated women more than they overrated men.

In the sections below, we examine whether the extent of these outcomes is driven by our interventions or programmers' demographics.

8.1 Prediction Quality

Table 7 examines the effect or main explanatory variables on MAE errors. We find the strongest effect coming from giving engineers representative training data. On average, representative training data reduced MAE by 0.21 standard deviations. Unbiased training data appears to help MAE for female math ability better than male math ability by about 0.05 standard deviations.

The next most effective intervention was a simple reminder, described above. This was approximately 0.06σ less effective than representative training data and affected MAEs approximately equal between male and female math performance.

Our technical education intervention had a *negative effect* on prediction accuracy, raising MAEs by 0.17 standard deviations (0.15 for male and 0.19 for female). This magnitude essentially wiped out the benefit of the reminder, making the technical education statistically equivalent to the control group.

A natural question is whether these negative results depend on whether the subjects correctly implemented the technical guidance. When we control for implementation, we find a positive effect of implementing sample selection correction of varying degrees of statistical significance.

Our programmer demographic variables generally show no large effects. The coefficients on female programmers and the IAT score are not only small and statistically insignificant. We can also reject the hypothesis that they are equal to the size of the treatment effects. Table 7 also examines whether programmers' predictions are more accurate about people similar to themselves. Awareness of diversity issues in populations may focus the attention of minority engineers towards these subsets of the test data. Our results on this topic in Table 7 somewhat noisily estimated, but they suggest that (if anything) women engineers' predictions of female math skills are *worse* than their scores of men.

Table 8 examines the direction of these errors. As discussed above, our engineers' predictions were on average higher than ground truth. Representative training data reduces this overestimation by 0.17 standard deviations of skill, for both men and women. Because it reduces the overrating of women by slightly more, the unbiased data reduces the gender inequality in overrating.

The reminder produces opposite effects – it increases the overestimation of women relative to men. This may be a natural byproduct of programmers assuming that their biased training data would lower the algorithms' predictions of women, requiring a compensating factor to improve their assessments. In fact, it leads to the overrating of women compared to their true abilities and to that of men. Programmers compensating efforts to help women appear to have deepened the gender inequality favoring women.

The technical advice seems to move patterns of rating towards greater gender equality, but not by statistically significant amounts. Those who implemented the sample selection moved the overrating towards even greater favoring of women.

Our demographic variables once again show little effect, although we can reject the hypothesis that they are as influential as any of our treatment effects. As previously discussed, our incentives sub-treatments were generally underpowered. Our estimates in Tables 7 and 8 are relatively imprecise, and not distinguishable from zero (both individually and as part of a joint test).[22]

8.2 Mechanisms

Recognizing Bias. In Table 9, we present average statistics on subjects recognizing bias. The first column reports whether subjects thought the training data they received was unrepresentative of the full population. This is a way to check that our manipulations worked, and we see that our treatment did in fact affect subjects. Those in the unbiased data condition were less likely to report having biased data, although this effect is not particularly strong.

Those in the reminder condition were more likely to report they had biased data. Interestingly, women were also more likely to report biased data, and the effect of being a woman is extremely similar to receiving a reminder about the potential biases in the data. This aligns with the intuition women are at least more aware of biases than men. Finally, the technical education did not affect this answer.

The second column of the table reports whether subjects thought about helping specific groups. In this case, no coefficient is significant, with the exception of the technical guidance. Subjects in this condition were more likely to report thinking about helping specific groups.

Consideration of Sample Selection Strategies. Subjects could have taken care of the biases in the data by using sample selection correction. Table 10 and Table 11 report results related to this possibility. In particular, Table 10 shows only receiving technical advice had a consistent effect. Subjects who could access technical guidance were more likely to consider, as well as to implement, sample selection corrections. Consistent with the treatment, they were much more likely to have read the technical white paper, showing our manipulation for this group worked well.

In Table 11 we present results of a follow-up survey question to subjects who did not implement sample selection correction. We investigate the reasons why they did not implement it. No experimental condition was associated with a higher likelihood of answering that there was no time, or that sample selection was not necessary.

However, in line with our manipulation, subjects in the technical advice condition were less

[22] We can rule out the hypothesis that one standard deviation increase in incentives would improve MAE as much as the unbiased training data or reminder. However, our ability to extrapolate outside the amount of variation in our experiment is limited. The effects of significantly greater incentives may be non-linear.

likely to answer they were not taught how to deal with sample selection. Women were *less* likely to say they had no time to implement a correction, and more likely to say they were not taught how to do it.

Variable Selection Strategies. In the survey, we asked our subjects how they selected the features in their model. We wanted to understand whether different ways of variable selection were associated with some of the treatments. We report the results in Table 12, Panel A and B. Subjects in the technical advice group were the only ones that were significantly affected in their feature selection. In particular, they were less likely to use all variables, as well as to use all variables with either no pruning or automated pruning. All other groups were unaffected in their model's feature selection.

Attitudes about the task. Finally, Table 6 presents the engineers' attitudes towards the task. In particular, we wanted to check that subjects considered the task legitimate, and had no strong reservation in working on this. First, the task is based on the premise that there exists a "ground truth" performance for math skills, and our partner company has access to it. In Panel A, we can see the great majority of subjects accepted such a premise.

Panel B focuses on bootcamp participants, as contracted engineers had already agreed to develop their algorithms for commercial use. We wanted to measure whether, among students at the bootcamp, there were potential ethical or legal concerns on using the algorithms developed for commercial purposes. Perhaps our subjects accepted performance in math jobs is objective but, because of personal reservations, did not want to predict it. Instead, we can see that only 7.2 percent of students were not comfortable with commercial use. 92.8 percent would have accepted the company using their algorithms for commercial use, and a similar percentage (91.9) was even interested in additional paid consulting. Finally, we wanted to check that our obfuscation worked, and programmers did not understand where the data came from. More than 83 percent of subjects had no idea where the data came from. Additionally, none of the remaining subjects guessed the OECD or PIAAC as the data source.

9 Conclusions

As algorithms spread in influence, concern has grown about algorithmic bias. However, the root causes of algorithmic bias are often unclear. In public discourse and academic literature, two theories of algorithmic have gained prominence.

The first theory emphasizes "Biased Training Data." Because machine learning applications are developed using historical data about outcomes, data coming from it would reflect and perpetuate any bias in the real world. A second theory emphasizes another factor "Biased Programmers." The Programmers developing algorithms are highly non-representative and may exhibit biases that are passed onto the algorithms they write.

Both of these theories are likely contributors to algorithmic bias. However, the two theories require different policy solutions. In this paper, we seek to measure the relative contributions of biased data and biased programmers. We then test several policies and educational interventions for reducing algorithmic bias. We compared them against the standard in most settings: An unrepresentative dataset of convenience based on history.

We find that representative training data is the most effective intervention, in line with the "Biased Training Data" hypothesis. Many researchers have discussed the limitations of using only datasets of convenience based on the available data. However, the very fact that these were the datasets commonly used, made it very hard to quantify the extent of this problem. We find little effect of altering programmer demographics or from programmers who score worse on psychology measures of implicit bias. This strongly suggests organizations should strive to collect better data and exert effort to increase their data reliability and inclusivity.

To obtain this result, we exploited a very high-quality training dataset, developed by the OECD. We believe this to be an upper-bar in terms of data quality, at a level certainly beyond what companies currently do.

We also find that a non-technical reminder is also effective at affecting altering outcomes. Our reminder intervention reduced error, however, it increased gender imbalances favoring women. Given the inherent difficulties in gathering unbiased data, these reminders may be an effective alternative. All engineers with biased data were aware their data was biased. However, a few paragraphs that focused their attention on this issue were enough to improve predictions.

Our technical education intervention turned out to be detrimental to accuracy and had little effect on gender parity. Our engineers appeared to be affected by a simple reminder, but they could not implement a few pages of technical guidance. This is particularly relevant to practice and education. Complex solutions may not be effective if they are misunderstood by practitioners.

Many of our results may be specific to our particular setting and interventions. Alternative technical guidance may have produced different outcomes. Similarly, engineers from other backgrounds may react differently to our treatments. Our paper should not be the final word on any of these topics, and there might be other sources of algorithmic bias besides prediction bias which we do not discuss in this paper. We hope that future algorithmic fairness researchers will embrace the challenge of empirically measuring the effects of educational and policy interventions. This new empirical science should ideally examine a wide variety of effects, including interventions and settings well beyond those of this paper.

References

Agan, Amanda and Sonja Starr, "Ban the box, criminal records, and racial discrimination: A field experiment," *The Quarterly Journal of Economics*, 2017, *133* (1), 191–235.

__ **, Bo Cowgill, and Laura Gee**, "The Effects of Salary History Bans: Evidence from a Field Experiment," *Working paper*, 2019.

Agrawal, Ajay, Joshua S Gans, and Avi Goldfarb, "Exploring the impact of artificial intelligence: Prediction versus judgment," *Information Economics and Policy*, 2019, *47*, 1–6.

Arrow, Kenneth, "The theory of discrimination," *Discrimination in labor markets*, 1973, *3* (10), 3–33.

Ayres, Ian, "Outcome tests of racial disparities in police practices," *Justice research and Policy*, 2002, *4* (1-2), 131–142.

Bertrand, Marianne and Esther Duflo, "Field experiments on discrimination," in "Handbook of economic field experiments," Vol. 1, Elsevier, 2017, pp. 309–393.

__ **and Sendhil Mullainathan**, "Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination," *American economic review*, 2004, *94* (4), 991–1013.

Bogen, Miranda and Aaron Rieke, "Help wanted: an examination of hiring algorithms, equity," Technical Report, and bias. Technical report, Upturn 2018.

Bohren, J Aislinn, Kareem Haggag, Alex Imas, and Devin G Pope, "Inaccurate statistical discrimination," Technical Report, National Bureau of Economic Research 2019.

Bordalo, Pedro, Katherine Coffman, Nicola Gennaioli, and Andrei Shleifer, "Stereotypes," *The Quarterly Journal of Economics*, 2016, *131* (4), 1753–1794.

Chawla, Nitesh V, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, 2002, *16*, 321–357.

Cortes, Corinna, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh, "Sample selection bias correction theory," in "International conference on algorithmic learning theory" Springer 2008, pp. 38–53.

Cowgill, Bo, "How Algorithms Impact Judicial Decisions," *Working paper*, 2017.

__ , "Automating Judgement and Decisionmaking: Theory and Evidence from Résumé Screening," *Working paper*, 2019.

__ **and Catherine E Tucker**, "Economics, fairness and algorithmic bias," *preparation for: Journal of Economic Perspectives*, 2019.

__ **and Patryk Perkowski**, "Agency and Homophily: Evidence from a Two-Sided Audit," *Working Paper*, 2019.

Daniel, William Wentworth, *Racial discrimination in England: based on the PEP report*, Vol. 257, Penguin, 1968.

Dasgupta, Nilanjana and Anthony G Greenwald, "On the malleability of automatic attitudes: combating automatic prejudice with images of admired and disliked individuals.," *Journal of personality and social psychology*, 2001, *81* (5), 800.

DellaVigna, Stefano, "Psychology and economics: Evidence from the field," *Journal of Economic literature*, 2009, *47* (2), 315–72.

Gaddis, S Michael, *Audit studies: Behind the scenes with theory, method, and nuance*, Vol. 14, Springer, 2018.

Hoffman, Mitchell, Lisa B Kahn, and Danielle Li, "Discretion in hiring," Technical Report, National Bureau of Economic Research 2015.

_ , _ , and _ , "Discretion in hiring," *The Quarterly Journal of Economics*, 2017, *133* (2), 765–800.

Kessler, Judd B, Corinne Low, and Colin D Sullivan, "Incentivized Resume Rating: Eliciting Employer Preferences without Deception," *American Economic Review*, 2019.

Kleinberg, Jon, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan, "Human decisions and machine predictions," *The quarterly journal of economics*, 2018, *133* (1), 237–293.

Krautenbacher, Norbert, Fabian J Theis, and Christiane Fuchs, "Correcting classifiers for sample selection bias in two-phase case-control studies," *Computational and mathematical methods in medicine*, 2017, *2017.*

Larson, Jeff, Surya Mattu, Lauren Kirchner, and Julia Angwin, "How we analyzed the COMPAS recidivism algorithm," *ProPublica (5 2016)*, 2016.

McGhee, Debbie E.; Schwartz Jordan L.K. Greenwald Anthony G.;, "Measuring Individual Differences in Implicit Cognition: The Implicit Association Test," *ournal of Personality and Social Psychology*, 1998, *74* (6), 1464–1480.

Neumark, David, "Experimental research on labor market discrimination," *Journal of Economic Literature*, 2018, *56* (3), 799–866.

O'Neil, Cathy, *Weapons of math destruction: How big data increases inequality and threatens democracy*, Broadway Books, 2017.

Pager, Devah, "The use of field experiments for studies of employment discrimination: Contributions, critiques, and directions for the future," *The Annals of the American Academy of Political and Social Science*, 2007, *609* (1), 104–133.

Parkes, David C, Rakesh V Vohra et al., "Algorithmic and Economic Perspectives on Fairness," *arXiv preprint arXiv:1909.05282*, 2019.

Phelps, Edmund S, "The statistical theory of racism and sexism," *The american economic review*, 1972, *62* (4), 659–661.

Schleicher, Andreas, "PIAAC: A new strategy for assessing adult competencies," *International Review of Education*, 2008, *54* (5-6), 627–650.

T., Pogacar R. Pullig C. Kouril M. Aguilar S. LaBouff J. P. Chakroff A. Carpenter, "Conducting IAT Research within Online Surveys: A Procedure, Validation, and Open Source Tool," *PsyArXiv*, 2018.

Tetlock, Philip E, *Expert political judgment: How good is it? How can we know?-New edition*, Princeton University Press, 2017.

Thompson, Nicholas, "Playing With Numbers.," *Washington Monthly*, 2000, *32* (9), 16–23.

Wolfers, Justin and Eric Zitzewitz, "Prediction markets," *Journal of economic perspectives*, 2004, *18* (2), 107–126.

Zadrozny, Bianca, "Learning and evaluating classifiers under sample selection bias," in "Proceedings of the twenty-first international conference on Machine learning" ACM 2004, p. 114.

_, John Langford, and Naoki Abe, "Cost-Sensitive Learning by Cost-Proportionate Example Weighting.," in "ICDM," Vol. 3 2003, p. 435.

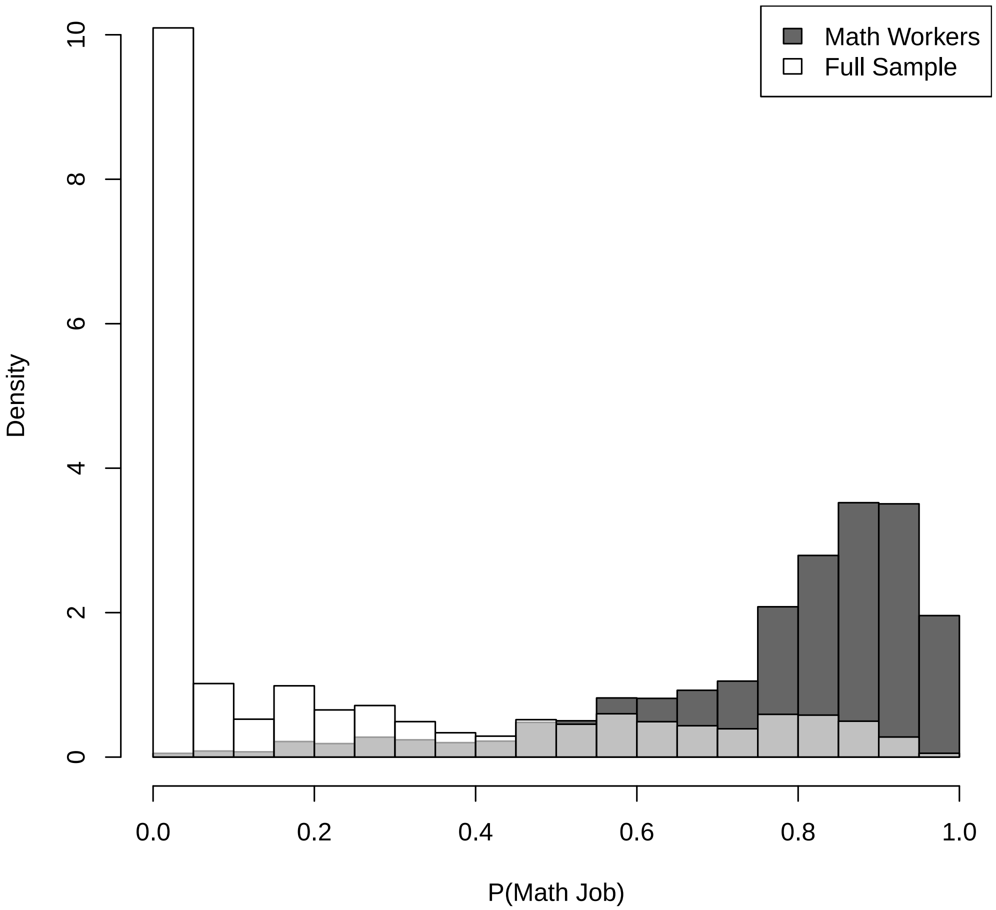

Figure 1: Probability of Becoming a Math Worker: Math Workers vs the General Public

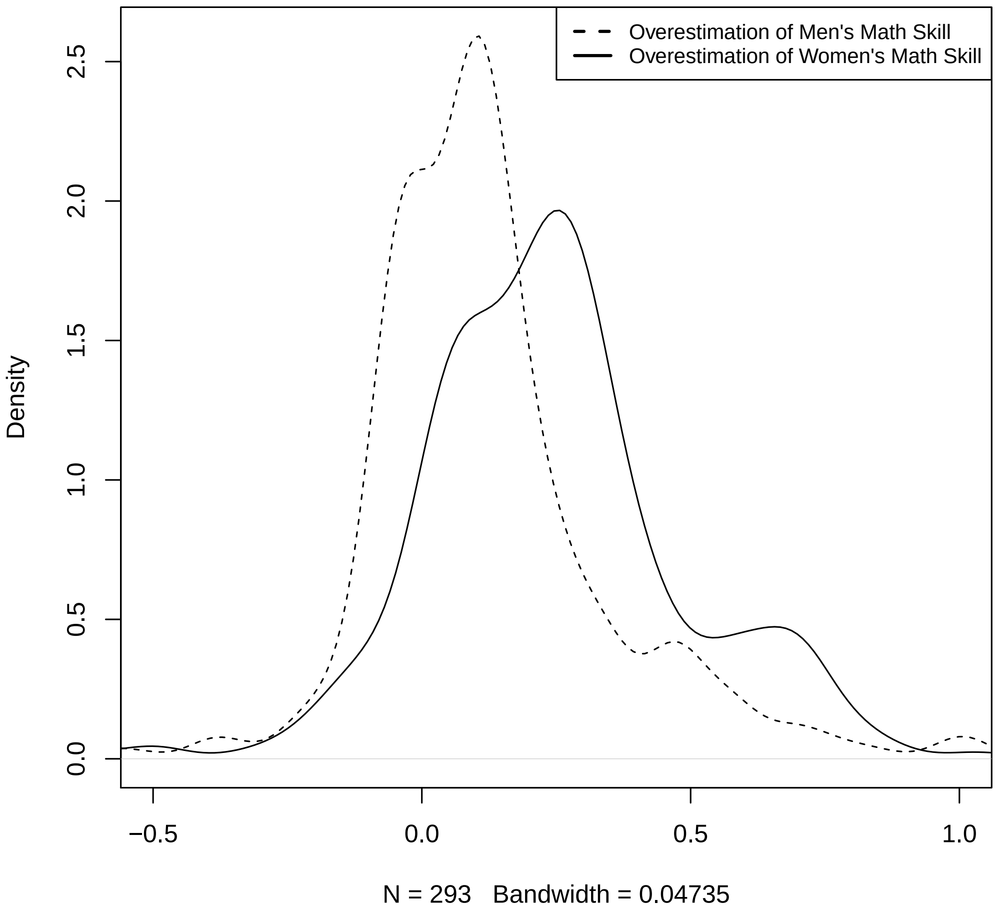

Figure 2: Algorithmic Overestimation of Math Skill by Gender

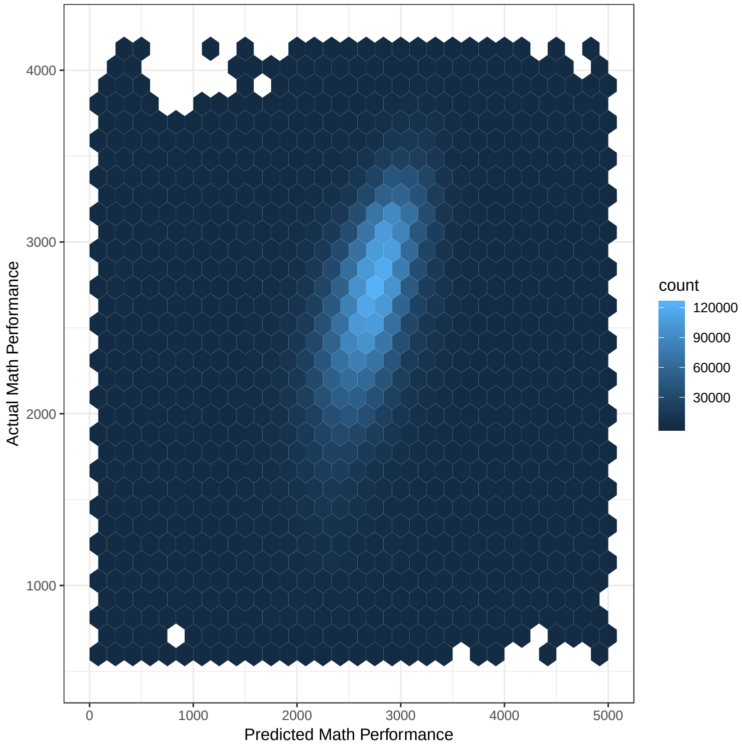

Figure 3: Predicted Math Performance vs Actual Math Performance: Raw Score

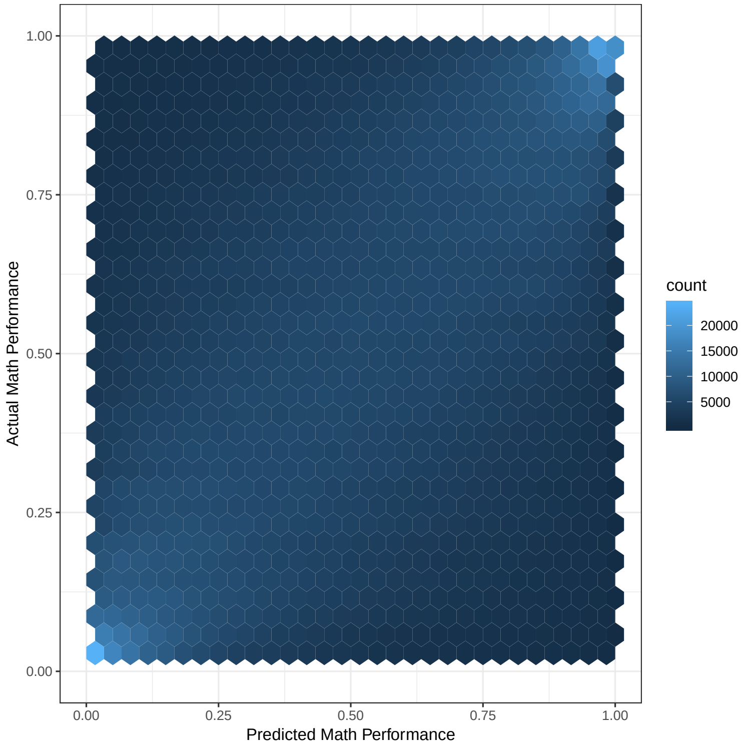

Figure 4: Predicted Math Performance vs Actual Math Performance: Percentile

Table 1: **Descriptive Statistics: Subjects**

All Subjects

	Bootcamp	Contracted	Total
White	0.21	0.58	0.28
East Asian	0.61	0.15	0.52
Other Asian	0.14	0.17	0.15
Black or Latin	0.04	0.10	0.05
Male	0.69	0.80	0.71
Years of Work (All)	1.25	3.86	1.76
Years of Work (Programming)	0.73	3.86	1.35
Completed Survey	0.98	0.95	0.98
Submitted Assignment	0.98	0.98	0.98
Observations	339	60	399

Table 2: **Math Workers vs. Full Population**

	Math Workers	Full Population	Difference
Male	0.625	0.494	0.130***
Birth Region: N. America and W. Europe	0.643	0.572	0.070***
Birth Region: Central and Eastern Europe	0.169	0.237	-0.068***
Birth Region: East Asia and the Pacific (poorer countries)	0.004	0.003	0.001
Age: 24 or less	0.088	0.175	-0.087***
Age: 25-34	0.271	0.205	0.066***
Age: 35-44	0.278	0.213	0.064***
Age: 45-54	0.235	0.214	0.021***
Age: 55 plus	0.127	0.192	-0.065***
Education: Above high school	0.647	0.387	0.260***
Education: High school	0.291	0.378	-0.087***
Education: Less than high school	0.037	0.218	-0.181***
Math Skill (Norm.)	0.607	0.000	0.607***
Major: Agriculture and veterinary	0.025	0.029	-0.004**
Major: Engineering, manufacturing and construction	0.276	0.187	0.088***
Major: General programmes	0.101	0.189	-0.088***
Major: Health and welfare	0.052	0.096	-0.044***
Major: Humanities, languages and arts	0.054	0.084	-0.030***
Major: Science, mathematics and computing	0.152	0.097	0.055***
Major: Services	0.039	0.073	-0.034***
Major: Social sciences, business and law	0.253	0.174	0.079***
Major: Teacher training and education science	0.048	0.070	-0.023***

Table 3: **Engineer Covariates By Gender**

Panel A: All Subjects

	Women	Men	Difference
White	0.205	0.298	-0.093**
East Asian	0.616	0.509	0.107*
Other Asian	0.143	0.144	-0.001
Black or Latin	0.036	0.049	-0.013
Years of Work (All)	1.275	1.966	-0.691***
Years of Work (Programming)	0.970	1.504	-0.534**
Completed Survey	0.955	0.979	-0.024
Submitted Assignment	0.982	0.982	-0.000
Observations	397		

Panel B: Bootcamp Participants Only

	Women	Men	Difference
Undergrad.	0.495	0.403	0.092
CS Major	0.576	0.584	-0.008
Homework Avg.	0.885	0.887	-0.002
Employer is a Household Name	0.347	0.305	0.042
Observations	337		

Table 4: **Engineering Approach**

	Bootcamp	Contracted	Total
Used Python	0.97	0.79	0.94
Random Forests	0.50	0.63	0.52
Linear Model (Lasso, Ridge, OLS, etc)	0.36	0.28	0.35
Neural Nets	0.20	0.30	0.22
Non-Linear Regression	0.05	0.02	0.04
Ensemble Approach	0.33	0.39	0.34
Observations	339	60	399

Table 5: **Summary Performance Statistics (Overall)**

	Mean	Median	SD
Abs(error)/σ	0.76	0.65	0.36
Abs(error)/σ, Male Math Ability	0.74	0.64	0.34
Abs(error)/σ, Female Math Ability	0.77	0.66	0.37
Overestimate/σ of Male Math Ability	0.10	0.10	0.35
Overestimate/σ of Female Math Ability	0.22	0.23	0.37
Female Overestimation - Male Overestimation	0.12	0.13	0.09
Female Overestimation / Male Overestimation	1.13	1.54	6.49
Observations	389		

Table 6: **Willingness to Work on Problem**

Panel A: All Subjects

	Mean
Agrees: Performance is Measurable in Math Jobs	0.71
Observations	388

Panel B: Bootcamp Participants Only

	Mean
Comfortable w/ Commercial Use of Code, No Strings Attached	0.50
Comfortable w/ Commercial Use of Code, Only You Pay Me	0.41
Never Comfortable w/ Commercial Use of Code	0.08
Interested in Paid Consulting	0.92
No Idea who Partner Company Is	0.85
Observations	331

Table 7: **Prediction Performance: Absolute Error**

	Abs(error)/σ	Abs(error)/σ	Abs(error)/σ
	All	Male	Female
Representative Training Data	-.18***	-.16***	-.2***
	(.048)	(.048)	(.049)
Reminder	-.1**	-.1**	-.1**
	(.05)	(.05)	(.05)
Tech. White Paper	.13**	.12**	.14**
	(.054)	(.051)	(.058)
newslopetoshow	-.000013	-.000014	-.000012
	(.000031)	(.00003)	(.000032)
newupdatedbaseline	4.3e-07	5.4e-07	2.8e-07
	(5.2e-06)	(5.1e-06)	(5.3e-06)
fall_slope	.000082	.000076	.000088
	(.0001)	(.0001)	(.00011)
fall_thresh60	3.1e-06	2.2e-06	4.1e-06
	(5.1e-06)	(5.3e-06)	(5.0e-06)
Female Programmer	.063	.047	.08*
	(.043)	(.04)	(.045)
IAT Score (σ)	.00093	-.00032	.0022
	(.02)	(.02)	(.021)
iat_ms	.023	.025	.021
	(.046)	(.044)	(.047)
East Asian	-.018	-.026	-.011
	(.069)	(.07)	(.07)
Other Asian	-.043	-.049	-.037
	(.1)	(.1)	(.1)
Black or Latin	.063	.046	.081
	(.067)	(.067)	(.069)
Mean of Control Group	.82	.8	.83
P-value of Incentives Joint Test	.81	.86	.75
Observations	388	388	388
R^2	.19	.18	.21

Table 8: **Prediction Performance: Overestimation by Gender**

	Overestimation Male	Overestimation Female	Female-Male Overestimation	Female Overest./ Male Overest.
Representative Training Data	-.16***	-.19***	-.03**	-2.6***
	(.053)	(.053)	(.015)	(.79)
Reminder	.051	.072	.021	.6
	(.053)	(.054)	(.015)	(.97)
Tech. White Paper	.043	.038	-.0053	-.94
	(.049)	(.053)	(.014)	(1.1)
newslopetoshow	-.000024	-.000018	6.6e-06	.00022
	(.000029)	(.00003)	(8.0e-06)	(.00063)
newupdatedbaseline	-3.8e-06	-3.6e-06	2.3e-07	.00018*
	(4.4e-06)	(4.5e-06)	(1.1e-06)	(.0001)
fall_slope	-.00011	-.00012	-6.7e-06	-.00065
	(.00015)	(.00016)	(.000028)	(.0021)
fall_thresh60	2.3e-06	3.8e-06	1.6e-06	.000033
	(7.4e-06)	(7.8e-06)	(1.6e-06)	(.0001)
Female Programmer	.039	.054	.015	.38
	(.047)	(.05)	(.012)	(.71)
IAT Score (σ)	-.013	-.013	-.00016	-.62*
	(.023)	(.023)	(.0053)	(.37)
iat_ms	-.021	-.03	-.0094	-1.6*
	(.053)	(.058)	(.02)	(.94)
East Asian	.1	.11	.0045	2.3**
	(.065)	(.067)	(.013)	(1.1)
Other Asian	.16*	.19**	.028	1.3
	(.092)	(.095)	(.018)	(.81)
Black or Latin	.17	.15	-.018	3.5**
	(.11)	(.12)	(.029)	(1.4)
Mean of Control Group	.1	.22	.12	1.5
P-value of Incentives Joint Test	.81	.85	.79	.39
Observations	388	388	388	388
R^2	.22	.23	.18	.21

Table 9: **Recognizing Bias**

	Training Data is Not Representative	Thought about Helping Specific Groups
Representative Training Data	-.21***	.019
	(.07)	(.043)
Reminder	.13**	.068
	(.062)	(.046)
Tech. White Paper	-.0042	.11*
	(.064)	(.058)
newslopetoshow	-.000039	.000032
	(.000035)	(.00003)
newupdatedbaseline	1.4e-06	2.9e-06
	(5.2e-06)	(4.4e-06)
fall_slope	-.00018	-.00012
	(.00018)	(.000097)
fall_thresh60	-7.1e-06	2.5e-06
	(.000012)	(5.0e-06)
Female Programmer	.17***	-.024
	(.053)	(.045)
IAT Score (σ)	.0033	-.028
	(.028)	(.019)
iat_ms	.097	.045
	(.11)	(.092)
East Asian	-.054	-.065
	(.066)	(.053)
Other Asian	-.029	-.085
	(.087)	(.069)
Black or Latin	-.11	-.12
	(.14)	(.092)
Mean of Control Group	.62	.079
P-value of Incentives Joint Test	.6	.58
Observations	388	388
R^2	.3	.19

Table 10: **Sample Selection Strategy**

	Considered Sample	Implemented Sample	Read Tech. Advice
	Selection Correction	Selection Correction	White Paper
Representative Training Data	-.17**	.034	.04
	(.07)	(.035)	(.059)
Reminder	.0034	.013	.013
	(.068)	(.034)	(.054)
Tech. White Paper	.2***	.12**	.53***
	(.073)	(.046)	(.062)
newslopetoshow	.000024	.000036	.000033
	(.000038)	(.000025)	(.00003)
newupdatedbaseline	-9.3e-06*	5.6e-07	-4.1e-06
	(5.2e-06)	(3.6e-06)	(4.2e-06)
fall_slope	.000053	.000019	-.000085
	(.00021)	(.000028)	(.00017)
fall_thresh60	4.5e-07	-1.0e-06	5.3e-06
	(.000013)	(1.5e-06)	(.000012)
Female Programmer	-.097	.045	.053
	(.06)	(.035)	(.05)
IAT Score (σ)	.0079	.008	.01
	(.027)	(.02)	(.021)
iat_ms	-.11	-.061	-.082
	(.13)	(.04)	(.12)
East Asian	.041	-.0068	.14**
	(.073)	(.044)	(.057)
Other Asian	.1	-.0035	.13*
	(.097)	(.056)	(.071)
Black or Latin	-.039	.056	.07
	(.14)	(.093)	(.13)
Mean of Control Group	.43	.04	.17
P-value of Incentives Joint Test	.45	.56	.65
Observations	388	388	388
R^2	.25	.23	.38

Table 11: **Why No Sample Selection?**

	No Time	Not Necessary	Not Taught
Representative Training Data	-.059	-.003	-.02
	(.067)	(.068)	(.071)
Reminder	-.058	-.011	.096
	(.065)	(.067)	(.069)
Tech. White Paper	.0092	-.028	-.19***
	(.071)	(.071)	(.067)
newslopetoshow	-.000027	-.000027	.000062*
	(.00004)	(.000038)	(.000038)
newupdatedbaseline	-5.2e-06	3.1e-06	-5.9e-06
	(5.4e-06)	(5.4e-06)	(5.2e-06)
fall_slope	.000047	.000088	.00017
	(.00018)	(.0002)	(.0002)
fall_thresh60	6.4e-06	6.2e-06	5.1e-06
	(.000011)	(.000012)	(.000013)
Female Programmer	-.053	.041	.19***
	(.059)	(.06)	(.063)
IAT Score (σ)	.00051	-.0078	.045*
	(.026)	(.026)	(.025)
iat_ms	.17	-.12	.13
	(.12)	(.1)	(.12)
East Asian	-.02	-.027	.14*
	(.066)	(.075)	(.072)
Other Asian	.14	-.11	.092
	(.09)	(.096)	(.098)
Black or Latin	-.15	-.17	-.054
	(.15)	(.13)	(.13)
Mean of Control Group	.36	.31	.36
P-value of Incentives Joint Test	.74	.85	.35
Observations	388	388	388
R^2	.2	.18	.26

Table 12: **Variable Selection Strategies**

Panel A:

	Read codebook, used what intuitively made sense.	Used All Features	Used All Features No Pruning
Representative Training Data	.15**	.13*	.015
	(.072)	(.067)	(.046)
Reminder	.17**	.095	.02
	(.068)	(.063)	(.045)
Tech. White Paper	-.021	-.19***	-.043
	(.073)	(.066)	(.051)
newslopetoshow	-6.9e-06	.000042	-.000012
	(.00004)	(.000036)	(.000029)
newupdatedbaseline	3.5e-06	6.5e-06	3.3e-07
	(5.6e-06)	(5.1e-06)	(3.8e-06)
fall_slope	.00011	.00002	-.00011
	(.00022)	(.00021)	(.00012)
fall_thresh60	-3.2e-06	-9.1e-06	1.6e-06
	(.000014)	(.000012)	(5.9e-06)
Female Programmer	.01	-.042	-.028
	(.063)	(.057)	(.037)
IAT Score (σ)	-.034	-.018	.0075
	(.028)	(.027)	(.019)
iat_ms	-.15	-.1	-.064
	(.1)	(.12)	(.075)
East Asian	.062	.1	-.053
	(.071)	(.075)	(.056)
Other Asian	.14	-.071	-.066
	(.096)	(.093)	(.067)
Black or Latin	.11	-.18	-.14**
	(.13)	(.15)	(.059)
Mean of Control Group	.28	.68	.099
P-value of Incentives Joint Test	.96	.44	.9
Observations	388	388	388
R^2	.2	.2	.16

Panel B:

	Used All Features Automated Pruning	Used All Features Manual Pruning	Unsupervised Feature Selection
Representative Training Data	.17**	.027	.098
	(.072)	(.071)	(.064)
Reminder	.19***	.014	.0044
	(.07)	(.063)	(.058)
Tech. White Paper	-.19**	.027	.061
	(.076)	(.066)	(.061)
newslopetoshow	.00007*	.00002	-.000048
	(.000042)	(.000035)	(.000035)
newupdatedbaseline	5.8e-06	2.6e-06	2.7e-07
	(5.9e-06)	(5.0e-06)	(4.8e-06)
fall_slope	-.00013	5.5e-06	-.0003*
	(.0002)	(.00021)	(.00016)
fall_thresh60	-.000015	-1.2e-07	.00002*
	(.000014)	(.000012)	(.000011)
Female Programmer	.02	.015	.059
	(.062)	(.057)	(.053)
IAT Score (σ)	-.036	-.016	.01
	(.029)	(.027)	(.025)
iat_ms	-.074	-.0042	-.016
	(.11)	(.11)	(.11)
East Asian	.11	.013	-.026
	(.078)	(.071)	(.065)
Other Asian	-.08	-.02	.13
	(.1)	(.092)	(.09)
Black or Latin	-.14	.03	-.0083
	(.13)	(.15)	(.14)
Mean of Control Group	.4	.28	.21

Table 13: **Results of Audit-Manipulations of Inputs**

	(max) a1_same	(max) a2_same	(max) a3_same
Representative Training Data	.12	.1	.17
	(.15)	(.15)	(.14)
Reminder	-.028	-.008	-.042
	(.15)	(.15)	(.15)
Tech. White Paper	.027	-.21	.029
	(.17)	(.17)	(.17)
newslopetoshow	0	0	0
	(.)	(.)	(.)
newupdatedbaseline	0	0	0
	(.)	(.)	(.)
fall_slope	-.000044	.000059	-.000019
	(.00021)	(.00022)	(.00019)
fall_thresh60	7.9e-06	.000012	1.0e-07
	(.000014)	(.000013)	(.000013)
Female Programmer	-.052	-.18	-.084
	(.13)	(.13)	(.12)
IAT Score (σ)	.058	.0078	.027
	(.079)	(.078)	(.079)
iat_ms	.13	-.023	.16
	(.2)	(.21)	(.21)
East Asian	.013	.18	-.067
	(.16)	(.14)	(.15)
Other Asian	-.028	.021	-.018
	(.21)	(.2)	(.21)
Black or Latin	-.48	-.25	-.58
	(.44)	(.24)	(.48)
Mean of Control Group	.59	.41	.63
P-value of Incentives Joint Test	.84	.6	.99
Observations	96	96	96
R^2	.18	.23	.2

Table 14: **Pairwise Correlations Between Subjects**

	Correlation	Correlation	Correlation
Same Gender	.072**		
	(.033)		
Same Race	.0087		
	(.023)		
Both Men		.17**	.16**
		(.076)	(.075)
Both Women		-.14**	-.13*
		(.069)	(.068)
Both White		.021	.024
		(.072)	(.071)
Both Asian		.016	.014
		(.063)	(.063)
Both Unbiased Data			.31***
			(.093)
Both Reminder			-.045
			(.06)
Both Tech Advice			-.076
			(.065)
Observations	75855	75855	75855
R^2	0.483	0.486	0.490

Notes: The outcome variable has been standardized, so that effects are reported in standard deviations.

Appendix: For Online Publication Only

A Survey Questions

1. *Are you comfortable with your code and predictions being used for commercial purposes?*
2. *If our research partner wants to pay you for additional consulting about this project, should we introduce you?*
3. *What was your "easily achievable baseline" MSE?*
4. *What was your "target" MSE?*
5. *How much MSE improvement above the benchmark was necessary for 1% in extra credit?*
6. *What MSE do you expect to achieve on the test set? If you're not sure, enter the MSE of the model you turned in when applied to your problem*
7. *What programming language did you use?*
8. *How many hours did you spend on the assignment?*
9. *How did you select the features in the model? (e.g., I read the codebook and utilized features that intuitively made sense; I applied unsupervised ML approaches (PCA or something similar) to reduce the feature space ahead of time; etc.)*
10. *How did you test your approaches?*
11. *What was the functional form of your machine learning algorithm? (ie, neural net, random forest, ridge regression, etc)?*
12. *Did your final piece of code use many interactions/combinations of multiple variables?*
13. *How did you select which interactions to include?*
14. *Did you attempt to regularize variables?*
15. *My best guess of who the research partner company is: (open answer)*
16. *Describe your TRAINING data*
17. *Describe your TEST data*
18. *Do you agree that performance in math-related jobs can be measured objectively – for example, by verifying answers against provably correct answers?*
19. *Did you read one of the white papers accompanying the instructions? Note: It may not have been necessary depending on your assignment*
20. *In completing your assignment, did you attempt any form of sample selection correction? Note: It may not have been necessary depending on your assignment*

21. *What sample selection correction strategy did you implement? (e.g., Sampling with replacement, Costing, etc.)*

22. *Why didn't you implement a sample selection correction strategy?*

23. *I suspect that the average student spoke with (number) other students to strategize about approaches for this task*

24. *Did you think about improving outcomes for particular types of people in the data?*

25. *Why were you thinking about improving outcome for particular types of people in the data? Rank the following reasons: (e.g., "I thought it would improve my MSE"; "It may harm MSE, but I want to help certain group overcome historical adversity"*

26. *Did you do change your algorithm to benefit particular types of people in the data?*

27. *How did you change your approach to your algorithm (briefly describe in English)?*

28. *Why didn't you change your algorithm?*

B Reminder

We report below the text of the reminder that subjects in the third condition received,

> *Note: As a reminder, our recruiters' expectations about who to hire may have been systematically wrong. Performance data is available only for workers like the ones we've hired in the past. This may reflect our recruiters' stereotypes or biases about who is good at math.*
>
> *As you write your algorithm, please be mindful that your training dataset may originate in a* biased social system. *Adjusting your algorithm to account for discrimination in hiring, self-sorting of applicants, or other sources of such bias could improve your accuracy on the test set. You will be evaluated only on the accuracy of your predictions on the test set.*
>
> *Please note: For the workers we've hired in the past, we are confident their performance outcomes in your training dataset have been labeled accurately. Performance is based on math and there are objective measures of quality. However, you only have this data about the candidates we previously hired.*