NBER WORKING PAPER SERIES

HUMAN DECISIONS AND MACHINE PREDICTIONS

Jon Kleinberg
Himabindu Lakkaraju
Jure Leskovec
Jens Ludwig
Sendhil Mullainathan

Working Paper 23180
http://www.nber.org/papers/w23180

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
February 2017

We are immensely grateful to Mike Riley for meticulously and tirelessly spearheading the data analytics, with effort well above and beyond the call of duty. Thanks to David Abrams, Matt Alsdorf, Molly Cohen, Alexander Crohn, Gretchen Ruth Cusick, Tim Dierks, John Donohue, Mark DuPont, Meg Egan, Elizabeth Glazer, Judge Joan Gottschall, Nathan Hess, Karen Kane, Leslie Kellam, Angela LaScala-Gruenewald, Charles Loeffler, Anne Milgram, Lauren Raphael, Chris Rohlfs, Dan Rosenbaum, Terry Salo, Andrei Shleifer, Aaron Sojourner, James Sowerby, Cass Sunstein, Michele Sviridoff, Emily Turner, and Judge John Wasilewski for valuable assistance and comments, to Binta Diop, Nathan Hess, and Robert Webberfor help with the data, to David Welgus and Rebecca Wei for outstanding work on the data analysis, to seminar participants at Berkeley, Carnegie Mellon, Harvard, Michigan, the National Bureau of Economic Research, New York University, Northwestern, Stanford and the University of Chicago for helpful comments, to the Simons Foundation for its support of Jon Kleinberg's research, to the Stanford Data Science Initiative for its support of Jure Leskovec's research, to the Robert Bosch Stanford Graduate Fellowship for its support of Himabindu Lakkaraju and to Susan and Tom Dunn, Ira Handler, and the MacArthur, McCormick and Pritzker foundations for their support of the University of Chicago Crime Lab and Urban Labs. The main data we analyze are provided by the New York State Division of Criminal Justice Services (DCJS), and the Office of Court Administration. The opinions, findings, and conclusions expressed in this publication are those of the authors and not those of DCJS. Neither New York State nor DCJS assumes liability for its contents or use thereof. The paper also includes analysis of data obtained from the Inter-University Consortium for Political and Social Research at the University of Michigan. Any errors and all opinions are our own. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

Human Decisions and Machine Predictions
Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan
NBER Working Paper No. 23180
February 2017
JEL No. C01,C54,C55,D8,H0,K0

ABSTRACT

We examine how machine learning can be used to improve and understand human decision-making. In particular, we focus on a decision that has important policy consequences. Millions of times each year, judges must decide where defendants will await trial—at home or in jail. By law, this decision hinges on the judge's prediction of what the defendant would do if released. This is a promising machine learning application because it is a concrete prediction task for which there is a large volume of data available. Yet comparing the algorithm to the judge proves complicated. First, the data are themselves generated by prior judge decisions. We only observe crime outcomes for released defendants, not for those judges detained. This makes it hard to evaluate counterfactual decision rules based on algorithmic predictions. Second, judges may have a broader set of preferences than the single variable that the algorithm focuses on; for instance, judges may care about racial inequities or about specific crimes (such as violent crimes) rather than just overall crime risk. We deal with these problems using different econometric strategies, such as quasi-random assignment of cases to judges. Even accounting for these concerns, our results suggest potentially large welfare gains: a policy simulation shows crime can be reduced by up to 24.8% with no change in jailing rates, or jail populations can be reduced by 42.0%with no increase in crime rates. Moreover, we see reductions in all categories of crime, including violent ones. Importantly, such gains can be had while also significantly reducing the percentage of African-Americans and Hispanics in jail. We find similar results in a national dataset as well. In addition, by focusing the algorithm on predicting judges' decisions, rather than defendant behavior, we gain some insight into decision-making: a key problem appears to be that judges to respond to 'noise' as if it were signal. These results suggest that while machine learning can be valuable, realizing this value requires integrating these tools into an economic framework: being clear about the link between predictions and decisions; specifying the scope of payoff functions; and constructing unbiased decision counterfactuals.

Jon Kleinberg
Department of Computer Science
Department of Information Science
Cornell University
Ithaca, NY 14853
kleinber@cs.cornell.edu

Himabindu Lakkaraju
Stanford University
Department of Computer Science
Gates Computer Science Building
353 Serra Mall
Stanford, CA 94305
himalv@stanford.edu

Jure Leskovec
Stanford University
Department of Computer Science
Gates Computer Science Building
353 Serra Mall
Stanford, CA 94305
jure@cs.stanford.edu

Jens Ludwig
University of Chicago
1155 East 60th Street
Chicago, IL 60637
and NBER
jludwig@uchicago.edu

Sendhil Mullainathan
Department of Economics
Littauer M-18
Harvard University
Cambridge, MA 02138
and NBER
mullain@fas.harvard.edu

1 Introduction

Machine learning is surprisingly effective at a wide variety of tasks traditionally associated with human intelligence, from recognizing faces in photos to scoring written essays. If these tools can be applied successfully to basic tasks involving human vision and language, why not to more complex human decisions as well? We examine the promise and the pitfalls of such tools within the context of an important judicial decision: Every year in the United States, the police arrest over 10 million people (FBI, 2016). Soon after arrest, judges decide whether defendants must wait in jail while their legal fate is being decided. Since cases can take several months on average to resolve, this is a consequential decision for both defendants and society as a whole.[1]

Judges are by law supposed to base their bail decision solely on a prediction: what will the defendant do if released? Will he flee? Or commit another crime? Other factors, such as whether the defendant is guilty, do not enter this decision. The reliance of the bail decision on a prediction makes this an ideal application that plays to the strengths of machine learning.[2] A face recognition algorithm, for example, is trained on a dataset of photos with and without faces; it produces a function that 'predicts' the presence of a face in a given photo. Analogously, we apply a machine learning algorithm— specifically, gradient-boosted decision trees—trained on defendant characteristics to predict crime risk. Our algorithm makes use of a dataset of $758,027$ defendants who were arrested in New York City between 2008 and 2013.[3] We have detailed information about defendants, whether they were released pre-trial, and if so, whether they went on to commit a new crime.[4] In this case, the inputs (the analogue of the photo in a face recognition application) are the defendant attributes available to judges when they decide: prior rap sheet, current offense, and so on. The algorithm outputs a prediction of crime risk.[5]

The goal of our paper is not to identify new ways of optimizing the machine learning

[1] For example, the average length of stay is about two months in New York City; see report. Annualized costs of jailing a person are on the order of $30,000. Jail spells also impose costs on defendants in terms of lost freedom (Abrams and Rohlfs, 2011), impacts on families, higher chance of a finding of guilt, and declines in future employment (Dobbie et al., 2016).

[2] This application connects to the pioneering work by Richard Berk on using machine learning to predict crime risk; see, for example, the summary in Berk (2012). A different strand of work in economics focuses on using machine learning tools for causal inferences; see, for example, Belloni, Chernozhukov and Hansen (2014) and Athey and Imbens (2016).

[3] As is standard, we randomly partition our data into a *training* and *hold-out* set to protect against over-fitting. All results for the performance of the algorithm are measured in the hold-out set.

[4] In most jurisdictions judges are asked to focus on both the risk of failing to appear at a required future court date (which can lead to a new criminal charge) and re-arrest. However under New York state law, judges in bail hearings are only supposed to consider flight risk (failure to appear, or FTA). Unless otherwise indicated, in what follows, for convenience, we use the term "crime" to refer to failure to appear in the New York data, and to either failure to appear or re-arrest in the national data. We show in Section 6 that our results for the relative performance of the algorithm versus judge decisions carry through irrespective of the specific definition of "crime" that we use.

[5] In what follows, by " prediction" we will always mean the output of a probability rather than a binary outcome as sometimes seen in classification models.

algorithm. Instead our goal is to understand whether these machine learning predictions can help us understand and improve judges' decisions. Answering these questions raises significant econometric challenges uncommon in typical machine learning applications; indeed most of our work begins *after the machine learning prediction is generated.* For example, while we can make predictions for all defendants, we only have outcomes for the released defendants. We do not know what jailed defendants would have done if released. We develop an empirical approach to overcome these problems and meaningfully compare human decisions to machine predictions.

A first step in this comparison is to ask if there is even a significant difference between them. In fact, we find that judge's decisions often do not comport with the algorithm's predictions. Many of the defendants flagged by the algorithm as high risk are treated by the judge as if they were low risk. For example, while the algorithm predicts that the riskiest 1% of defendants will have a 62.6% crime risk, judges release 48.5% of them. Of course the algorithm's predictions could be wrong: judges, for example, could be astutely releasing the subset of this group they know to be low risk. But the data show that those defendants the algorithm predicted to be risky do in fact commit many crimes: they fail to appear for their court appearances at a 56.3% rate, go on to commit other new crimes at a 62.7% rate, and even commit the most serious crimes (murder, rape and robbery) at a 4.8% rate.

While mis-prediction by judges is one candidate explanation for these findings, another is that judges set a high risk threshold for detention–they knowingly releasing these high risk defendants. With data only on judges' decisions, but not on their predictions of risk, we cannot evaluate this possibility directly. We can, however, evaluate it indirectly. Suppose we have two judges, one lenient and one strict, who hear similar cases. The lenient judge by definition sets a higher threshold for detention. If judges jail by descending risk then the stricter judge should simply lower that bar. That is, she ought to jail the riskiest people released by the most lenient judge before jailing anyone else. So we can see if judges' implicit risk rankings match the algorithm's by looking at who they jail as they become stricter. We can empirically implement this exercise using a tool from the causal inference literature: which judge hears a case is essentially an arbitrary consequence of who was working in the courtroom when the case was heard, and so (as we demonstrate) is as-good-as randomly assigned.[6] This use of as-good-as-random assignment to judges

[6]As we discuss below, in our data we find that conditional on borough, court house, day of week and calendar year, observable defendant characteristics appear to be uncorrelated with judge leniency. For applications of this design in the causal inference literature see for example Kling, 2006, DiTella and Schargrodsky, 2013, Aizer and Doyle, 2015, Mueller-Smith, 2015, Bhuller et al., 2016, Dobbie et al., 2016, Gupta et al., 2016, Leslie and Pope, 2016, and Stevenson, 2016.

is essential to overcoming the inference problem created by not knowing what the jailed defendants would have done if released.

Performing this exercise suggests that judges are not simply setting a high threshold for detention but are mis-ranking defendants. Stricter judges do not simply jail the riskiest defendants; the marginal defendants they select to detain are drawn from throughout the entire predicted risk distribution. Judges in the second quintile of leniency, for example, could have accomplished their additional jailings by simply detaining everyone in the riskiest 12.0% of the defendant risk distribution. Instead only 33.2% of their additional jailings come from this high-risk tail; the remaining additional detentions are to lower-risk people. Similar mis-rankings occur throughout the judge-leniency distribution.

These mis-rankings prove costly. Relative to the most lenient quintile, stricter judges increase jailing rates by 13.0 percentage points and reduce crime by 18.7%. Had they detained in order of predicted risk, they could have achieved the same reduction in crime while increasing jailing rates by only 6.3 percentage points. Alternatively, keeping their increase in the jailing rate at 13.0 percentage points, detaining in order of predicted risk could have reduced crime by 32.9% instead of 18.7%. Put differently, this means we could have jailed only 48.2% as many people or had a 75.8% larger crime reduction.[7] These mis-rankings suggest potentially large welfare gains from using algorithmic predictions in the ranking and release process.

We perform several counter-factuals to scope the size of potential gains that could result from specific policy changes. One that we study extensively is re-ranking: what would happen if *all* release decisions were based on predicted risk? Re-ranking requires a shuffling of the judge decisions—jailing some people the judges released, and releasing some they jailed. While the effect of one counter-factual is easy to quantify—jailing someone who had been released—the other is not: what crimes might the jailed commit if they had been released instead? In documenting the extent of judicial mis-prediction, we have, so far, carefully avoided this problem by focusing solely on the effects of jailing additional defendants. The policy simulation of re-ranking however also requires releasing jailed defendants. As a first step in addressing this problem we impute outcomes for these defendants using outcomes of other defendants with similar observables who judges did release.[8] This allows us to quantify the release rate / crime rate trade-off.

[7] One might worry that these results are due not to the power of the algorithm but to the most lenient judges having some unique capacity to screen out particularly bad defendants; we present evidence in Section 4.2 in fact that this is not the case.

[8] In practice, we implement this approach by randomly dividing our data into training, imputation, and test sets. As before, the risk predictions are formed on the training set and they are evaluated on the test set. But when an imputation is needed, we use an independent imputation set. This ensures that the risk prediction is not unduly benefiting from being constructed from the same data as the imputation procedure.

How this trade-off is made depends on society's preference, but current judge decisions suggest two natural points to consider. First, at the same release rate as what the judges currently choose, the algorithm could produce 24.7% fewer crimes. Second, to produce the same crime rate as judges currently do, the algorithm could jail 41.8% fewer people. These comparisons are useful because they illustrate the algorithm's gains without imposing any preferences on how to trade off crime and jail.[9] Of course these calculations rely on a strong 'selection on observables' assumption; unobservable variables seen by judges could bias our results. Several pieces of evidence, including from the quasi-random assignment of cases to judges, suggest that these biases are likely not very large; for example, we find qualitatively similar results even under fairly extreme assumptions about the importance of unobservables.[10] As a whole, our estimated percentage gains are large in total magnitude because they apply to a large base: at any point time, there are over 750,000 people in local jails across the US.[11]

Both our primary findings and these policy counter-factuals could be misleading if judges (and society) care about outcomes other than crime. The algorithm may then reduce crime but at the expense of these other goals. There are two possible such confounds. The first is that judges may also care about racial equity. Though the algorithm does not use race as an explicit input, race may correlate with other inputs. So one might worry that the algorithm reduces crime simply by aggravating racial disparities. Yet the opposite appears to be true in our data. An appropriately done re-ranking policy can reduce crime and jail populations while *simultaneously* reducing racial disparities. In this case, the algorithm is a force for racial equity.

The second potential confound arises if judges have a more complicated set of preferences over different kinds of crimes than we have assumed. Perhaps we are reducing overall crime while increasing the specific kinds (say, very violent crimes) that judges may care most about.[12] Yet we find similar reductions in all crime categories, including even the most serious violent crimes. Different types of crime are correlated enough so that, for example, training solely on flight risk—which we will also refer to as FTA (failure to

[9]These numbers are not meant to suggest a policy in which algorithms actually make these decisions. Surely in practice they would be used instead as decision aids, and of course judges may not comply fully with these tool's recommendations. Our calculations are simply intended to highlight the scope of the potential gains. Understanding the determinants of compliance with prediction tools is beyond the scope of this paper, though some work has focused on it (Dietvorst et. al. 2014; Yeomans et. al. 2016).

[10]We also consider other sources of confounding. For example, in Section 6.1 we address (i) the possibility that judges face capacity constraints in jail; (ii) address temporal concerns by training the algorithm in one period and evaluating it fully on a later period; and (iii) examine whether our results are due to the fact that judges in practice in setting money bail may also have to predict what bail amount different defendants could pay.

[11]As presented in a report by the Vera Institute.

[12]In New York, for example, suppose judges also care about public safety risk and not just flight risk as New York state law dictates (and our algorithm initially focuses on).

appear)—reduces all other crimes as well, including violent crime. Of course, if we were to train the algorithm to predict violent crimes instead of FTA, we could do even better: for example, we could cut the murder, rape and robbery rate by 57.3% without increasing the number of people detained.[13]

Why, then, are judges mis-predicting? We find suggestive evidence that their judicial decisions are too variable. In particular, we train an algorithm to predict whether judges will release a defendant with a given set of characteristics or not — as before using the defendant characteristics as inputs, but now with the judge's decision to release or not as the outcome to be predicted. In any one case, when the judge's decision deviates from this prediction, some variable unobserved to us must have influenced the judge. A common economic assumption is that these unobserved variables reflect private information. Judges see many variables that are not recorded, and so must make better decisions than the predicted judge—which in this view is merely an information-impaired version of themselves. To test this we build a release rule based on the predicted judge, which orders the defendants by the algorithm's predicted probability that the judge will release them, and then releases the defendants in this order. We find this predicted judge outperforms the actual judges by a wide margin.[14] Whatever these unobserved variables are that cause judges to deviate from the predictions—whether internal states, such as mood, or specific features of the case that are salient and over-weighted, such as the defendant's appearance—they are not a source of private information so much as a source of mis-prediction. The unobservables create noise, not signal.[15]

Our findings also do not appear to be unique to New York. We replicated our analysis on a national data set that covers over 151,461 felony defendants arrested between 1990 and 2009 in 40 large urban counties across the country. In most of these states (unlike New York), judges are asked to focus on public safety risk as well as flight risk. So we now train the algorithm to predict *any* infraction: either flight or rearrest for some other crime. In these new data, with broader geographic coverage and a different outcome variable, we find a qualitatively similar pattern of results to those from New York. We again find judges releasing predictably very risky defendants. And we again find that

[13] In other jurisdictions, judges are asked to focus on flight *and* public safety risk. So, when we use the national data below, we predict any crime (FTA or re-arrest). Still, a related problem arises. If we think of the judge as having an implicit cost function for crime, based on specific weights for flight risk and public safety risk, it is not enough to improve on the cost function using some weighting system that does not necessarily conform to the one used by the judge. In our analysis of a national dataset we again find that training an algorithm on a single candidate outcome does better relative to current judge decisions on all the crime categories, suggesting the algorithm does better whatever the weights.

[14] Dawes (1971) refers to this as 'judgmental bootstrapping'.

[15] One strand of research, for example, emphasizes how highly available and salient information is over-weighted (Kahneman and Tversky 1974; Bordolo et al. 2010). Relatedly, a consistent finding in finance is that asset prices respond to noise as if it were signal (Shiller 1981; Black 1986; Mendel and Shleifer 2012).

policy simulations suggest large potential gains relative to current judge decisions: the algorithm could reduce crime by 18.8% holding the release rate constant, or holding the crime rate constant, the algorithm could jail 24.5% fewer people.

The bail application provides a template for when and how machine learning might be used to analyze and improve on human decisions. A key lesson from our analysis is that such prediction problems require a synthesis of machine learning techniques with several methods that are central to the economics toolkit. First, key to our analysis is a focus on a specific decision - and, importantly, on a decision that relies on a prediction. Many applied empirical papers implicitly focus on informing decisions where the key unknown is a causal relationship; for example, the decision to expand college scholarship eligibility depends on the causal effect of the scholarship. The causal effect of jail on flight risk, though, is known. What is unknown is the flight risk itself. The bail decision relies on machine learning's unique strengths—maximize prediction quality—while avoiding it's weakness: not guaranteeing causal or even consistent estimates (Kleinberg et. al. 2015).[16] [17]

Second, our analysis needed to account for the full payoff function. In causal inference, biases arise when omitted variables correlate with the outcome. By way of contrast, for prediction, biases arise when omitted variables correlate with *payoffs*. Predictions based on only one of the variables that enter the payoff function can lead to faulty conclusions. We worried, for example, that our improved performance on crime was being undermined by creating racial inequity. The problem is put into particularly sharp relief by considering a different decision that on its surface seems very similar to bail: sentencing. Recidivism, which is one of the inputs to deciding the punishment for someone who has been found guilty, can be predicted. Yet many other factors enter this decision—deterrence, retribution, remorse—which are not even measured. It would be foolish to conclude that we can improve on sentencing decisions simply because we predicted recidivism. Outperforming the decision maker on the single dimension of recidivism risk would not imply the decision-maker is either mis-predicting, or that we can improve their decisions. We chose bail explicitly because these *omitted payoff* biases are specific and narrow in scope.[18]

[16] Additionally, it is worth noting that while these tools are often thought to be useful primarily in applications where there are large numbers of predictor variables ('wide data'), our bail study makes clear these tools can also be useful in a broader range of settings. Our bail dataset has not more than a few hundred variables, yet our algorithm still seems to perform well relative to both judges and (as noted below) standard econometric methods. Our analysis thus helps illustrate a subtle implication of machine learning's ability to consider a wide range of very flexible functional forms without over-fitting the data. For these machine learning tools, the 'width' of a dataset is really defined by the set of possible *functions* of the predictors, not the number of predictors. Machine learning can add value even without large numbers of variables.

[17] It would be inaccurate, though, to suggest causal inference plays no role here. Assessing the full impact of judicial decision-aids, for example, requires answering causal questions such as how judges respond to them.

[18] This does not mean that risk tools might not aid in other decisions such as parole or sentencing. For example Richard Berk and his colleagues have shown risk to be predictable in diverse criminal justice contexts; see for example the summary in Berk (2012) and subsequent results in Berk and Bleich (2013) and Berk et al. (2014). It merely means that we cannot

Finally, evaluating whether machine predictions improve upon human decisions often requires confronting a basic selection problem: outcome data ('labels' in machine learning parlance) can be missing in a non-random way. Importantly, the availability of outcome data depends on the very decision-maker we are trying to improve upon: the judge's release decisions determine the people for whom we see crime outcomes.[19] As we have seen, this *selective labels* problem complicates our ability to compare human judgments and machine predictions. In our example we exploited the fact that it is one-sided: we could quantify with confidence the effect of jailing defendants the judges released. The challenge was quantifying the effect of releasing jailed defendants. We crafted our analysis to take advantage of this one-sidedness, by exploiting a natural experiment in which cases are as-good-as-randomly assigned to judges who differ in their release rates. We could then compare what happens when more defendants get jailed by the algorithm rather than by stricter judges. In practice machine learning analyses often ignore selective labels problems. But without resolving this challenge it is hard to compare human decisions to algorithmic predictions.[20]

These methodological challenges are why machine learning techniques need to be integrated with methods for analyzing decision-making. Through a narrow lens, the focus of machine learning may seem to be solely about increasing prediction accuracy. In behavioral applications, though, predictions by themselves provide little value. They only become useful when their role in decision-making is made clear and precise counterfactuals whose welfare gains can be calculated are constructed. The foundational efforts in psychology comparing human predictions to statistical rules (e.g. Dawes, Faust and Meehl, 1989) suffered from the same issues. They largely ignored selective labels and, to a lesser degree, they also ignored omitted payoff biases.[21] While this earlier work proved visionary, given these biases it is hard to judge the robustness of the statistical evidence. It is telling that in our application, much of the work happens *after* the prediction func-

conclude that decision-making is sub-optimal on the basis of evidence about predictability alone.

[19]Bushway and Smith (2007) made a similar observation about the challenges posed by the 'treatment rule implicit in existing data' (p. 379) in the context of explaining why skepticism within criminology about the predictability of criminal behavior may be premature. Their focus was on the role of observable characteristics; if, for example, age is negatively related to risk, but judges assign all youthful offenders to some program that substantially reduces their risk, we may observe little to no correlation between age and risk in the data we observe that was generated by the judge decisions.

[20]These problems are common when applying prediction methods not just to social science applications, but to more traditional machine learning applications as well. For example, recommender systems, such as those used by Netflix, are always generalizing from instances where they have labels (people who have watched movies, or reacted to algorithmic recommendations) to situations where there are none (what if we encouraged this person to watch a movie they otherwise would not have known about?).

[21]For example, looking only at those who were hired in a recruiting application. Similarly, while some examples under study (medical diagnoses) had few other elements in the payoff, many other examples (hiring) surely did. An algorithm or statistical rule that improves upon a single, narrow dimension of hiring or admissions does not necessarily improve overall welfare.

tion was estimated to deal with selective labels and omitted payoffs. This focus is a key differentiator from these earlier lines of research.

In fact one might even ask to what degree was machine learning itself central to this work? Practically, we find that our machine learning algorithm provides a 25% larger gain in crime or jail reductions than logistic regression would have, with gains in prediction accuracy that are particularly pronounced in the tails of the risk distribution. This is a sizable benefit, especially as it comes at close-to-zero marginal costs. At the same time, it is noteworthy that a standard logistic regression also produces gains over the judge. While part of the gain from machine learning comes from the machinery it provides, the other part comes from the kind of problem it leads us to focus on. Micro-empirical work typically neglects prediction problems such as the jail-or-release decision. Our results suggest studying these problems could be valuable; and that machine learning can be a useful tool in this endeavor, when integrated into an economic framework that focuses on payoffs, decisions and selection biases.

2 Data and Context

2.1 Pre-trial bail decisions

Shortly after arrest, defendants appear at a bail hearing. In general judges can decide to release the defendant, set a dollar bail amount that the defendant needs to post in order to go free, or to detain the defendant with no chance of posting bail.[22] As noted above, these hearings are *not* about determining whether the person is guilty, or what the appropriate punishment is for the alleged offense. Judges are asked instead to carry out a narrowly defined task: decide where the defendant will spend the pre-trial period based on a prediction of whether the defendant, if released, would fail to appear in court ('FTA') or be re-arrested for a new crime.

In cases where the judge is setting money bail, they are technically making two predictions - the defendant's crime risk and their ability to post a given bail amount.[23] For simplicity we treat these as a single compound decision - release or not. This could have

[22] The 'outright release' decision can involve either a release on recognizance, or release on an unsecured bond that does not require the defendant to put any money down but the defendant would be liable for the bond amount if they skip a future required court date. The bail requirement can come in different forms, from requiring the defendant to put down cash equal to some percentage (such as 10%) of the bond's value, to full cash or property bonds that require defendants to put down the full amount. Some jurisdictions also allow private bail bondsmen to help people post bond in exchange for a fee. Defendants can also sometimes be released with conditions, such as electronic monitoring.

[23] Judges do not select arbitrarily low bail amounts for people they believe to be of low risk of flight or re-offending because a non-trivial bail amount even in these cases creates an incentive for the defendant to show up in court rather than lose the posted bond amount.

several possible effects on the error rate: judges may be making mistakes in either of their two predictions (crime risk and ability to pay); but at the same time, forcing the algorithm to make a single decision narrows its choice set, which on the surface should limit its performance relative to a broader space of available choices. We show below that how we handle this does not matter much in practice.

The information judges have available to them when making these decisions includes the current charges for which the person was arrested (the 'instant offense') and the defendant's prior criminal record ('rap sheet'). In some places, pre-trial services interviews defendants about things that may be relevant for flight risk, such as employment status or living circumstances. Of course the judge also sees the defendants, including their demeanor and what they are wearing (which is typically what they wore at arrest).

The context for most of our analysis is New York City, which has the advantage of providing large numbers of observations and was able to provide data that identifies which cases were heard by the same judges. Yet the pre-trial system in New York is somewhat different from other places. First, New York is one of a handful of states that asks judges to *only* consider flight risk, not public safety risk.[24] So we focus our NYC models on FTA, though below we also explore what happens when we also consider other crime outcomes. Second, in New York many arrestees never have a pre-trial release hearing because either the police give them a desk appearance ticket,[25] or the case is dismissed or otherwise disposed of in bond court. So we must drop them from our analysis. Third, judges in New York are given a release recommendation based on a six-item check-list developed by a local non-profit,[26] so our analysis technically compares the performance of our algorithm against the combined performance of the judge plus whatever signal they take from this existing check-list tool. We return to this below. To determine how important any of these local features are, we replicate our analysis in a national dataset as discussed in detail below.

2.2 Data

We have data on all arrests made in New York City between November 1, 2008 and November 1, 2013. The original data file includes information about 1,460,462 cases.

[24] See Phillips (2012 p. 25, 53). Another way New York City is different is that private bail bondsmen and supervised release programs are relatively less common (Phillips 2012, p. 33, 41)

[25] At the station the police issue the arrestee a ticket that includes information about when the next required court date is, and then release the person.

[26] The six items on the tool developed by the New York City Criminal Justice Agency, Inc. (CJA) capture whether the defendant has a phone, a NYC-area address, an activity that occupies them full-time (such as school or a job), any prior bench warrants, or open criminal cases, and whether the defendant expects someone to come to court to support them; see New York City Criminal Justice Agency, Inc. (2016, p. 14).

These data include much of the information available to the judge at the time of the bail hearing, such as the instant offense[27] and the rap sheet, which includes prior FTAs. The dataset also includes the outcome of each case, including pre-trial release, FTA, and any re-arrest prior to resolution of the case. The only measure of defendant demographics we use to train the algorithm is age.[28]

Of the initial sample, 758,027 were subject to a pre-trial release decision and so relevant for our analysis.[29] We then randomly select 203,338 cases to remain in a "lock box" to be used in a final draft of the paper; it is untouched for now.[30] This leaves us with a working dataset for training and preliminary evaluation of our algorithm of 554,689 cases.

[Table 1 about here.]

Table 1 presents descriptive statistics for our analysis sample. As is true in the criminal justice systems of many American cities, males (83.2%) and minorities (48.8% African-American, 33.3% Hispanic) are over-represented. A total of 36.2% of our sample was arrested for some sort of violent crime (nearly two-thirds of which are simple assaults), 17.1% for property crimes, and 25.5% for drug crimes, and the remaining arrests are for a mix of miscellaneous offenses such as driving under the influence, weapons, and prostitution. Fully 73.6% of defendants were released prior to adjudication. Of those released, 15.2% fail to appear, while 25.8% are re-arrested prior to adjudication, 3.7% are arrested for a violent crime, and 1.9% are arrested for the most serious possible violent crimes (murder, rape, and robbery).

Table 1 also makes clear judges are paying some attention to defendant characteristics in deciding who to release, since the average values differ by release status. Exactly how good judges are in making these decisions relative to an algorithm's predictions is our focus for the rest of the paper.

We also carry out an analysis below that takes advantage of the fact that we can identify which cases were heard by the same judge, and that cases are quasi-randomly assigned to judges within the 1,628 unique borough, courthouse, year, month and day of week 'cells' within our dataset. For this analysis we restrict our attention to the 577 cells that contain at least five judges in order to do comparisons across within-cell judge-leniency quintiles. These cells account for 56.5% of our total sample, with an average of

[27] Unlike the judge, we only have detailed information on the most serious charge filed against the defendant, not all charges.

[28] Previous research demonstrates a strong age patterning to criminal behavior, and courts have generally found consideration of age to be legally acceptable.

[29] We exclude 272,381 desk appearance tickets, as well as the 295,314 cases disposed of at arraignment, the 131,731 cases that were adjourned in contemplation of dismissal, and eliminate some duplicate cases.

[30] The hold-out set is constructed by randomly sampling some judges and taking all of their cases, selecting a random selection of cases from the remaining judges, and also putting the last 6 months of the data into the hold-out set.

909 cases and 12.9 judges per cell. Appendix Table A1 shows this sample is similar on average to the full sample.

3 Empirical Strategy

Our empirical analysis essentially consists of two steps. First, we train a prediction algorithm using a set of training data in which each individual instance—corresponding to a defendant in this case—is described by a set of features and an outcome. The algorithm is designed to predict the outcome given the features. We then take this prediction algorithm to new data and assess its accuracy. In the typical machine learning literature, the second step would consist simply of reporting some measure of how the predictions fit the true values of the outcome. However we instead are interested in what those predictions tell us about the quality of current judge decisions, and whether using the algorithmic predictions can improve those decisions. So the first stage of our analysis will look quite similar to standard machine learning practice. But the second stage, evaluation, will look quite different.

3.1 The Machine Learning Black Box

For our purposes here, machine learning can be viewed as a "black box" methodology that provides the following type of guarantee. We provide it with (i) a set of instances, each described by a set of input features x and an outcome y to be predicted; (ii) a class of allowable functions m that can be used to build a prediction for y of the form $\hat{y} = m(x)$; and (iii) a loss function $L(y, \hat{y})$ that quantifies the cost of prediction errors. In our case, the functions $m(x)$ we consider will generate probability values in the interval from 0 to 1, and the loss function will penalize deviations of these values from the corresponding outcome variable y. The goal is to use the minimization of the loss to guide the search for a prediction function $m(x)$ that generates accurate predictions out of sample. So procedurally, to avoid over-fitting we evaluate the performance of the prediction function in some hold-out or test set that is separate from the dataset used to train the prediction algorithm itself. Figure 1 provides a schematic representation of these basic elements.

[Figure 1 about here.]

The first step of this procedure is to partition the data. Our working dataset consists of 554,689 observations, which we randomly partition into a 80% training set of 443,751 observations that we use to build the algorithm and a 20% hold-out test set of 110,938

observations on which we measure the algorithm's performance as reported in this draft. As shown in Figure 1, following Tan, Lee and Pang (2014) we also form a 'pure hold-out,' which is not used anywhere in this draft. At the final stage of our analysis, after final revisions, we will show results on this pure hold-out set. This allows us to change the algorithm based on feedback, without fear that we are data mining the test set.

The algorithm is trained using the Bernoulli loss function:

$$L[(y_i, m(x_i)] = -[y_i \times log(m(x_i)) + (1 - y_i) \times log(1 - m(x_i))]$$

There are many candidate algorithms one could use. Indeed the bewildering diversity of choices creates an apparent challenge.[31] Yet there is a connective tissue across (nearly) all of the algorithms. This can be seen by way of contrast with a familiar "algorithm", the logistic regression. The logit would use a set of coefficients ($\hat{\beta}$) to fit the dependent variable y ("crime") using a set of pre-specified explanatory variables x, which in our case is a set of categorical variables for age, current charge and past criminal record. This differs from most machine learning models in one crucial way. The set of x variables is always pre-specified, is usually a small number, and the functional form is fixed. Consequently, absent researcher discretion, neither the number of independent variables nor the functional form for how they're used changes with the data size, or any other feature of the data.

Machine learning algorithms, in contrast, typically allow for a free complexity parameter (or parameters). For example, a LASSO version of the logit would estimate a predictor with non-zero coefficients on only some of the independent variables. The complexity parameter in this case would determine the extent of this "sparseness" (number of non-zero β_j). As another example, in a decision tree, the depth of the tree or the number of leaf nodes determines its complexity. In this case, the interactivity of the final predictor is determined by this complexity parameter. With gradient boosted trees, which we use, complexity is measured by a combination of the depth of each tree, the number of trees that are fit in the sequence, and the weight given to subsequent trees in the series (the 'learning rate'). Whatever the particular algorithm, the ability to choose complexity is a crucial ingredient.

An important practical breakthrough with machine learning is that the data themselves can be used to decide the level of complexity to use. This procedure allows one to estimate non-parametric functions where the 'complexity' is decided on as part of the estimation procedure. In the LASSO logit example, how many coefficients are non-zero is determined by the data itself. This feature of machine learning allows researchers to begin with a

[31] For some examples, see Friedman, Tibshirani and Hastie (2001) and Murphy (2012).

much richer array of potential candidate variables—such as many higher order interaction terms in a linear model—in the initial specification and let the algorithm endogenously decide which to use.

The intuition behind this procedure is simple. More complex functions will generally fit the data better but they will also overfit the data, responding to the idiosyncratic features of any given sample. The key is in understanding whether increasing complexity is leading to better *out of sample* fit. To do this, one uses the given data to create an 'out of sample' test as far as the algorithm is concerned. As shown in Figure 1, specifically, the data is partitioned (into 'folds'); in our case, five folds of 88,750 observations each. In four of the five folds, the algorithm is fitted for each possible value of the complexity parameter. In the fifth fold, performance is evaluated at each level of the complexity parameter. Since data in the fifth fold were not used to estimate the algorithm, any over-fit becomes apparent. Typically, one finds that increasing complexity improves fit and then as the algorithm begins to overfit badly, decreases fit. The chosen complexity parameter is the one that yields the lowest prediction loss in the held-out fold.[32]

As a result the data help determine what level of complexity creates best predictive performance. One consequence is that as data size grows, one can fit more complex models. The link between data size and model complexity has led to a rule of thumb in the literature: more data trumps better algorithms. Despite the variety of algorithms to choose from, in practice the choice of algorithm typically matters less for prediction accuracy than does the sheer volume of data available to train the algorithm, which allows all algorithms to fit more complex functions (see for example Banko and Brill, 2001). So, in our application, for example, the large amount of data we have might allow for high levels of interactivity between the fifty or so variables we have—possibly providing more predictive power than a linear regression (or logit) that uses these variables as main effects would allow.

In our analysis we use gradient boosted decision trees (Friedman 2001). This algorithm is essentially an average of multiple decision trees that are built sequentially on the training data, with each subsequent iteration up-weighting the observations that have been predicted most poorly by the sequence of trees up to that point.[33] The entire

[32]In practice, one typically repeats this procedure 5 times, so that each fold can serve as the left out fold; we follow this practice. There is nothing special about the partitioning in this way. One can use a bootstrap sample for the same purpose. The key insight here is to evaluate—at different levels of complexity—performance on a sample different from the one used to build the prediction model.

[33]In a decision tree, the data is divided through a sequence of binary splits. For example, the first split might be whether the person has ever committed a crime before. Each successive split can depend on the results of priors splits. At each final ('leaf') node, there is a value which is the prediction for every data point in that space. The depth of the tree determines its complexity. Gradient boosted trees are sums of many trees, each of relatively small depth. Importantly, each additional tree in this sum is estimated by fitting the prediction error of the prior prediction. They are more thoroughly described in

cross-validation procedure provides a set of parameter values for each of these complexity parameters. We then fit a model, on the entire training set, using these parameters. The prediction function is evaluated on a distinct hold-out set, as seen in Figure 1. We use this separate hold-out set for all of our results, rather than using cross-validated performance, so that we can more readily compute standard errors.

3.2 Evaluating the Results

In our bail application both the train and test datasets in Figure 1 include two groups of defendants, those released versus detained by the judges. We build our algorithm using data for released defendants within the training set, for whom we observe both crime outcomes and case characteristics. We can then calculate predicted risk, $m(x_i)$, for all defendants in the randomly-selected test set.

[Table 2 about here.]

Table 2 provides some intuition about what variables are important for predicting risk in the NYC data, and the degree to which the ability of machine learning to consider flexible functional forms is adding value. A regression of the algorithm's predicted values against a linear additive function of the baseline covariates yields an Adjusted R-squared of 0.51, which provides some initial indication that there is non-linear structure in the data that machine learning tools help identify. We show below that this additional non-linear structure captures useful signal.

Now that we have these predictions, we can evaluate them in our test set. Current standard practice in machine learning would be to compare these predictions to actual outcomes (or 'labels') for those members of the test set for whom labels are available. In our application this would be the set of defendants the judges released. A common metric for measuring prediction accuracy would be something like the area under the receiver operating characteristic curve (AUC), which in our case equals 0.707.[34]

The problem with this standard approach is that it tells us nothing about whether the algorithm's predictions can improve on decision quality. Suppose we had a jurisdiction that released (say) 50% of pre-trial defendants. This release rate is a function of judge predictions, which depend on two types of defendant characteristics: (i) a set of 'observable' features that the algorithm sees, corresponding to the features x_i that provide the input

Friedman (2001).

[34]The ROC curve reports the combination of false positives and true positives that can be achieved by the model, assuming the exercise is to classify cases where one-half of the sample has a label of 1 and the other half of 0. An AUC of 0.5 represents what we would expect under random guessing, while an AUC of 1.0 is perfect discrimination.

to the machine-learning algorithm, and (ii) a set of 'unobservable' features as well, which we denote z_i. Thus, the judge's decision is a function $h(x_i, z_i)$ of both the observables and the unobservables, while the algorithm's prediction is a function $m(x_i)$ based only on the observables, as described above. If judges were perfect at rank-ordering defendants by whether they are above versus below median risk (that is, deciding whom to detain versus release), even an algorithm with a very high AUC would be unable to improve on the release decisions of the judges. Alternatively suppose judges were literally releasing defendants at random. Even a weakly predictive algorithm with a low AUC could in that case improve release decisions.

That is, we do not ultimately care about prediction quality: how $m(x_i)$ compares to y_i within the released set. We care about decision quality: using the algorithm to benchmark how well the judge sorts defendants by risk, or $m(x_i)$ versus $h(x_i, z_i)$. Of course we do not directly observe judge predictions, we only observe judge release decisions, $R_i \in 0, 1$. In what follows we compare our machine predictions $m(x_i)$ to the predictions implicit in judge release decisions, under the assumption that judges release in order of their own predictions of defendant risk; that is, $R_i = f(h(x_i, z_i)) = 1(h(x_i, z_i) < h^*)$.

4 Judge decisions and machine predictions

4.1 How risky are the riskiest people judges release?

So who do the judges release? Figure 2 bins defendants in our test set into 1,000 equal-sized groups based on the predicted risk values from the algorithm, $m(x_i) = \hat{y}_i$. The three-dimensional figure graphs three values for each group: the predicted crime risk, the release rate and the observed crime rate. The projections on to the sub-spaces therefore give us both the calibration curve (predicted crime rate against observed crime rate) and the judge's release curve (release rate against predicted crime rate). Additionally, we have colored each bin by the fraction of defendants in that bin who have a prior failure to appear (FTA) at a required court date. As we saw in Table 2 we see in this figure again: prior failure to appear is a large component of predicted crime risk.[35]

We see that at the low end of the predicted risk distribution, where most defendants are concentrated, judges release at a rate of over 90%. As predicted risk increases the judge

[35] As a reminder because New York state law asks judges to only focus on FTA risk in making pre-trial release decisions, our algorithm is trained to predict FTA only (not re-arrest). For convenience, unless otherwise specified we use the term 'crime' to refer to what the law asks judges to focus on in making pre-trial release decisions. In our New York data this is FTA, although below we show the results hold for measures of re-arrest as well, while in our national replication this is either FTA or re-arrest.

release rate declines, which implies that the predictions of the judges and the algorithm are correlated.[36]

[Figure 2 about here.]

But we also see that the algorithm and the judges disagree, particularly at the high end of the risk distribution. If the predictions of the judges and the algorithm were identical, we would expect to see a step function in Figure 2: There would be some predicted-risk threshold where the release rate would be 0 above and 1 below. But that is not what we see. The curve relating judge release rates to the algorithm's predicted crime risk flattens out as predicted risk increases. Among the riskiest 1% of released defendants, who have a predicted risk of 62.6%, the judge releases them at a rate of fully 48.5%. The judges are treating many people with high predicted risk as if they were low risk.[37]

Of course the algorithm's predictions are just predictions. In principle these defendants could actually be low risk, and the judges might realize this even if the algorithm does not. That is, perhaps the judges are able to identify defendants who look low risk with respect to the characteristics available to the algorithm, x_i, but are actually high risk with respect to features that only the judges see, z_i.

But Figure 2 also shows that the people the algorithm predicts are risky are indeed risky. This is easiest to see in the projection of the curve onto the two-dimensional space at the bottom of the figure relating observed crime rates to predicted risk, $E[y_i|m(x_i)]$, among released defendants. This plot lies almost exactly along the 45 degree line over the full risk distribution. The defendants the judges release do not seem to have unusual unobservables that cause their observed outcomes to systematically diverge from what the algorithm had predicted.

This provides more than just evidence that our algorithm's predictions are well calibrated; it shows that the defendants judges released who were predicted to be high risk are in fact high risk. We see, for example, that using just information the judge had at the time of the bail hearings, the defendants predicted to be riskiest by the machine learning algorithm—the riskiest 1%—go on to have an *observed* crime rate of $\bar{y} = 56.3\%$.

[36] If the judge release decisions are based on their prediction of defendant risk then $R_i = f(h(x_i, z_i))$. The judge's release rate conditional on the algorithm's predicted risk is $E[f(h(x_i, z_i))|m(x_i)] = P[h(x_i, z_i) < h^*|m(x_i)]$, so $Corr(R_i, m(x_i)) < 0$ implies $Corr(h(x_i, z_i), m(x_i)) > 0$..

[37] In principle a different reason why we might not see a clean division of defendants released versus detained around some risk threshold is if the different judges hearing cases in our dataset each use a different threshold. In that case at any given predicted risk level we would see some defendants released and some detained because that risk level would be above the threshold used by some judges but below that used by others. But this could only explain the pattern we see in Figure 2 if some judges basically released almost everyone and other judges detained almost everyone, since we see judges releasing people from throughout the entire predicted risk distribution. In practice we do not see this much variation across judges in their release rates.

This also provides an opportunity to explore the value-added of machine learning over using more standard econometric methods for making these predictions. Table 3 compares the predicted risk distribution of the ML algorithm to that produced by a logistic regression; specifically, we compare the cases flagged as risky by either procedure. Each row of the Table uses a different threshold, from the 1^{st} percentile of risk (row 1) to the 25^{th} percentile of risk (row 4). At the 1^{st} percentile, we see substantial disagreement in who is flagged as risky—only 30.6% of cases are flagged as top percentile in the predicted risk distribution by both our ML algorithm and logistic regression (column 1). These defendants identified as high risk by both procedures also have the highest *realized* crime rates (60.8% in column 3). Those flagged only by the ML algorithm are nearly as risky (54.4% in column 2), while those flagged only by the logit are far less risky (40% in column 3). As a result, ML-flagged defendants (column 4) are riskier as a whole than logit-flagged ones (column 5). This pattern repeats in the other rows but begins to attenuate the further we move down the predicted risk distribution (rows 2 through 4). By the time we reach the 25^{th} percentile of the distribution (row 4) the two procedures agree on 73% of the cases. As a whole, these results suggest that even in these data, which contain relatively few variables (compared to sample size), the ML algorithm finds significant signal in combinations of variables that might otherwise be missed. These gains are most notable at the tail of the distribution and (somewhat predictably) attenuate as we move towards the center. This intuition suggests that were we look at outcomes that have relatively lower prevalence (such as violent crimes as we do in Section 6.1.1) the difference in results between the two prediciton procedures would grow even starker.

[Table 3 about here.]

4.2 Mis-ranking or high detention threshold?

Establishing that the marginal defendant as identified by the machine learning algorithm—the available defendant that the algorithm would rank highest for detention—has such a high level of predictable risk is certainly suggestive of mis-prediction by the judge. But as a logical matter there is an alternative explanation as well: Judges may simply place a high cost on jailing defendants, which would lead them to detain only those with even higher risk.

How can we tell whether judges are mis-predicting risk or simply seting a very high risk threshold for detention? If we could directly observe judge predictions of each defendant's risk, $h(x_i, z_i)$, this would be a trivial exercise: We could simply examine whether judges

expected everyone they detained to be of even higher risk than the marginal defendant they released. We cannot test this within any individual judge's caseload because we do not directly observe $h(x_i, z_i)$.

Looking across the caseloads of judges with different levels of leniency (release rates) does allow us to uncover the *implicit* rank-ordering of defendants. In particular, it allows us to quantify the risk of the marginal defendants detained. Suppose for instance that we have two judges who differ in their release rates, equal to say 90% and 80%, and that defendants are randomly assigned to judges. Because we can calculate the algorithm's predicted risk for each defendant in each judge's caseload, we can compare the distributions of predicted risk among the two judge's caseloads to determine where in the distribution the additional defendants jailed by the stricter judge come from.[38]

Two things are worth noting. First, this procedure does not rely on judges having similar rank orderings. For example, we allow for the possibility that the most lenient judge—the one with the higher overall release rate—actually jails more people in some predicted-risk bins $m(x_i)$. Second, this procedure does make a specific assumption about the distribution of unobservables. It requires that, within each predicted risk bin, the different judges have similar distributions of unobservables amongst the released: specifically that their average risk is the same for more and less lenient judges. For example, this procedure will be biased if the most lenient judges are better at using unobservables in deciding whom to release. We examine this assumption in greater detail below.

Our empirical application of this approach in the NYC data takes advantage of the fact that we have (anonymous) judge identifiers, together with the fact that conditional on borough, court house, year, month, and day of week, average defendant characteristics do not appear to be systematically related to judge leniency rates within these cells (see Appendix A).[39] The other thing we need for this design to work are differences in judge leniency within cells. Consistent with past research, that is what we see in our data as well. The most lenient quintile judges release 82.9% of defendants. Relative to the most lenient judge quintile, the less lenient quintiles have average release rates that are 6.6, 9.6, 13.5 and 22.3 percentage points lower, respectively.

[Figure 3 about here.]

The release decisions of the most lenient quintile of judges are shown in the middle panel

[38] That is, if $E[R^L] = 0.9$ and $E[R^S] = 0.8$ are the release rates for the lenient and strict judges, respectively, then at each value of the algorithm's predicted risk we can observe $E[R^L|m(x)]$ and $E[R^S|m(x)]$ and calculate $E[R^L|m(x)] - E[R^S|m(x)] = P[h^L(x_i, z_i) < h^{*L}] - P[h^S(x_i, z_i) < h^{*S}]$.

[39] While neither defendants nor judges are randomly assigned to arraignment hearings, as an empirical matter it appears that on average the caseload within (say) a given Brooklyn courthouse in February 2009 in one Monday looks like another February 2009 Monday's caseload.

of Figure 3; the top box shows the 17.1% of defendants they jailed, the bottom shows the 82.9% they released, both shaded by the algorithm's predicted risk, $m(x_i) = \hat{y}_i$. The left-hand panel of Figure 3 illustrates how the algorithm would select marginal defendants to detain to get down to the second-most-lenient quintile's release rate, which is the pattern we would *expect* to see if judges were detaining defendants in descending order of predicted risk. These marginal defendants could all be selected from the top 12.0% of the defendant predicted-risk distribution; that is, the algorithm would prioritize the highest-risk defendants within the most lenient quintile's released set to detain as we contracted down the released set.

The right-hand panel of Figure 3 shows what judges *actually* do in practice: when judges become stricter, they select their marginal defendants to detain essentially from throughout the predicted risk distribution. The additional cases to detain do come disproportionately from the highest-predicted-risk bins. But instead of selecting all the marginal defendants from the highest-risk 12.0% of the distribution, only 33.2% come from this riskiest tail.[40] The judges are choosing to jail many low-risk defendants ahead of those with higher predicted risk.

[Figure 4 about here.]

Figure 4 shows a similar pattern holds for all of the judge-leniency quintiles as well. We sort defendants by predicted risk and bin them into 20 equal-sized groups. The black segment at the top of each bin shows what share of defendants in that bin is detained by the most lenient quintile judges. The blue segments on the left shows that the algorithm would prioritize for detention people in the highest predicted risk bins if the goal were to lower the release rate from the most lenient quintile's rate down to the second-most-lenient quintile's rate (top panel), or third-most-lenient quintile's rate (second panel), etc. The blue shading on the right shows from which risk bins the judges actually select marginal defendants to detain - essentially from throughout the predicted-risk distribution. It is worth noting that there are no predicted-risk bins where there are more defendants *released* by a stricter judge than by a more lenient judge. The judges do not seem to disagree much with each other about how to rank-order defendants based on their observable characteristics.

This Figure clearly illustrates that all leniency quintiles of judges behave as the second quintile does: jailing predictably low-risk individuals while high-risk ones are available.

[40] Above we showed that the difference in release rates between the most and second-most lenient quintile's caseloads is 6.6 percentage points. The second-most lenient quintile jails only an extra 2.19 percentage points from the top 12.0% of the risk distribution.

The extent of this error is quantified in the first two columns of Table 4. The first column shows how far into the risk distribution judges would need to go were they to jail by predicted risk. The second column shows the fraction that actually comes from this part of the distribution. For example, all the additional jailing of the second quintile could be had by jailing everyone in the top 12.0% highest risk, whereas only 33.2% come from this part of the distribution. This later percentage is relatively common through all the quintiles. As a whole these results show that all quintiles of leniency at best jail only partly according to predicted risk.

[Figure 5 about here.]

Given that the judges and the algorithm disagree about whom to detain on the margin, a natural question is: Who's right? Are the judges using private information, z_i, to identify marginal defendants who have even higher risk than those the algorithm can identify using just observables? This can be tested directly. Starting with the release set of the most lenient judges, we can choose additional incarcerations according to predicted risk. For each amount of additional incarceration, this allows us to calculate the crime rate that we observe for each of these (smaller) release sets. Importantly, because case loads are on average similar across judges, these numbers can be compared to the outcomes produced by the stricter judges. These results are presented graphically in Figure 5 as well. The solid (red) curve calculates the crime that would have resulted if additional defendants had been detained in order of the algorithm's predicted risk. Each of the points denotes the judges of different quartiles. Since any additional detention reduces crime for purely mechanical reasons (defendants incapacitated in jail cannot commit crime), even randomly selecting additional defendants to jail from the most lenient quintile's caseload would also reduce crime, as shown by the dashed line in the figure.[41]

[Table 4 about here.]

When comparing each of the stricter quintile judges to what the algorithm could achieve, two points are particularly salient: (i) how much does crime fall when the algorithm increases jailing rates by the same amount; and (ii) what jailing increase does the algorithm need to achieve the same crime reduction as the judge. These numbers are presented in Table 4.

We see significant gains over what judges manage. The second quintile of judges reduce crime by 9.9% by increasing the detention rate by 6.6 percentage points. The same crime

[41] The judge detention decisions are better than random, though one cannot tell whether they are doing much or only modestly better without a counter-factual.

reduction *could* have been accomplished by increasing the detention rate by only 2.8 percentage points, or equivalently by increasing the detention rate by 6.6 percentage points we could have reduced crime by 20.1%. Put differently, relative to the observed judge outcomes we could have reduced the increase in jail population by only 42.1% as much, or increased the size of the crime drop by 103.0%. The magnitudes of these effects diminish somewhat as we move to the other leniency quintiles. By the fifth (least) lenient quintile of judges, we could increase the jail population by only 50.3% to get the same crime drop or increase the crime drop by 62.3%. Were we to average across all four of these quintiles we could jail only 48.2% as many people, or we could get crime reductions that are 75.8% larger.

We have established that detaining defendants using a combination of the lenient judge decisions and the algorithm can yield lower crime at a given release rate than what results from the actual decisions of the stricter judges. But perhaps this is not so much a testament to the algorithm as it is to the skill of the most lenient judges, and in particular with respect to their potential skill using unobservables in selecting defendants to detain. That is, the combined jailing decisions of the algorithm and most-lenient quintile judges could dominate those of the less-lenient quintile judges because the most-lenient quintile uses unobservables more effectively than do the other judges - which the algorithm itself could not replicate.[42]

Note that this counter-explanation to our findings suggests a testable hypothesis: If the most-lenient quintile judges were unusually effective in using unobservables to select people to jail, then the people that the most-lenient quintile released would be less crime-prone compared to people with similar observable characteristics who were released by less-lenient judges. This means that if we train an algorithm on the released set of the most-lenient quintile and then used that to impute crime rates to defendants released by the less-lenient quintile judges, the imputed values would be below the actual crime outcomes within the caseloads of the less-lenient judges.[43] Yet what we see is that the imputed and actual values are quite similar on average for each of the stricter judge quintiles, as shown in the bottom four panels of Figure 6. Within each of the stricter quintiles the imputed and actual values are well calibrated across the full range of the

[42]To see the potential problem, suppose a lenient and strict judge are predicting risk using just two binary factors: one observable to the algorithm (x_i); the other unobservable (z_i). We assume random assignment of cases to judges. Let $y_i = \alpha x_i + \beta z_i$ and $\alpha > 0$ and $\beta > 0$. Suppose that among the $x_i = 1$ people, the lenient judge releases all those who have $z_i = 0$ and detains all those with $z_i = 1$. In this example, the crime rate among young people released by the lenient judge will be α, lower than what we see among those released by the strict judge, $(\alpha + \beta E[z_i|x_i = 1])$; i.e., $E[y_i|R_i^L = 1, m(x_i)] < E[y_i|R_i^S = 1, m(x_i)]$.

[43]In our stylized example from the previous footnote, the crime rate we impute to $x_i = 1$ defendants in the strict judge's caseload using the imputer trained on the lenient judge's caseload would equal α, below the crime rate we actually observe for defendants released by the stricter judge, $\alpha + \beta E[z_i|x_i = 1]$.

risk distribution - indeed not very different from plotting the predicted values calculated using the full training set against observed outcomes within the full training set (shown in the top panel of Figure 6). These findings are not consistent with the idea that the most lenient judges are unusually effective in using unobservables, and suggest instead that it is the algorithm itself that is dominating the choices of the stricter judges.

[Figure 6 about here.]

A different test we can construct takes advantage of the fact that there is *intra-judge* as well as inter-judge variation in leniency. That is, judges who hear cases across different borough, court house, year, month, day of week 'cells' often have different release rates across cells. If judges are using unobservables to help make release decisions then (conditional on observables) we should see that crime rates are relatively lower in cells where the judge has relatively higher release rates - that is, observed crime rates in these high-release cells should be lower than we would have predicted based on their observable characteristics alone. But this is not what we see.[44]

Judges do not seem to be simply setting a high bar for detention. They instead seem to be mis-ranking defendants when deciding whom to detain.

5 Translating predictions into policies

To scope the magnitudes of the potential gains to be had from using algorithmic predictions to inform release decisions, we simulate the results of specific policy changes. For example given our finding that judges mistakenly release predictably-risky defendants, one natural policy response could be a warning when such an error is about to happen. The resulting tool would be analogous to a 'driver assist' system that provides a warning whenever the car begins to do something the driver might not really wish to do, like drift into another lane. In the bail context, the warning might trigger whenever the judge is about to release a high-risk defendant, which would in effect *contract* the released set.

One drawback of driver assist systems is they are limited to warning of potential driver mistakes. They do not take advantage of the ability of the same technology to support an 'auto-pilot' that pro-actively initiates candidate maneuvers to avoid accidents. In the

[44] We implement this test with a simple regression. Let R^J be judge J's overall release rate among all the cases they hear, and let R^J_c be the judge's release rate within a given cell (c). Let y_{icJ} be the observed crime outcome for defendant (i) heard by judge (j) in cell (c), and let $m(x_{icJ})$ be the defendant's predicted crime rate as a function of observables. If judges are using unobservables effectively to make detention decisions then when we run the regression $y_{icJ} = \gamma_0 + \gamma_1 m(x_{icJ}) + \gamma_2[R^J_c - R^J] + \epsilon_{icJ}$ (clustering standard errors at the level of the judge-cell) we would expect to see $\gamma_2 < 0$. Yet our actual estimate for γ_2 within the test set is not distinguishable from zero (0.012 with a standard error of 0.012). By way of comparison, the estimate for γ_1 equals 0.970 (standard error of 0.012) in the test set.

judicial application, contraction limits us to jailing more people; we do not make use of the algorithm's ability to select people for release. A more dramatic policy change, then, would be a full *re-ranking* policy that would let the risk tool rank-order all defendants by predicted risk and make recommendations for all of the jail and release decisions. Such a re-ranking tool would be equivalent to an 'auto pilot' for bail decisions, which the judge could over-ride any time by grabbing the steering wheel.

Evaluating these policy changes requires producing counter-factuals, which raises two issues. First, we must know compliance rates: will the judge pay attention to the warning system and comply with the risk score? This cannot be known from the data we have. So we focus on the case of full compliance and scope the *potential* policy gains. The actual policy impacts cannot be known without experimental pilots. The impacts could be smaller if judges overrule either of the new algorithmic systems badly. The gains could be higher if the over-ride decisions of judges led to even better outcomes. Our policy scoping merely provides a base case.

The second problem arises in calculating crime rates. Once we make an assumption about judge compliance, this is a trivial exercise for the driver-assist-like contraction exercise since this policy would lead only to jailing some additional defendants, but not releasing any new ones. In this case we could evaluate impacts by calculating within the hold-out set the crimes committed by the new, contracted released set. But evaluating the re-ranking policy is more complicated, since it will release some defendants the judges jailed - resulting in a selective labels problem.[45] The contrast in the assumptions required for these two policy scoping exercises illustrates an important point: the degree to which the selective labels problem complicates our ability to evaluate policy counter-factuals depends on the specific policy being evaluated.

5.1 Contraction

As noted above, contraction is based on providing the judge with a 'warning' when a defendant is above a certain level of predicted risk. Thus, implementing contraction requires policy makers to decide the risk threshold that would activate the warning. A lower risk threshold averts more crime, but also triggers the warning for a larger share of

[45]The lack of labels for defendants who are detained by the judges can, if the judges have private information and use it effectively when making release decisions, bias our machine learning predictions - and so reduce prediction accuracy. But anything that reduces prediction accuracy for the machine learning algorithm simply has the effect of *understating* the gains from relying on machine rather than human predictions. In the same spirit another minor problem arises because machine learning algorithms are data hungry - prediction accuracy increases in the number of observations made available to the machine learner - so that anything that reduces the amount of observations available in the training dataset will reduce prediction accuracy.

people who would not have committed a crime had they been released. While the price society would be willing to pay is ultimately a normative choice, we can simulate the effects of any possible willingness to pay. In Figure 7 we show at each possible risk threshold the performance that such a system could achieve in terms of total crime reduced and extra people detained under the assumption that judges comply perfectly with all warnings.

[Figure 7 about here.]

Consistent with the previous finding that judges release very high risk individuals, we see a steep reduction in crime for small increases in jailing: for example, a 1 percentage point increase in detention reduces crime rates by nearly 5%. The choice of thresholds is, again, a normative matter. But we can get a potential clue from the implicit thresholds judges set as they increase leniency, as we saw in Figure 5. We know the predicted risk of the marginal defendants for each quintile. Of course these are the algorithm's predictions; we do not know what risk the judge "thought" they were releasing. If judges were unbiased in their predictions, the average predicted risk here would be a reasonable estimate. A more conservative estimate would assume some amount of mis-prediction on the part of the judge. Somewhat arbitrarily, we calculate the twenty-fifth percentile of predicted risk of the marginal defendants for each quintile. This is likely a lower bound on the risk threshold each quintile of judges had in mind. These lead to marginal risks of .397, .351, .338 and .307 for each of the leniency quintiles—shown as dashed lines in Figure 7.

We can use these dashed lines to simulate what would happen were this safeguard to be implemented at these thresholds. At the second quintile stringency threshold, we would increase jailing 2.9 percentage points more and reduce crime by 11.8 percent. For the other quintile thresholds these numbers involve a jailing increase of 4.4 percentage points and a crime decrease of -16.9%; jailing increase of 5.0 percentage points and crime decrease of -18.7%; and jailing increase of 6.7 percentage points and crime decrease of -23.4%. These results all reinforce that this policy could create steep crime reductions for modest jailing increases.

5.2 Re-ranking

Of course, policy makers would rightly be interested in tools that do not simply increase jailing. In re-ranking, we can set any jail rate. But evaluating this re-ranking requires a more ambitious policy simulation. While there is no perfect solution to the selective labels problem, one obvious initial possibility is to impute: assign the same crime outcomes to defendants the judges detained as those of observationally equivalent defendants whom

the judges actually released. This approach generates some obvious concerns, to which we return below.

To estimate the algorithm's release rule for this policy simulation, we divide our working sample into training (40%), imputation (40%), and test (20%) sets. We train one set of gradient-boosted trees on the training set, which yields predictions of each defendant's crime rate. We then rank-order all defendants in the test set by this predicted crime rate. For any release rate x, then, the algorithm simply releases the predictably riskiest fraction x of defendants. The actual crime rates of these defendants is then calculated.

[Figure 8 about here.]

Figure 8 graphs the crime rate (y-axis) that results at every possible target release rate (x-axis). Because we would like a crime rate that can be meaningfully compared across release rates, for purposes of Figure 8 we use the ratio of crimes committed by released defendants to the *total* number of defendants heard by the judge (not just the number released). If the algorithm chooses to release someone the judges released, we use the actual observed behavior of the defendant. If the algorithm chooses to release someone the judge detained, we use the imputed crime value.

Figure 8 suggests there could be sizable gains in outcomes under the algorithm's release rule relative to the judges. Under the current judge decisions, the judges release 73.6% of defendants and the crime rate (defined here as the ratio of crimes to all defendants) equals 11.3%. The curve showing the crime rate vs. release rate combinations achieved by the algorithm's release rule lies below this point. Because we do not know the utility function of the judges or society as a whole in weighing the gains from crime reduction versus reducing the jail population, we make a dominance argument. If we used the algorithm's release rule to match the current release rate of the judges, we would be able to reduce the crime rate by 24.7%. Alternatively, we could hold the crime rate constant and reduce the jail detention rate from 26.4% to 15.4%, a decline of 42.0%.

This policy simulation suggests the potential for large social impact, given that the US has well over 700,000 people in jail at any point in time. Such large gains are possible because at current release rates the risk of the marginal defendant is still relatively low, as shown in the bottom panel of Figure 8. With much larger reductions in the detention rate the risk of the marginal defendant begins to increase rapidly.

These potential gains are not just a matter of the algorithm beating a single judge who serves an out-sized caseload. Figure 9 shows the relative gains of the algorithm with respect to reducing crime (holding release rate constant) or reducing the jail population

(holding crime constant) for the 25 judges with the largest caseloads, rank-ordered by caseload size. We focus on this group so we have enough cases per judge to evaluate their individual performance; together they account for 47.2% of all cases in the test set. While there is some variation across judges, the algorithm dominates each judge.

[Figure 9 about here.]

The most important potential concern with this simulation is whether the gains are simply artifacts of imputation bias. Specifically, judges see factors we do not. If judges use this private information well, then those they release may not be comparable to those they do not, even conditional on observed x. Consider a very stylized example. Suppose that for young defendants, judges see gang tattoos and we do not, that judges know this type of tattoo is highly predictive of crime risk, and so they never release anyone with a tattoo. The imputer would attribute to all young people the crime rate of those without gang tattoos. This could seriously understate the increase in crime that would result from a risk tool that, for example, released all young people. However, two simple bounding exercises suggest this may not be a major problem in practice.

First, we find that the re-ranking gets much of its gain from there being many high-risk people to jail (a gain we can measure directly), not from ambitious assumptions about there being many low risk people to release. We illustrate this point in Figure 10.

[Figure 10 about here.]

In particular, notice re-ranking—when we are holding the release rate constant—can be understood as a set of swaps. We can begin with the released set of the judge and then take the riskiest released person and swap them for the least risky jailed person. As long as these two have different rates, we can keep swapping and improve performance. These swaps can be broken into two distinct steps: (i) jail a high risk defendant and release an *average* crime risk defendant; (ii) release a low risk defendant and jail an *average* crime risk defendant. When these steps are performed back-to-back they are the same as a swap but by decomposing them in this way we can decompose the gains from the two different activities. The top panel shows how each of these steps slowly moves us from the judge's crime rate to the algorithm's crime rate. In the bottom panel, the height of each bar in Figure 10 displays the gain from each step. The red bars are the jailing steps (which do not require imputation) and the blue bars are the release steps (which require imputation). It is clear from the bottom panel that most of the gains come from the jailing step. In fact, partway through, the algorithm starts releasing individuals that are

costly (higher than average risk) simply because there are still very large gains to be had in jailing high risk individuals.

[Table 5 about here.]

Table 5 presents a quantitative bounding exercise that examines how sensitive our policy scoping exercise is to our assumptions about the distribution of crime risk amongst the jailed. One thing to observe is that we are well calibrated within the lenient judge's data. So if at a particular risk level $\hat{y}$ we see the most lenient judge releasing at a rate $R_L(\hat{y})$, we can assume up to that fraction have average crime rate $\hat{y}$. For the remainder, we could imagine $\hat{y}$ is a poor imputation; for these we could assume that their true crime equals $min\{1, \alpha\hat{y}\}$. The last column of the table shows results for the most extreme possible assumption: the most lenient quintile of judges make perfect detention decisions (that is, $\alpha = \infty$ so that everyone the lenient judge detained would have committed a crime if released). We see that for a wide range of α this full re-ranking still produces large gains. Even at $\alpha = \infty$, the worst case, the drop in crime holding jail rate constant equals 58.3% of the gains we see in our main policy simulation, while the reduction in the jail rate, holding crime constant, equals 44.2% of the total gains reported above. Of course, the worst case is that all of these people are sure to commit a crime.

6 Judge mistakes

6.1 Are judges really making mistakes?

These policy simulations suggest large potential gains to be had if we use the algorithm's predictions to make release decisions. But could judges really be making such large prediction errors? Several factors could be confounding our analysis. In particular, perhaps judges have preferences or constraints that are different from those given to the algorithm. We explore several possibilities here.

6.1.1 Omitted payoff bias: Other outcomes

One potential concern is that when making release decisions, judges might have additional objectives beyond the outcome the algorithm is predicting. In this case it is possible the algorithm's release rule dominates the judges on *part* of what judges care about, but perhaps the judges dominate the algorithm on other inputs to their objective functions. We call this concern *omitted payoff bias*.

The most obvious version of this concern stems from the fact that New York state law asks judges to only consider flight risk when making pre-trial release decisions. Yet in practice judges might take other forms of defendant risk into consideration as well - such as risk of re-arrest. The degree to which this would bias our evaluation of the algorithm's performance depends partly on how highly correlated flight risk is with public safety risk.

[Table 6 about here.]

For starters we see in the top panel of Table 6 that those defendants who are at highest risk for FTA are at greatly elevated risk for every other crime outcome as well. The first row shows that the riskiest 1% of released defendants, in terms of predicted FTA risk, not only fail to appear at a rate of 56.4%, as already shown, but are also re-arrested at a 62.7% rate. They are also re-arrested for violent crimes specifically at a 6.1% rate, and re-arrested for the most serious possible violent crimes (murder, rape or robbery) at a 4.8% rate. The remaining rows show that identifying the riskiest 1% with respect to their risk of re-arrest (or re-arrest for violent or serious violent crimes in particular) leads to groups with greatly elevated rates for every other outcome as well.

The first row in the bottom panel of Table 6 reports the results of our policy simulation of potential gains in different forms of crime, relative to judge decisions, from re-ranking defendants for release by predicted FTA risk, holding overall release rates constant at the judge's current levels. An algorithm focused on FTA risk would, in addition to reducing FTA rates by 24.7%, *also* reduce overall re-arrest rates among released defendants by 11.1%, as well as re-arrest for violent crimes specifically by 15% and even re-arrest rates for the most serious violent crimes (murder, rape, robbery) by 13.3%.[46]

The next three rows of the table show that, were we to have trained an algorithm on those other outcomes, we could have achieved much bigger gains in them as well. For example holding release rate constant at the judge's level, an algorithm explicitly trained to predict risk of the most serious violent crimes could reduce the rate of such crimes by 57.3% (from 1.4% to 0.6%) while still also doing better than the judge on FTA rates.

6.1.2 Omitted payoffs bias: Prohibited predictors

Another additional objective judges might have beyond flight risk is racial equity. Even though we do not make race or ethnicity available to the machine learning algorithm, it

[46]One potential complication with looking at potential effects on re-arrests is the possibility of "replacement" - that is, if crime is committed in groups or if someone else would step in to fill an opportunity to commit a crime a jailed defendant would have committed; see for example the discussion in Donohue (2009). To the extent to which replacement occurs, our results will overstate the potential gains from reductions in other crimes, although this problem should not be relevant for our main results that focus on FTA. And to the extent to which replacement is an issue for other interventions that seek to reduce crime rates, we can still make meaningful comparisons between the gains from our bail tool versus other strategies.

is possible the algorithm winds up using these factors inadvertently - if other predictors are correlated with race or ethnicity.

This is perhaps the most important concern that has been raised by the possibility of using data-driven predictions to inform criminal justice decisions. As then-Attorney General Eric Holder noted in a 2014 speech, "we need to be sure the use of aggregate data analysis won't have unintended consequences," cautioning that these tools have the potential to "exacerbate unwarranted and unjust disparities that are already far too common in our criminal justice system and in our society."[47] To some this outcome is a foregone conclusion; for example law professor Bernard Harcourt has argued that "using risk-assessment tools is going to significantly aggravate the unacceptable racial disparities in our criminal justice system" (Harcourt, 2010, p. 2; see also Starr 2014).

[Table 7 about here.]

Table 7 compares the performance of current judge decisions to different versions of our algorithm. The first row shows the racial composition of the defendant pool as a whole. The second row shows the outcome of current judge decisions. At the current release rate, the crime rate (the ratio of crimes to total defendants)equals 11.3%, and 88.9% of inmates in jail are members of minority race or ethnic groups. The second row reproduces the results of our main algorithm, which does not have access to information about defendant race or ethnicity. As noted above, the resulting release rule lets us reduce crime by 24.7% at a given release rate, with a jail population that would turn out to have about the same share minority as we see under current judge decisions, equal to 90.1%. There is a slight increase by a few percentage points in the share of the jail population that is black, which is mostly offset by a small decline in the share Hispanic. But these are very small differences given the sampling variability in our test set.

The remaining rows of Table 7 show that it is possible to explicitly constrain the algorithm to ensure no increase in racial disparities in the jail with very little impact on the algorithm's performance in reducing crime. For example the third row shows what happens if we use our main algorithm's predicted risk to rank order defendants separately by their race and ethnic group (white, black and Hispanic). We then detain defendants in descending order of risk but stop detaining black and Hispanic defendants once we have hit the exact number of each group detained by the judges, to ensure that the minority composition of the jail population is no higher under the algorithm's release rule compared to the judges. Compared to the crime gains we achieved using the algorithm's

[47] http://www.justice.gov/opa/speech/attorney-general-eric-holder-speaks-national-association-criminal-defense-lawyers-57th

usual ranking-and-release rule, ensuring that we do no worse than matching the judges on racial disparities in the jail population leads to almost no loss in terms of the potential gain in reduced crime.

The second-to-last row shows what happens if we constrain the algorithm to be race neutral (that is, to ensure the share of each minority group in the jail population is no higher than their share within the general pool of defendants). The final row goes one step further and constrains the algorithm to ensure that the share of the jail population that is either black or Hispanic is no higher than either the share the judge jails or the share within the general defendant pool. In both cases we see that it is possible to reduce the share of the jail population that is minority - that is, reduce racial disparities within the current criminal justice system - while simultaneously reducing crime rates relative to the judge.

6.1.3 Omitted Payoff Bias: Suggestive Evidence from Philadelphia

Of course in principle there could be other outcomes judges care about as well beyond those we have considered here. For example, while the law asks judges to focus on crime risk, and to only consider factors such as employment or family circumstances as they may relate to the prediction of that risk, it is possible that some judges nonetheless take some consideration of these factors in making release decisions above and beyond their impact on risk. We cannot directly examine this in our own analysis because the data we have available to us does not measure employment or family status.

Nonetheless we can construct an indirect test for the possibility of other unmeasured judge objectives. If algorithmic prediction rules were failing on these other dimensions like job or family status, then one would expect that when presented with rules such as these, judges who were already optimizing would not make use of them. The perfect test of this hypothesis comes from the 1981 Philadelphia bail experiment (Goldkamp and Gottfredson, 1984, 1985a,b), which enrolled 16 judges and randomly assigned half to receive bail guidelines based on algorithmic predictions about likelihood of failure (like re-arrest). A total of 1,920 cases were handled by these judges. The algorithm was not derived from machine learning techniques, but nonetheless lets us test whether judges respond to these new risk predictions.

[Figure 11 about here.]

Figure 11 shows that receipt of the algorithmic predictions does indeed change judge behavior. The share of judge decisions that comply with the risk tool's recommendations

is much higher for the treatment than control group. This is inconsistent with the judges initially optimizing.[48]

6.2 Jail and caseload constraints

The previous sub-section examined the possibility that judges have different preferences from the outcome the algorithm is predicting. An alternative possibility is the algorithm only seems to yield improved outcomes relative to judges because the algorithm ignores some constraints that bind the judge's decisions.

For example, one particularly important constraint the judge may face is with respect to jail capacity. This could prevent the judge (but not the machine learner) from putting high-risk people in jail during times when the local jail is filled up.[49].

Table 8 shows that even after accounting for this concern, we continue to see large potential social-welfare gains from releasing defendants using machine rather than judge predictions. We re-train our algorithm in the NYC data, but now constraining the algorithm to have the same release rate as what we see over each three month window in New York City as a whole. The results are very similar to what we see in our main analysis in terms of either the average crime rate of the predictably-riskiest 1% that the algorithm can identify among the judge's released set, or in terms of the potential gains from the re-ranking policy simulation. The same finding holds if we constrain the algorithm to not exceed the release rate observed within each quarter-year, specific to the New York City borough in which each case is being heard.

[Table 8 about here.]

6.3 Algorithm (in)stability?

A different potential concern is that our analysis overstates the potential gains of the algorithm relative to the judges because the algorithm is unstable - that is, changing over time in ways that attenuate the potential gains of the algorithm relative to the judge

[48] More specifically, exposure to the algorithm causes judges to increase pre-trial release rates among the lowest-risk offenders (those arrested for the least serious offenses) and reduce release rates among the highest-risk defendants - exactly as we would predict if judges had initially mis-predicted failure risk among defendants (see, for example, Goldkamp and Gottfredson, 1984, Table 4.4; see also Abrams and Rohlfs, 2011).

[49] A simple stylized example helps illustrate the point. Suppose that on day 1 the local jail is at 100% capacity while on day 2 the jail is at 50% capacity. Suppose the local judge wishes to release 50% of defendants each day. In practice this judge would release 100% of defendants at their day 1 bond court because of the jail capacity constraints, including many high-risk defendants the judge wishes they could have detained if the jail had room. If the number of cases is the same day by day we would attribute to the judge an overall release rate of 75%, and so have the machine learner identify the lowest-risk 75% of cases each day and "release" them - even though in practice this would not have been possible on day 1 in our example

decisions. This could lead us to over-state the potential gains in the future from adopting an algorithmic release rule, since so far we have been comparing the relative performance of the algorithm to the judges averaged over our entire sample period.

Yet as shown in Table 8 there are no signs that our algorithm is particularly unstable. To check this, we re-train the algorithm using just data from the first few years of our sample period (2008 through 2012). Instead of evaluating the algorithm's performance on a randomly-selected set of cases drawn from the same time period over which the algorithm was trained, we now evaluate the algorithm using data from the first five months of data in 2013.[50] The gains from relying on the algorithm's predictions rather than current judge decisions are, if anything, even stronger when we predict out over time compared to what we see in our main analyses.

6.4 Understanding judge mis-prediction

The previous results suggests that judges are mis-predicting. We now attempt to understand *why* they are mis-predicting. This exercise can help shed light into what judges are getting wrong, and more generally highlights the potential of machine learning tools to help test theories of human decision-making and behavior, not just solve policy problems.

6.4.1 Which cases are hard?

For starters we can revisit where in the risk distribution judges are having the most trouble. While *ex ante* the answer is not obvious, looking at other domains can provide us with some initial intuition about what we might have expected. For example in education, studies find that principals do a good job identifying which teachers are in the tails of the performance distribution - the very high performers and the very low performers- but have a hard time distinguishing among teachers in the middle of the distribution (Jacob and Lefgren, 2008). As we will see, the results for judicial decisions in our data go against this intuition.

We can examine this question in our data by investigating where in the predicted-risk distribution judges have the most uncertainty. What we observe is just a binary indicator of whether the judges released a given defendant i, or $R(h(x_i, z_i)) = R_i$, which cannot convey much about when judges are most uncertain. However we can learn more about this by predicting the behavior of the judges. That is, we can train a new algorithm to

[50] As a reminder, we have taken out the data from May through November 2013 for our final hold-out set so the first part of 2013 are the latest observations available in our working dataset.

predict not the behavior of the defendants, but the behavior of the *judges*, to create a predicted version of the judge $\hat{J}(x_i)$.

[Figure 12 about here.]

In Figure 12 we sort all defendants in our test set into quintiles based on the algorithm's predicted risk of crime, $m(x_i)$, as shown along the x-axis. Then for each quintile of predicted defendant risk, we show the distribution of the predicted judge jailing probabilities, $\hat{J}(x_i)$. Additionally, we color each slice of the distribution by the actual jailing rate in that slice. We see from this that in each quintile of risk, those with highest predicted jailing rates are in fact jailed more: our predictions of the judge have some predictive power.

More importantly, we can also see in Figure 12 that cases in the lowest-risk quintiles have not only a low predicted jailing probability, but also have a quite compact distribution of predicted probability. These cases appear relatively easy - or at least judges behave consistently. In contrast, there is substantial dispersion in predicted jailing probabilities for the highest-risk cases; some high-risk defendants have high jailing probabilities while other cases with similarly high risk have low probabilities of jail—that is, they are treated as if they are low risk. Thus, in contrast to what we might have expected, judges in this case do not seem to struggle most with the middle and find the tails the easiest. Instead they seem to struggle most with one tail - the high risk cases.

6.4.2 Noisy predictions

Our prediction of the judge's release decisions can also give us some sense for why judges might be having this trouble. Recall we denoted judge predictions of defendant crime risk as $h(x_i, z_i)$, which is a function of both observables x_i (like criminal record) and factors z_i not observable to the algorithm (like the defendant's demeanor in the courtroom). The behavioral science literature suggests that this extra information could be part of the problem and may be the reason judges do poorly relative to the algorithm. For example, some studies suggest that highly salient interpersonal information (such as the degree of eye contact that is made) can be over-weighted; and that less salient but more informative data (like past behaviors) can be under-weighted.

If z is adding useful signal to judges in helping them predict defendant risk in our bail application, then $h(x_i, z_i)$ should be a better predictor than $\hat{h}(x) = E[h(x, z)|x]$. If z simply adds noise then the reverse should be true. While we do not observe $h(x_i, z_i)$, we do observe $R(h(x_i, z_i))$ as well as our prediction of the judge's release decision, $\hat{J}(x_i)$.

Notice that in making this prediction we never use data on crime. So if the predicted judge beats the judge, it must only be because the judge is misusing (on average) the unobserved z variables.

In essence, we would like to compare the judge to the predicted judge. We will do this in the same way we compared in Section 4.2 the judge to the algorithm. As before, we begin with the set of cases released by the most lenient quintile judges. We then jail additional defendants as we predict the judges would - jailing first those defendants with the highest predicted probability of judges jailing them. In other words, we begin by jailing those defendants who are released but whom we predicted have the highest probability of being jailed by the judges. (The difference with our previous comparison to the judges is that we had earlier jailed defendants by the algorithm's predicted crime risk.)

[Figure 13 about here.]

Figure 13 plots the results of this exercise. We plot performance of judges in each leniency quintile. We then show what crime rate that would result if we jailed as the predicted judge would. We also include for comparison here what would happen if we jailed according to predicted crime. We see here clearly that the predicted judge does better than judges themselves. In fact, simply jailing according to $\hat{J}$ gets us roughly halfway towards the gain we had from jailing according to predicted risk.

[Table 9 about here.]

As before, each quintile of judges is compared to two points on this curve and the results are in Table 9. We quantify the crime reduction that would have resulted if the algorithm had the same increase in jailing; and we measure what jailing increase leads to the same crime reduction. The predicted judge does significantly better. The second quintile of judges reduce crime by 9.9% by increasing the detention rate by 6.6 percentage points. The predicted judge would have achieved the same crime reduction by increasing the detention rate by only 4.7 percentage points (28.8% less than the judge); or alternatively, by increasing the detention rate by 6.6 percentage points we could have reduced crime by 13.4% (35.5% more than the judge).[51]

These results, though, could also be due to a 'wisdom of the crowd' effect: $\hat{J}$ is not the predicted version of a single judge, but rather the prediction of many judges. To

[51] We have also done the re-ranking policy simulation using $\hat{J}$ instead of predicted risk as before. This produces similar results to what we report here: the predicted judge does better than the judge but not as well as the algorithm designed to predict risk.

more fairly compare judges to predicted versions of themselves, we built a new set of individuated algorithms each of which is trained on a single judge's caseload.[52] We can then use any one of these individual predicted judges to contract down the released set of the most lenient-quintile judges' caseload as in Figure 5. In Figure 14, we show how each of these individual $\hat{J}$ outperforms the decisions of actual judges.

[Figure 14 about here.]

Our results taken together suggests one reason why judge release decisions can be improved upon: Their actual decisions are noisy relative to a mean ($\hat{J}$) that contains much more signal. In particular, this "noise" appears to be due to unobservable variables, which unduly influence these decisions.

6.4.3 Release vs. bail amount

Finally, there is a question of *what* exactly judges are mis-predicting. So far in our analysis we have made the simplifying assumption that judges release or jail, when in fact they set a bail amount as well. It is logically possible that in the results we presented above, the judge actually intended to jail high-risk people but simply mis-predicted what bail amount they would be able to make and assigned them bail amounts that were 'too low.' Put differently, it is possible that judges are actually mis-predicting 'ability to pay' rather than risk.

To examine this possibility we can examine the degree of predictable risk we find among defendants the judges release outright - that is, people the judges assign to release on recognizance (ROR). For this group there is no possibility of mis-predicting ability to pay, since there is no bail amount that is required for release; people are simply released without any requirement of cash or other collateral. We see that even among the set of people the judges released outright (ROR'd), there are people with high rates of predictable risk: the crime rate for the riskiest 1% of defendants released outright by the judge is similar to what we see among the full released set ($\bar{y} = 59.2\%$ versus $\bar{y} = 56.3\%$).[53]

[52] We restrict ourselves to judges that heard at least 5,000 cases in our study sample, to ensure that we have enough data to construct a meaningful algorithm.

[53] A different potential problem this poses is similar to the selective labels problem, coming this time from the fact that the bail amount also can affect a person's propensity to commit crime. The bail amount is held as collateral and so would be part of the penalty of being caught for a crime. For the released, you would expect a higher bail amount should lead to less crime. Specifically, when we release someone the judge has jailed, we impute to them the average crime rate of similar people the judge released. But the judge did not "release" those people: she set a bail amount for each of them. By imputing this crime rate, we are assuming the algorithm can discern a bail amount that produces the same crime rate as seen in the population released by the judge. This is feasible for instances in which the algorithm jails people the judge would have released, since the algorithm can in principle choose an arbitrarily high bail amount to ensure that the defendant is jailed. And so this problem of predicting risk versus ability to make bail should not be an issue when we go into the caseload of the most lenient quintile of judges and use the algorithm to select marginal defendants to detain. We see much larger declines

7 National replication

Our analysis so far of data from New York City suggests that judges seem to mis-predict and fail to release defendants by predicted risk, and that there could be sizable gains to be had from changing the way we make pre-trial release decisions. While NYC is a particularly important city, it is still just one jurisdiction. Perhaps the potential gains we report above from using machine learning predictions are due to something idiosyncratic about our dataset, or idiosyncratic to New York's criminal justice system. Could they be replicated in other jurisdictions?

To answer this question we replicate our analysis using a national dataset assembled by the US Department of Justice (DOJ) that captures data on felony defendants from 40 large urban counties over the period 1990-2009.[54] Unlike New York, most of these jurisdictions ask judges to focus on public safety as well as flight risk, rather than just flight risk, as part of pre-trial bond court hearings. So we train our algorithm on an outcome defined as committing either failure to appear or re-arrest. The DOJ dataset contains a total of 151,461 observations.

Descriptive statistics for this dataset are in Appendix Table A2.[55] The DOJ dataset unfortunately does not include judge identifiers. The release rate here, 60.6%, is lower than what we see in New York City, partly because New York City tends to have higher release rates than other cities, and partly because the DOJ dataset is limited to felony defendants. The share of released defendants who commit any crime (either FTA or re-arrest) is 30.6%, with rates for specific subcategories shown in the Table.

[Table 10 about here.]

The top panel of Table 10 shows that just as in the New York City data, judges in this nationally representative sample of urban counties are also releasing defendants with very

in crime from selecting marginal defendants using the algorithm compared to the choices of the stricter judges, gains that cannot be an artifact of judicial mis-prediction of ability to make bail.

[54]The dataset was collected as part of the state court processing statistics series, formerly called the National Pretrial Reporting Program (see USDOJ, 1990-2009). DOJ identified the 75 most populous counties in the US, sub-sampled 40 of them, and then collected data intended to be representative of all felony cases that were filed in these counties during a selected set of days during May of each study year, which included selected years between 1990 and 2009. The jurisdictions from which data were collected as part of this dataset are as follows (state followed by counties): Arizona (Maricopa, Pima); California (Los Angeles, Orange, San Bernardino, Ventura); Connecticut (Hartford); Florida (Broward, Miami-Dade, Hillsborough; Orange); Hawaii (Honolulu); Illinois (Cook); Indiana (Marion); Maryland (Baltimore, Montgomery, Prince George); Michigan (Oakland, Wayne); Missouri (Saint Louis); New Jersey (Essex, Middlesex); New York (Bronx, Kings, Nassau, New York, Suffolk); North Carolina (Wake); Ohio (Cuyahoga, Franklin, Hamilton); Tennessee (Shelby); Texas (Dallas, El Paso, Harris, Tarrant); Utah (Salt Lake City); Washington (King); and Wisconsin (Milwaukee).

[55]The dataset includes sampling weights that reflect two factors. First, they account for the higher probability of selection for large jurisdictions, with first-stage weight values that range from 1.1 for the largest counties to 2.5 for the smallest counties. In addition weights also account for the fact that the dataset captured felony cases over just 1 week for the largest counties and for a full month for the smallest counties, so the second-stage weights range from 4 down to 1. Exhibits in the main paper are presented without weighting. The appendix shows that the weighted results are fairly similar to those without weights.

high levels of predictable risk. Using just data available to the judges at the time of the pre-trial hearing, those in the top 1% of the predicted risk distribution have an observed crime rate of 76.6%, and yet are released by judges at a rate of 52.5%. The release rate for this high-risk set is not that much lower than the overall release rate of 61.6%, although the observed crime rate is much higher than the sample average.[56]

The bottom panel of 10 shows the results of carrying out our policy simulation for the potential gains from letting the algorithm rank-order all defendants and then forming a simple rule that detains in descending order of predicted risk.[57] Compared to the judges, a release rule based on machine learning predictions would let us reduce the crime rate by 18.8% holding the release rate constant, or, holding the crime rate constant, we could reduce the jail rate by 24.5%.[58]

The gains of the algorithm over the judges are at least as large in New York City as in the national DOJ data; and this is interesting given that in New York - unlike most other jurisdictions - judges already have access to the recommendations of some sort of risk tool to make their pre-trial release decisions.[59] One reason the existing tool, a weighted average of responses to six questions about the defendant and their current case, may not be a 'game-changer' in New York City is that it seems to have only modest predictive accuracy. Perhaps for that reason, judges in New York seem to put only modest weight on the tool's recommendations when making pre-trial release decisions.[60]

Finally in the national DOJ data we again find that the release decisions of the 'predicted judge' (the part of the judge's decisions projected down onto the set of observable case characteristics) leads to better outcomes compared to the actual decisions of the judges themselves. Judges nationwide, not just in New York, seem to be mis-weighting

[56]Given the DOJ dataset's panel structure, we identify the top 1% of the predicted risk distribution within each county-year cell, and then report the average observed crime rate among that group.

[57]As we did with our NYC data, we randomly partition the dataset into a training, imputation and test set and fit boosted decision trees in the training and imputation sets to rank-order defendants for release and help estimate outcomes under our algorithm's counter-factual decision rule. As we did with the policy simulation with the NYC data, for this exercise we define 'crime rate' here as the number of crimes (FTA or re-arrests) divided by the total number of defendants who pass through bond court, so that we can meaningfully compare crime rates at different potential release rates.

[58]These gains come from constraining the algorithm to have the same release rate within each county-year cell as judges, to avoid giving the algorithm an unfair advantage in being able to make release decisions the local judges could never make - for example releasing all defendants in low-crime places or time periods, and detaining all defendants in high-crime places or times.

[59]Comparing the size of the gains across our two datasets is complicated by the fact that defendant characteristics are different and sample sizes of the two datasets are different.

[60]For example, for felony arrests in New York, the recent citywide FTA rate is 11%, while the FTA rate among cases that the 6-question tool developed by CJA recommends for ROR is 7% and for cases the CJA tool has "not recommended for ROR" equals 16% (see New York City Criminal Justice Agency, Inc., 2016, p. 33-4.) One common accuracy statistic in machine learning is known as "lift," the ratio of the prevalence of an outcome in some identified sub-group relative to the population prevalence. So the lift of the CJA tool's "high risk" felony defendant category is 16%/11%=1.45. For non-felony cases the lift is 22%/14%=1.57. By way of comparison, we noted before that the top 1% of the predicted risk distribution as identified by our own algorithm in the NYC data has a FTA rate of 56%; since the FTA base rate in our sample is 15%, the implied lift is 56%/15%=3.73. CJA's analysis of the degree to which judges follow the recommendations of the tool suggests that their tool's recommendations explain 18% of the variation in case outcomes in NYC arraignments.

qualitative factors that are not in the administrative data available to the algorithm. In sum, for each of our key findings from the NYC data that we can estimate using the DOJ data, the result seems to replicate in this national dataset.

8 Conclusion

Reducing jail and prison populations without increasing crime is a key policy priority. Empirical economics research in this area typically focuses on causal questions, such as identifying prevention and rehabilitation interventions. Our analysis here takes a very different approach: improved prediction. More accurately identifying high risk defendants in the New York data reduces crime rates by 25%, holding release rates constant, or reduces pre-trial jailing rates by over 40% with no increase in crime. We see qualitatively similar gains in a national dataset as well.

Working through the bail example also highlights how empirical economics work on prediction policy problems enriches the machine learning technology as well. The potential for policy impact comes from solving the problem:

$$\text{Data} \rightarrow \text{Prediction} \rightarrow \text{Decision}$$

Applied work in machine learning typically focuses on the Data $\rightarrow$ Prediction link. The objective is to search through different candidate prediction functions and identify the one with the greatest prediction accuracy - a 'bake off.' Algorithm performance tends to be quantified on predictive value, rather than decision value.

The bail example, though, illustrates why understanding the Prediction $\rightarrow$ Decision link is at least as important. Being clear about how predictions translate to decisions can substantially influence how the prediction function is and evaluated. In our application, for example, it is not enough to evaluate prediction quality. We must evaluate decision quality and some of the counter-factuals needed to estimate this immediately raise the selective labels problem. The focus on decisions also makes clear the need to fully specify judicial preferences and how too narrow a focus can induce omitted payoff biases.

While a key contribution of our paper is to highlight the importance of these econometric concerns in machine learning applications, and how to address them, even in our analysis there remain some open questions. For example, we do not have data on employment and family status of defendants and judges' release decisions may take these factors into account (despite the law asking them to focus solely on crime risk). The Philadelphia data suggest that this omission alone is unlikely to drive our findings. Still,

it would be useful to collect more data on this issue: more detailed experimentation with decision-aids would help unravel judicial preferences.

Our application also highlights that solving prediction policy problems can also help to understand human decision making. We saw, for example, the role of unobservable variables in adding noise to judges' decisions. We see that judges effectively under-weight key observable variables like prior criminal record.

Prediction policy problems represent a scientifically interesting and socially important class of problems to be addressed. These results suggest that the impact of applying empirical predictions could be large, consistent with a large behavioral literature that suggests people have biases in making probability estimates and predictions. But progress on these problems will require a synthesis of multiple perspectives—both the techniques of machine learning as well as a focus on issues that will seem very familiar to economists: how preferences and constraints shape decisions.

References

Abrams, David S. and Chris Rohlfs (2011) "Optimal bail and the value of freedom: Evidence from the Philadelphia bail experiment." *Economic Inquiry*. 49(3): 750-770.

Aizer, Anna and Joseph Doyle, Jr. (2015) "Juvenile incarceration, human capital and future crime: Evidence from randomly-assigned judges." *Quarterly Journal of Economics*. 130(2): 759-803.

Athey, Susan, and Guido Imbens (2016). "Recursive partitioning for heterogeneous causal effects." *Proceedings of the National Academy of Sciences*. 113.27: 7353-7360.

Banko, Michele, and Eric Brill (2001). "Scaling to very very large corpora for natural language disambiguation." *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2001.

Belloni, A., Chernozhukov, V. and Hansen, C. (2014). "High-dimensional methods and inference on structural and treatment effects." *The Journal of Economic Perspectives*. 28(2): 29-50.

Berk, Richard (2012) *Criminal Justice Forecasts of Risk: A Machine Learning Approach* Springer.

Berk, Richard, and Justin Bleich (2013) "Forecasts of violence to inform sentencing decisions." *Journal of Quantitative Criminology*. 30(1): 9-96.

Berk, Richard, Justin Bleich, Adam Kapelner, Jaime Henderson, Geoffrey Barnes, and Ellen Kurtz (2014) "Using regression kernels to forecast a failure to appear in court." ArXiv Preprint 1409, no. 1798 (20tp://arxiv.org/pdf/1409.1798.pdf

Bhuller, Manudeep, Gordon Dahl, Katrine V. Loken, and Magne Mogstad (2016) "Incarceration, recidivism and employment." Cambridge, MA: National Bureau of Economic Research Working Paper 22648.

Black, Fischer (1986) "Noise." *The Journal of Finance*. 41.3 (1986): 528-543.

Bordalo, Pedro, Nicola Gennaioli, and Andrei Shleifer (2012). "Salience theory of choice under risk." *The Quarterly Journal of Economics*. 127 (3): 1243-1285.

Bushway, Shawn and Jeffrey Smith (2007) 'Sentencing using statistical treatment rules: What we don't know can hurt us.' *Journal of Quantitative Criminology*. 23: 377-87.

Dawes, Robyn M., David Faust, and Paul E. Meehl (1989) "Clinical versus actuarial judgment." *Science*. 243(4899): 1668-74.

Dawes, Robyn M. (1971) "A case study of graduate admissions: Application of three principles of human decision making," *American Psychologist*. 26: 180-88.

Di Tella, Rafael, and Ernesto Schargrodsky (2013) "Criminal recidivism after prison and electronic monitoring." *Journal of Political Economy*. 121.1: 28-73.

Dietvorst, Berkeley J., Joseph P. Simmons, and Cade Massey (2015) "Algorithm aversion: People erroneously avoid algorithms after seeing them err." Journal of Experimental Psychology: General 144.1: 114.

Dobbie, Will, Jacob Goldin and Crystal Yang (2016) "The effects of pre-trial detention on conviction, future crime, and employment: Evidence from randomly assigned judges." Cambridge, MA: National Bureau of Economic Research Working Paper 22511.

Donohue, John J. (2009) "Assessing the relative benefits of incarceration: Overall changes and the benefits on the margin." In *Do Prisons Make Us Safer? Benefits and Costs of the Prison Boom*, Steve Raphael and Michael A. Stoll (Eds). New York: Russell Sage. pp. 269-342.

Federal Bureau of Investigation (2016) *Crime in the United States, 2015.* Washington, DC: Criminal Justice Information Services Division, Federal Bureau of Investigation, US Department of Justice. https://ucr.fbi.gov/crime-in-the-u.s/2015/crime-in-the-u.s.-2015

Friedman, Jerome H. (2001) "Greedy function approximation: A gradient boosting machine." *Annals of statistics*. 29(5): 1189-1232.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. Springer, Berlin: Springer series in statistics.

Goldkamp, John S. and Michael R. Gottfredson (1984) *Judicial Decision Guidelines for Bail: The Philadelphia Experiment*. Washington, DC: US Department of Justice, National Institute of Justice

Research Report NCJ95109.

Goldkamp, John S. and Michael R. Gottfredson (1985a) *Judicial Decision Guidelines for Bail: The Philadelphia Experiment, 1981-82. ICPSR08358-v1.* Ann Arbor, MI: Inter-University Consortium for Political and Social Research [distributor] http:doi.org/10.3886/ICPSR08358.v1

Goldkamp, John S. and Michael R. Gottfredson (1985b) *Policy Guidelines for Bail: An experiment in court reform.* Philadelphia, PA: Temple University Press.

Gupta, Arpit, Christopher Hansman, and Ethan Frenchman (2016) "The heavy costs of high bail: Evidence from judge randomization." Columbia University Working Paper.

Harcourt, Bernard E. (2010) "Risk as a proxy for race." University of Chicago Law School, John M. Olin Law and Economics Working Paper Number 535.

Jacob, Brian A. and Lefgren, Lars (2008). "Principals as agents: Subjective performance assessment in education." *Journal of Labor Economics.* 26(1): 101-136

Kleinberg, Jon, Jens Ludwig, Sendhil Mullainathan and Ziad Obermeyer (2015) "Policy prediction problems." *American Economic Review, Papers and Proceedings.* 105(5): 491-5.

Kling, Jeffrey R. (2006) "Incarceration length, employment, and earnings." *American Economic Review.* 96(3): 863-76.

Leslie, Emily and Nolan Pope (2016) "The unintended impact of pretrial detention on case outcomes: Evidence from NYC arraignments." University of Chicago Working Paper.

Mendel, Brock, and Andrei Shleifer (2012) "Chasing noise." *Journal of Financial Economics.* 104.2 (2012): 303-320.

Mueller-Smith, Michael (2015) "The criminal and labor market impacts of incarceration." Working paper, University of Michigan.

Murphy, Kevin P. (2012) *Machine Learning: A Probabilistic Perspective.* Cambridge, MA: MIT Press.

New York City Criminal Justice Agency, Inc. (2016) *Annual Report 2014.* New York, NY: New York City Criminal Justice Agency Inc.

Phillips, Mary T. (2012) *A Decade of Bail Research in New York City.* New York, NY: New York City Criminal Justice Agency.

Shiller, Robert J (1981) "Do stock prices move too much to be justified by subsequent changes in dividends?" *American Economic Review* 71.3: 421-436.

Starr, Sonja B. (2014) "Evidence-based sentencing and the scientific rationalization of discrimination." *Stanford Law Review.* 66: 803-872.

Stevenson, Megan (2016) "Distortion of justice: How the inability to pay bail affects case outcomes." Working paper, University of Pennsylvania.

Tan, Chenhao, Lillian Lee, and Bo Pang (2014) "The effect of wording on message propagation: Topic- and author-controlled natural experiments on Twitter," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014).*

US Department of Justice, Bureau of Justice Statistics. *State Court Processing Statistics, 1990-2009: Felony Defendants in Large Urban Counties [computer file].* Conducted by Pretrial Justice Institute (formerly, the Pretrial Services Resource Center). ICPSR02038-v5. Ann Arbor, MI: Inter-university Consortium for Political and Social Research.

Yeomans, Mike, Anuj Shah, Jon Kleinberg, and Sendhil Mullainathan (2016). "Making Sense of Recommendations," Working paper, Harvard University.

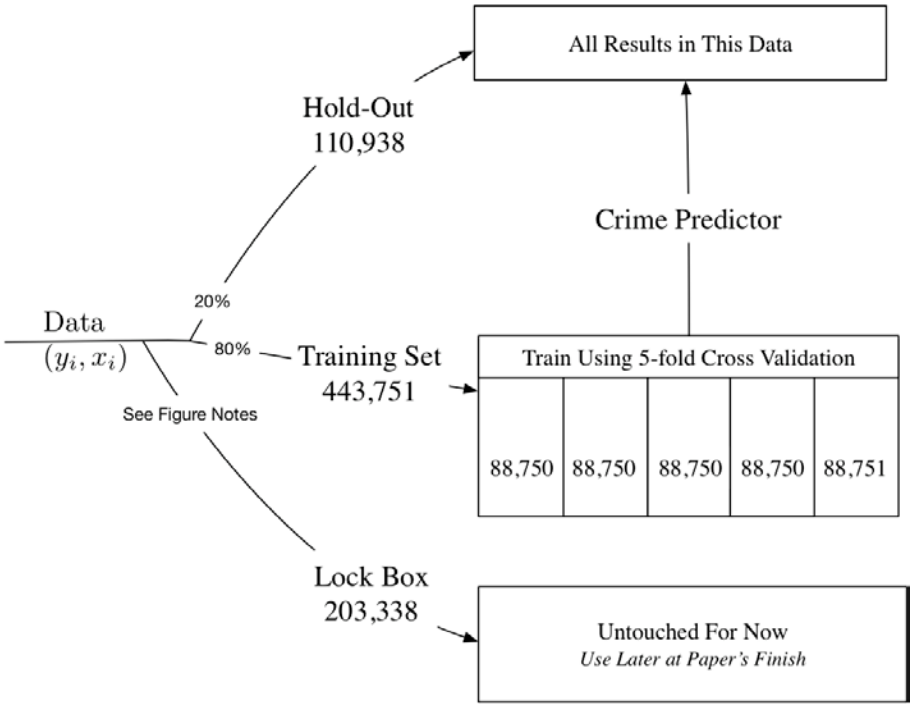

Figure 1: Schematic for Data Flow and use

Notes: We show here the partitioning and analysis strategy for our dataset from New York City covering arrests from November 1, 2008 through November 1, 2013. The original sample size is 1,460,462. For our analysis we drop cases that were not subject to a pre-trial release hearing, which leaves us with a total of 758, 027 observations. We selected the final hold-out set of 203,338 by taking all cases arraigned in the last 6 months of our dataset (all cases arraigned after May 1, 2013), randomly selecting all cases heard by judges among the 25 judges with the largest caseloads until reaching 10% of total observations, which winds up selecting 7 judges, and randomly selecting 10% of all observations (these samples can be overlapping). To prevent human data-mining, we will only use this hold-out set when the paper is set for publication. In this draft we evaluate all of our results by randomly selecting a test set of 20% of the remaining 556,842 observations in our working sample, and using the rest of the data to train the various algorithms we build in our analysis.

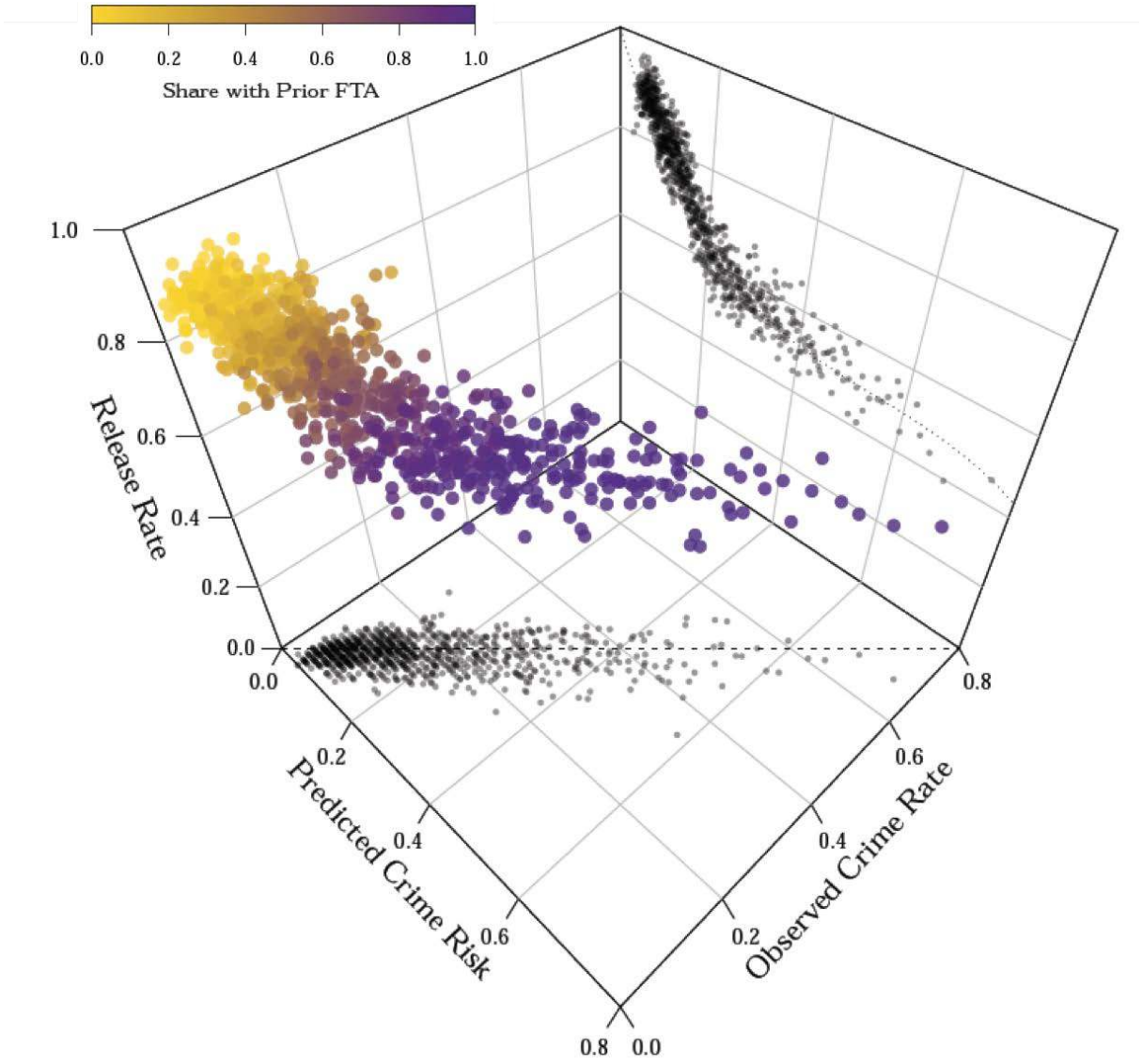

Figure 2: Outcomes for the Relased Ranked by Predicted Risk

Notes: The figure above shows the results of an algorithm built using 221,876 observations in our NYC training set, applied to the 110,938 observations in our test set (see Figure 1). We report the observed judge's release rate (y-axis) against both the algorithm's predicted crime risk for each observation and the observed crime rate (observed only for those defendants in the test set released by the judges) for 1,000 bins sorted by predicted risk. The coloring shows share observations in each bin with a prior failure to appear. The bottom and back panels show the projection of the figure onto the two dimensional {predicted crime risk, observed crime rate} space and the {predicted crime risk,judge release rate} space.

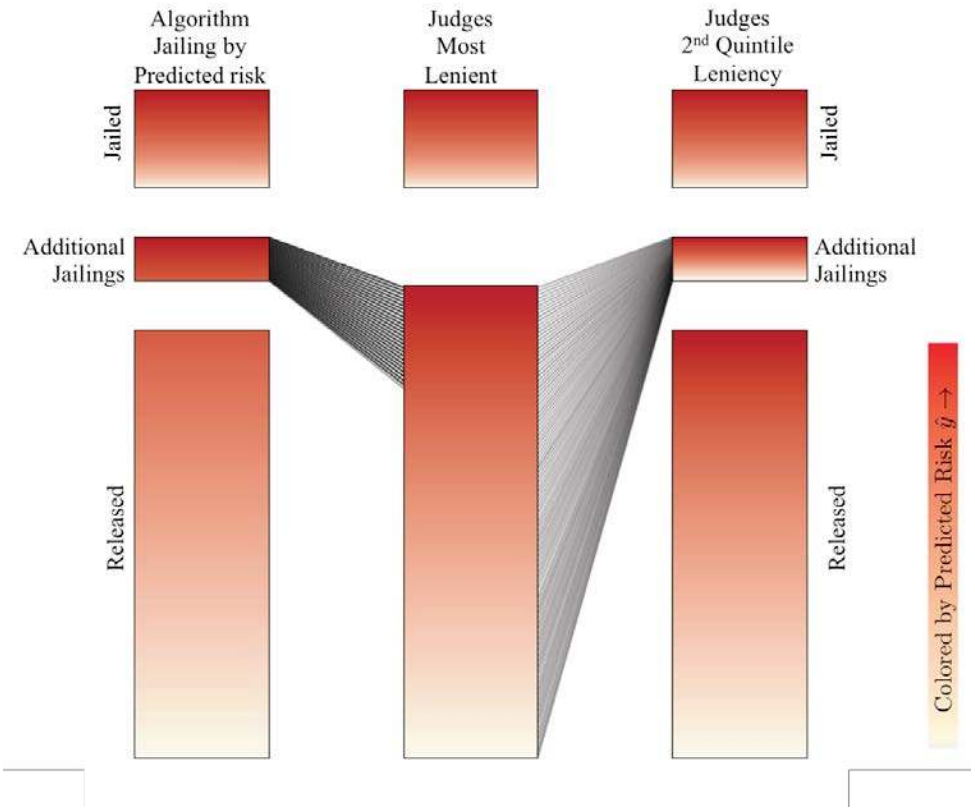

Figure 3: Who is Jailed as Judges Become More Stringent?

Notes: The bar in the middle of the figure shows who the most lenient quintile judges jail (top) and release (bottom) in our NYC dataset. The color shading shows the algorithm's predicted crime risk. The bar at the right shows how the 2nd most lenient quintile judges implicitly select their marginal detainees to get from the most lenient quintile's release rate down to their own release rate. The arrows show where within the risk distribution of the lenient quintile's released set the stricter judges are selecting the marginal detainees. The bar at the left shows how the algorithm would select marginal detainees to achieve the same reduction in release rate.

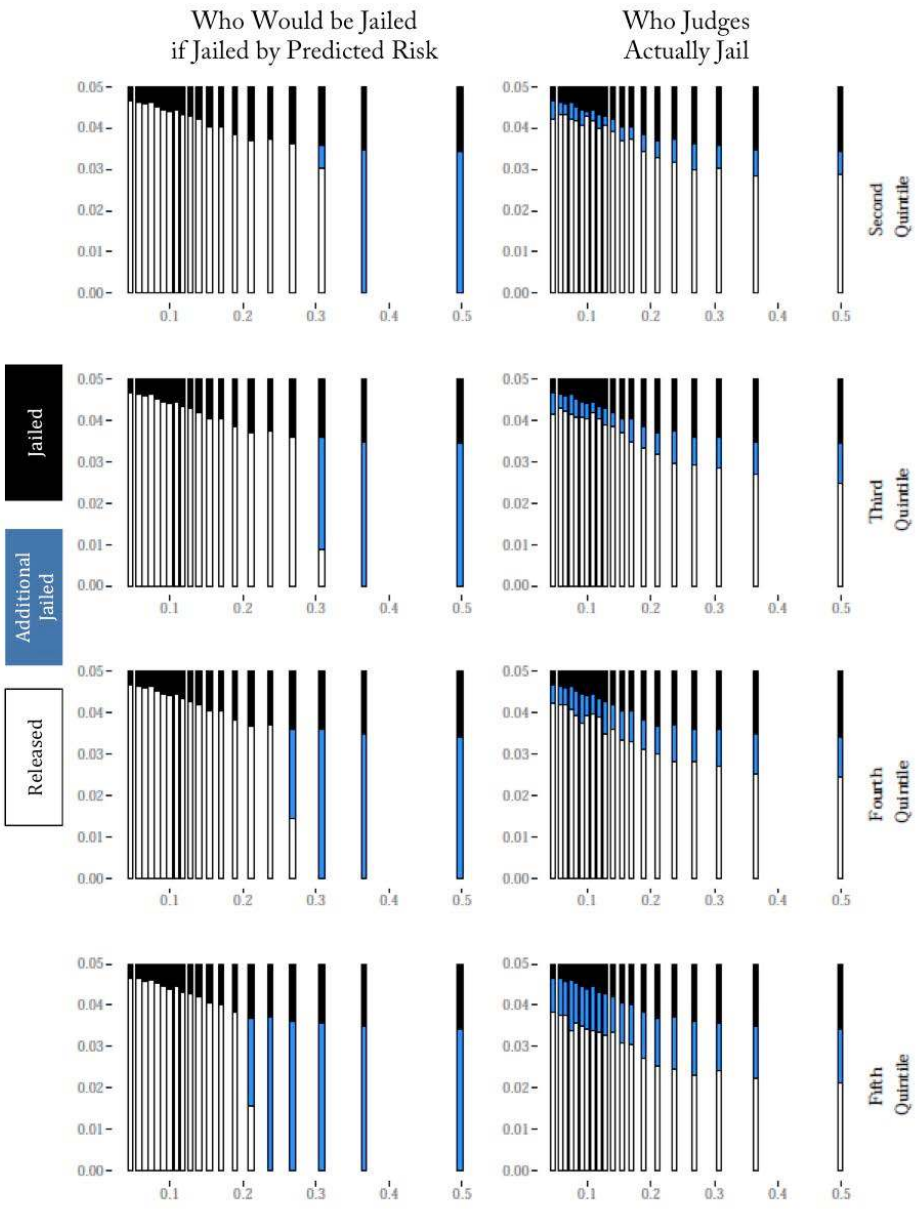

Figure 4: Who do Stricter Judges Jail? Predicted Risk of Marginal Defendants

Notes: This figure shows where each of the quintiles of stricter judges in NYC select their marginal defendants (relative to the most lenient quintile), compared to how the algorithm would select marginal detainees. Within each panel, we divide the sample up into 20 bins by predicted crime risk (shown on the x-axis). The black segment at the top of each bar shows the share of each bin the most lenient quintile judges jail. In the top right-hand panel, we show which defendants the second-most-lenient quintile judges implicitly select to jail to get from the most lenient judge's release rate down to their own lower release rate (blue), and who they continue to release (white). The left-hand top panel shows whom the algorithm would select instead. Each of the remaining rows shows the same comparison between the judge and algorithm decisions for the other less-lenient judge quintiles.

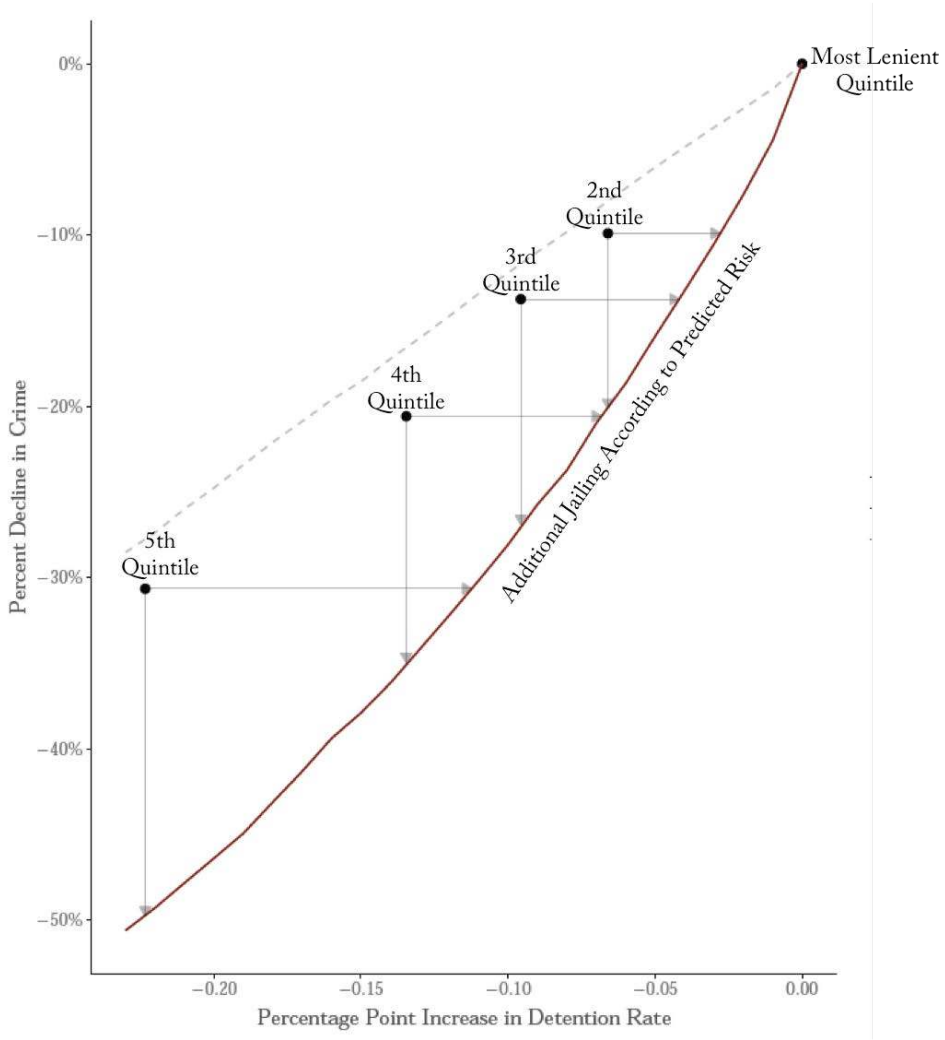

Figure 5: Comparing Impact of Detaining in order of Predicted Risk To What Judges Achieve

Notes: This figure looks at performance when additional defendants are jailed according to a predictive model of who judges would jail. It compares crime rates and release rates to the decisions of stricter judges. The right-most point in the graph represents the release rate of the most lenient quintile of judges, with the crime rate that results. The red line shows the crime reductions that realize if we jail additional defendants from this pool according to predicted behavior of judges. By comparison, the light dashed line shows the decline in crime (as a percentage of the lenient quintile's crime rate, shown on the y-axis) that results from randomly selecting additional defendants to detain from within the lenient quintile's released cases, with the change in release rate relative to the lenient quintile shown on the x-axis. As another comparison, the green curve shows the crime rate / release rate tradeoff that comes from jailing additional defendants within the lenient quintile's released set in descending order of the algorithm's predicted crime risk. The four points on the graph show the crime rate / release rate outcomes that are observed for the actual decisions made by the second through fifth most lenient quintile judges, who see similar caseloads on average to those of the most lenient quintile judges.

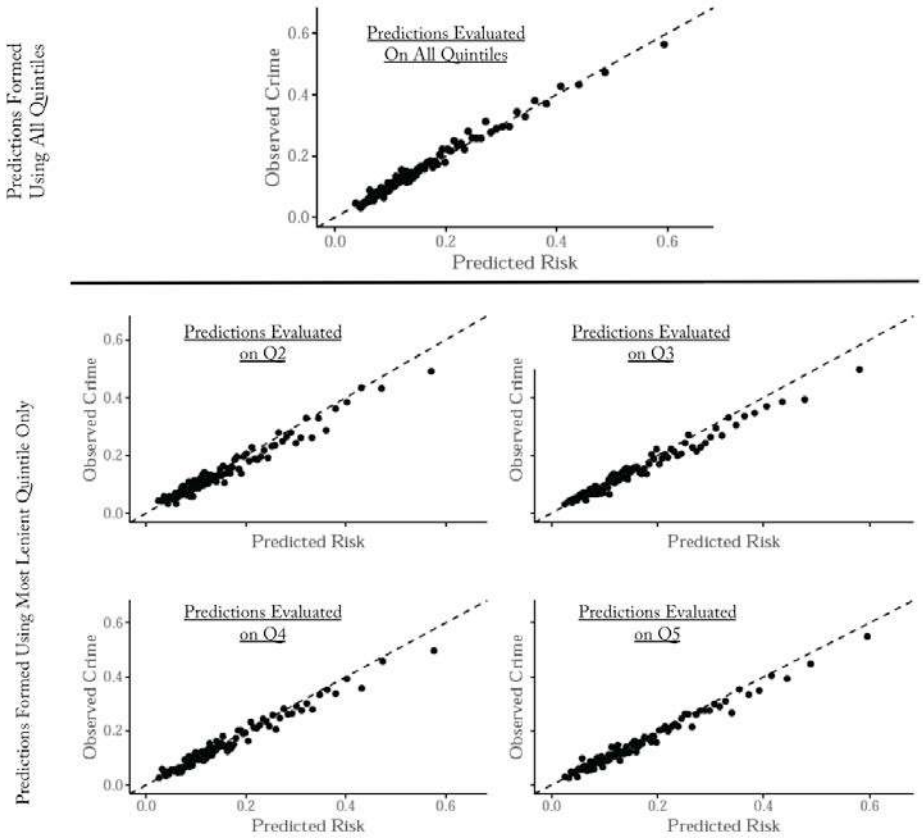

Figure 6: Predicting in Less Lenient Quintiles Using Data from Most Lenient

Notes: This figure tests for whether the most lenient quintile judges in our NYC dataset are better at using 'unobservables' in making release / detain decisions than are the less-lenient quintile judges. The top panel reproduces the calibration curve from Figure 2, plotting the algorithm's predicted crime risk against observed crime rates within the test set. For the remaining panels, we train an algorithm using just the set of defendants released by the most lenient quintile judges, and then use that algorithm to generate predicted crime risk to compare to observed crime rates for the set of defendants released by the less-lenient quintiles of judges.

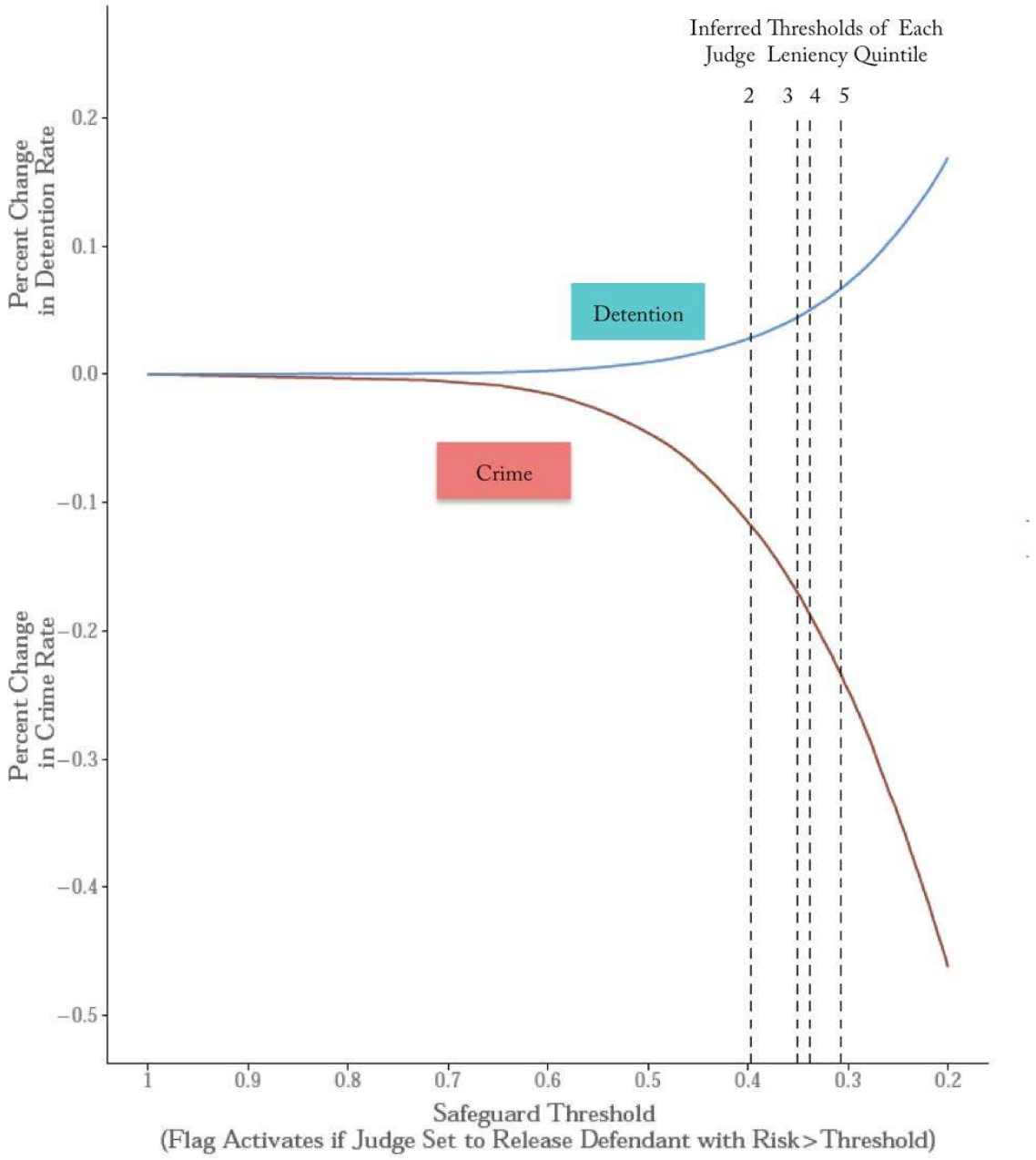

Figure 7: Simulating Effects of a Warning System

Notes: The figure above simulates the change in outcomes that would result in NYC from applying an early warning system that would warn judges when they are about to release someone with predicted crime risk above some threshold, assuming perfect judge compliance. We plot different candidate thresholds plotted along the x-axis. The top panel shows the percentage point change in the detention rate that would result, while the bottom panel shows the percent change in crime rate that would result.

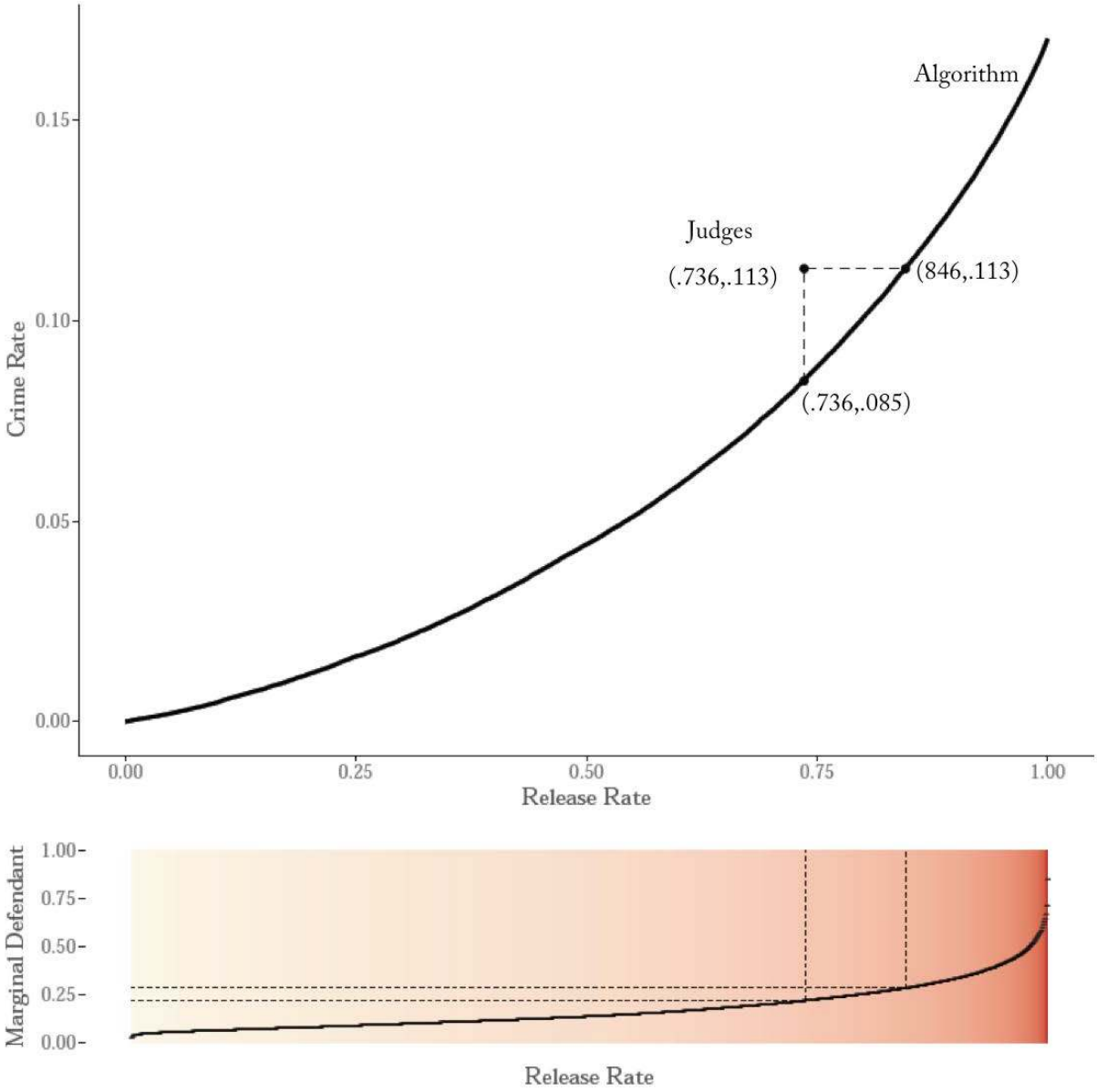

Figure 8: Simulation of Crime Rate - Release Tradeoff Algorithm Allows

Notes: The curve in the top panel shows the crime rate and release rate combinations that would be possible in NYC if judges were given a risk tool that could re-rank all defendants by their predicted crime risk and recommend them for detention in order of risk. Since we would like a crime rate that can be meaningfully compared across release rates, the y-axis shows the ratio of crimes committed by released defendants to the *total* number of defendants, not just the number released.The curve shows what gains would be possible relative to actual current judge decisions, assuming perfect compliance with the new tool. The curve in the bottom panel shows the risk level of the marginal person detained at each possible release rate under the algorithmic release rule.

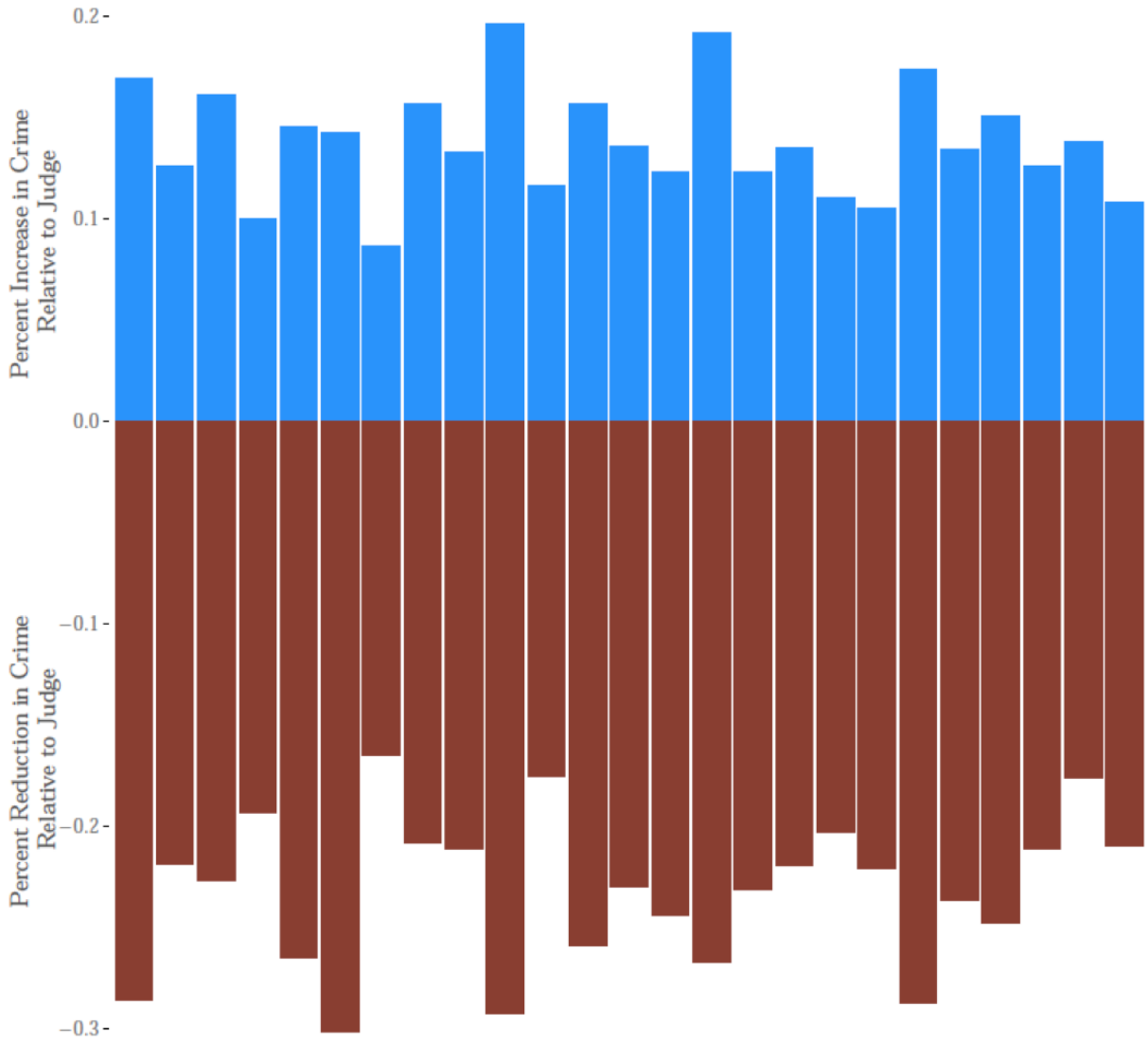

Figure 9: Gains For Each of 25 Judges From Re-Ranking

Notes: This figure shows the potential gains from an algorithmic release rule versus current judge decisions separately for each of the 25 judges in our NYC dataset with the largest caseloads within our test set. We use the algorithm trained on data from all judges and cases in the training set, and then compare potential gains of the algorithm versus each of these judges one at a time (each bar represents comparison to a separate judge) in terms of gains in release rates holding crime rate constant (top panel) and reduction in crime rates holding release rates constant (bottom panel).

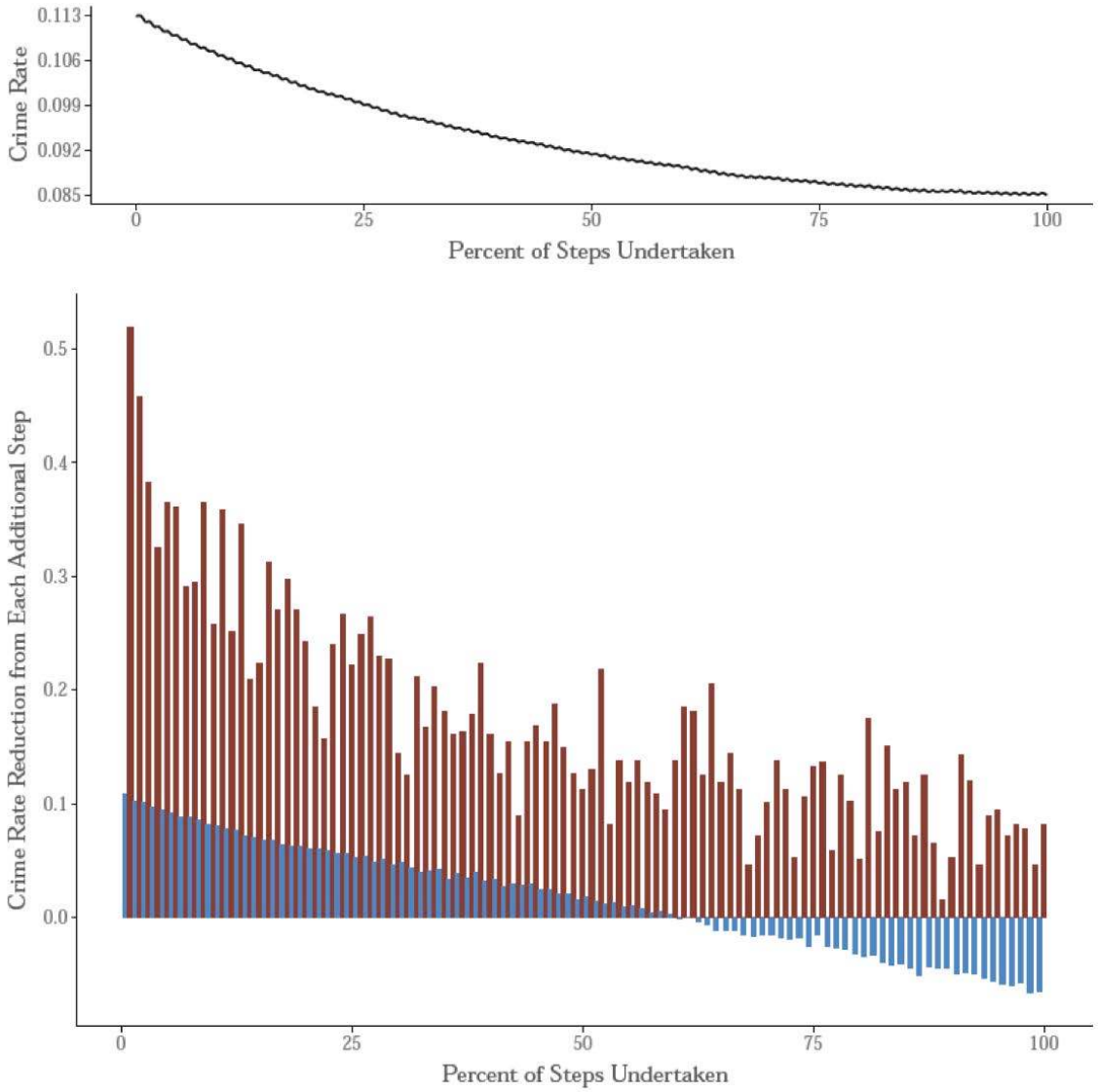

Figure 10: Decomposing Policy Simulation Gains: Gains from Releasing and Jailing

Notes: This figure shows where the total crime gains come from in our NYC data with the algorithmic release rule that re-ranks all defendants by predicted risk, in terms of releasing defendants the judges jailed versus detaining defendants the judge released. We calculate the total set of detain / release "swaps" the algorithmic rule makes compared to current judge decisions and plot the share of those swaps made on the x-axis in both the top and bottom panels. The y-axis in the top panel is —the overall crime rate, starting with the crime rate observed for current judge decisions going down to the crime rate we observe under the algorithmic release rule. The bottom panel shows the reduction in crime that is achieved for each marginal person detained by the algorithm relative to jailing someone with the sample average risk, as well as the reduction in crime that results from releasing the marginal defendant from jail compared to releasing someone with the sample average risk.

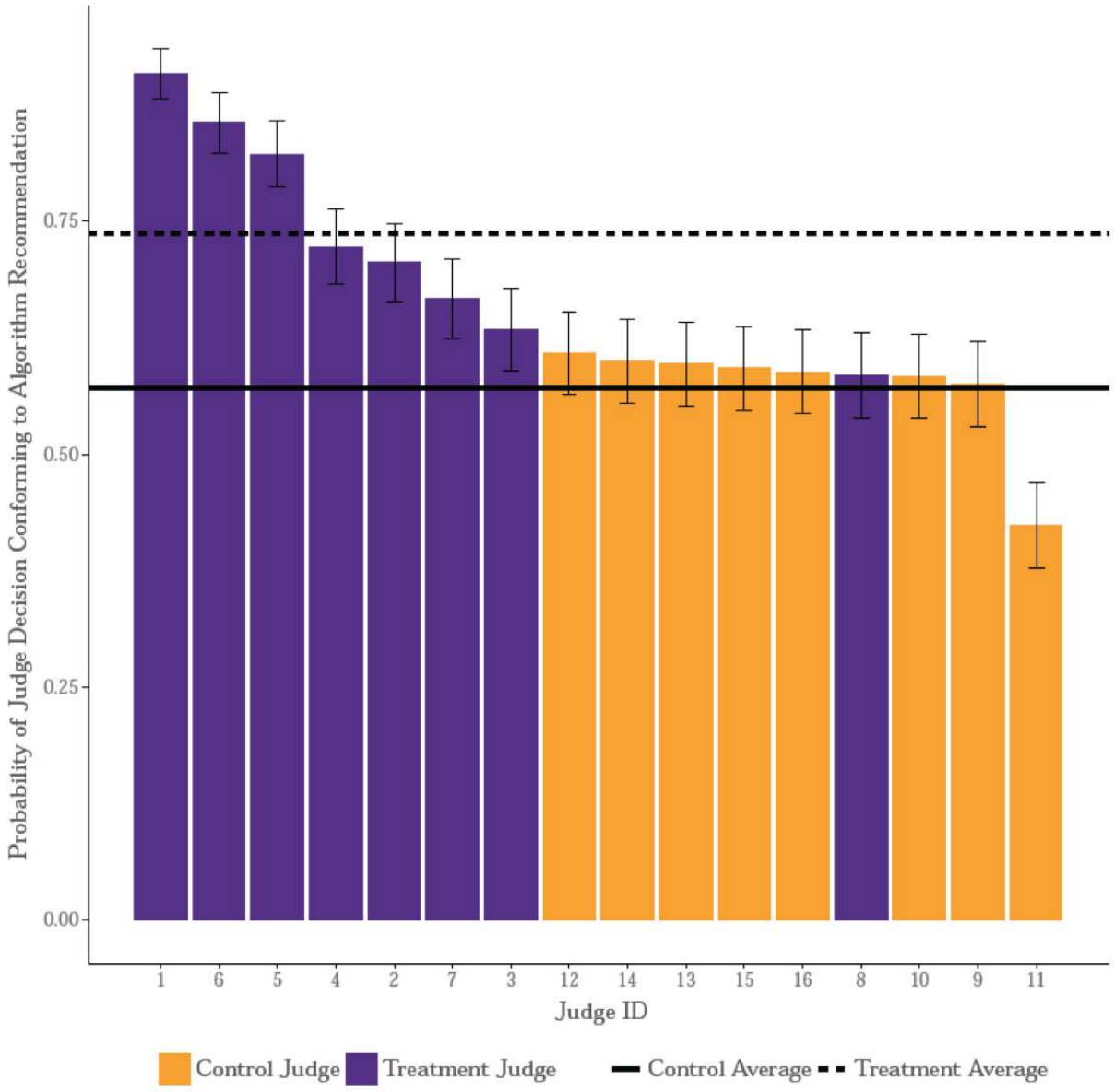

Figure 11: Judges' Response to Risk Tools in Philadelphia

Notes: This figure presents results from the Philadelphia bail experiment (Goldkamp and Gottfredson, 1984) showing judge compliance with the risk tool that was randomly assigned to some judges but not others. Each bar shows the share of a given judge's release decisions that align with the risk tool's recommendation, shaded by whether the judge was given the risk tool (treatment group in purple) or not (control in gold). The horizontal lines in the figure show the average share of decisions that comply with the risk tool recommendations for the treatment group overall (dashed line) and control group (solid line).

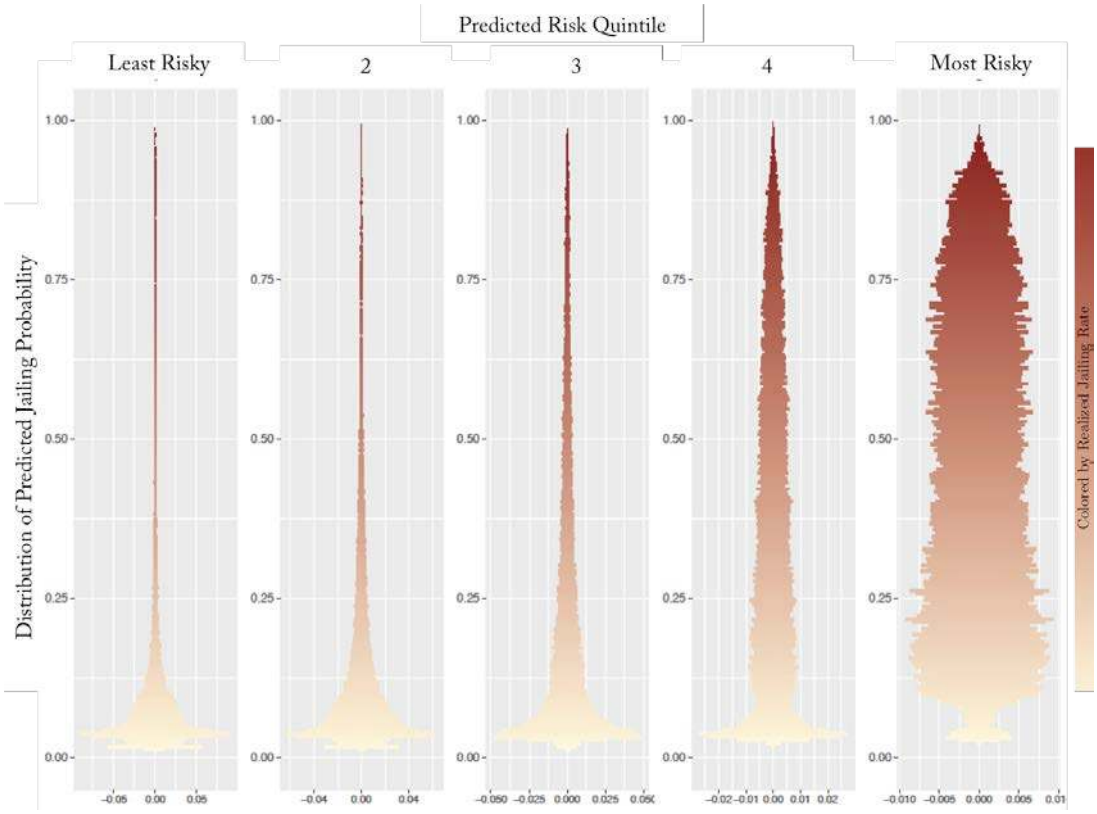

Figure 12: Distribution of Predicted Jail Probabilities By Predicted Crime Risk

Notes: This figure shows the relative "spread" in the predicted judge jail probabilities for cases in our NYC test set grouped together by the algorithm's predicted crime risk. In each predicted risk quintile we graph the distribution (using a volcano plot) of the conditional distribution of predicted judge jailing probabilities. We further color each point by the realized release rate at each rate.

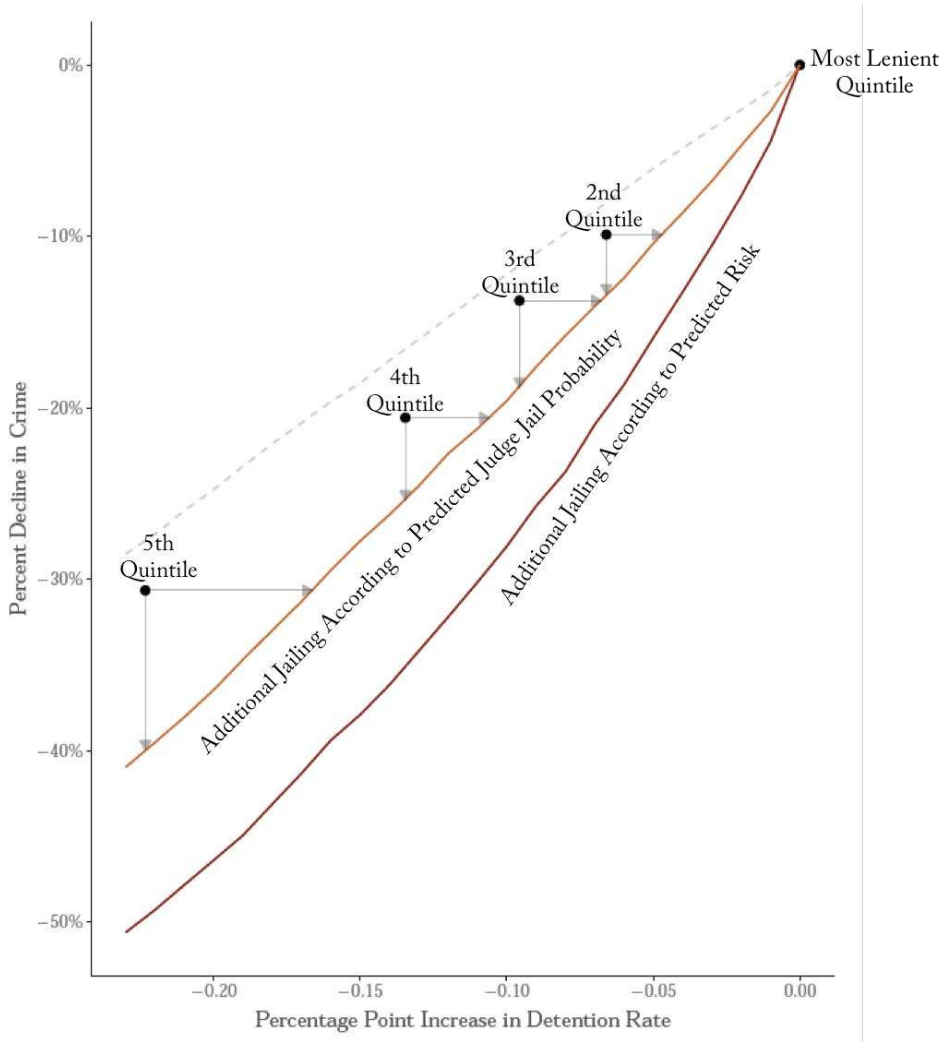

Figure 13: Effect of Detaining Defendants Judges Usually Detain

Notes: This figure compares the change in crime rates and release rates that could be achieved by jailing additional defendants using the algorithm's predicted crime risk compared to the decisions of stricter judges. The right-most point in the graph represents the release rate of the most lenient quintile of judges, with the crime rate that results. The light dashed line shows the decline in crime (as a percentage of the lenient quintile's crime rate, shown on the y-axis) that results from randomly selecting additional defendants to detain from within the lenient quintile's released cases, with the change in release rate relative to the lenient quintile shown on the x-axis. The red curve shows the crime rate / release rate tradeoff that comes from jailing additional defendants within the lenient quintile's released set in descending order of the algorithm's predicted crime risk. The additional curve on the graph shows the crime rate / release rate outcomes we would get from jailing additional defendants within the lenient quintile judges' caseloads in descending order of an algorithm's predicted probability that the judges jail a given defendant. The four points on the graph show the crime rate / release rate outcomes that are observed for the actual decisions made by the second through fifth most lenient quintile judges, who see similar caseloads on average to those of the most lenient quintile judges.

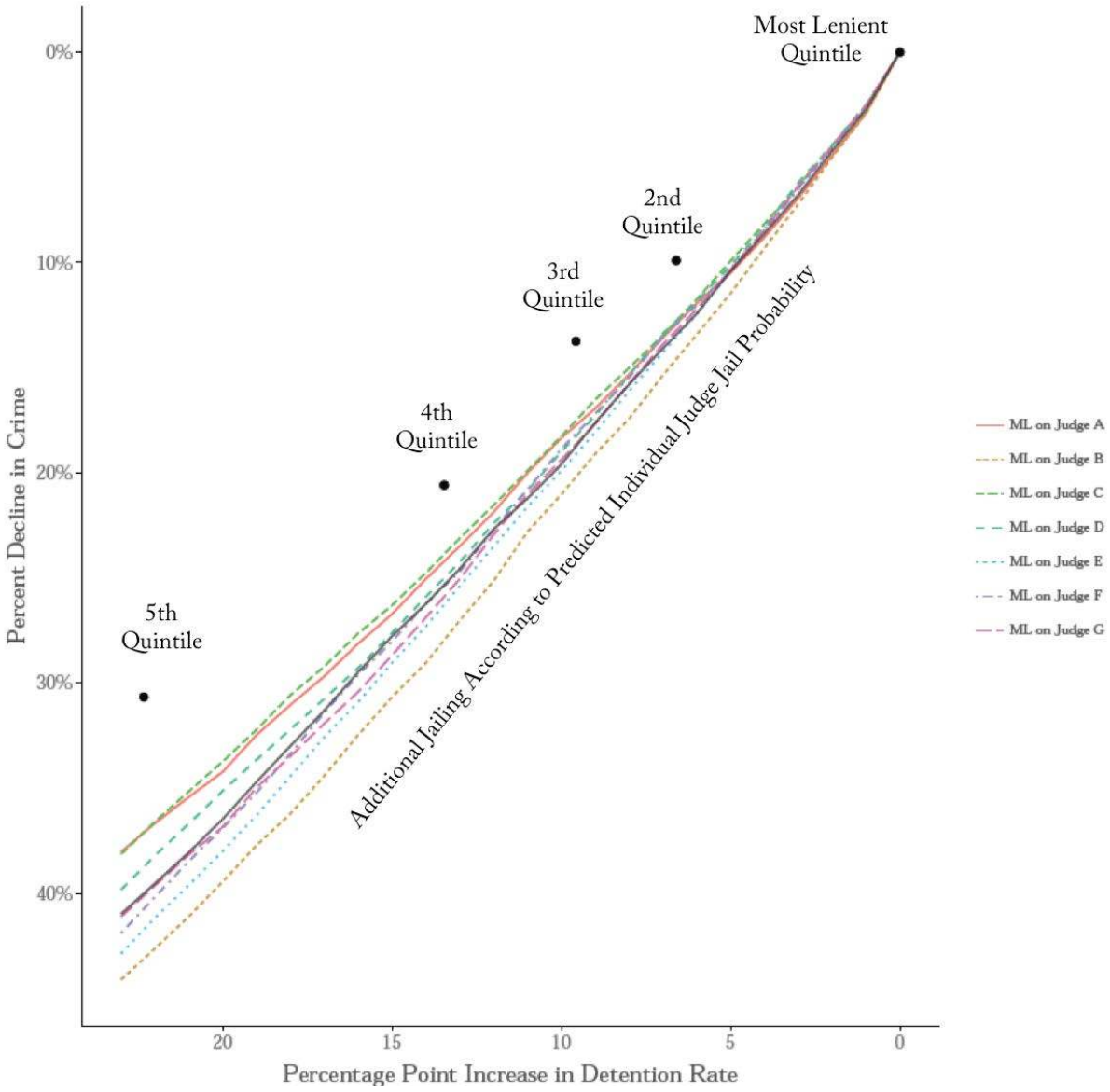

Figure 14: Effect of Detaining Based on Individual Judge Predictions

Notes: This figure looks at performance when additional defendants are jailed according to a predictive model of individual judge's decisions. Each line shows performance of each of seven judges with the highest caseloads. The four points on the graph show the crime rate / release rate outcomes that are observed for the actual decisions made by the second through fifth most lenient quintile judges, who see similar caseloads on average to those of the most lenient quintile judges.

Table 1: Summary Statistics

	Full Sample	Judge Releases	Judge Detains	P-value
Sample Size	554,689	408,283	146,406	
Release Rate	.7361	1.0000	0.00	
Outcomes				
Failure to Appear (FTA)	.1521	.1521		
Arrest (NCA)	.2581	.2581		
Violent Crime (NVCA)	.0372	.0372		
Murder, Rape, Robbery (NMRR)	.0187	.0187		
Defendant Characteristics				
Age	31.98	31.32	33.84	<.0001
Male	.8315	.8086	.8955	<.0001
White	.1273	.1407	.0897	<.0001
African American	.4884	.4578	.5737	<.0001
Hispanic	.3327	.3383	.3172	<.0001
Arrest County				
Brooklyn	.2901	.2889	.2937	.0006
Bronx	.2221	.2172	.2356	<.0001
Manhattan	.2507	.2398	.2813	<.0001
Queens	.1927	.2067	.1535	<.0001
Staten Island	.0440	.0471	.0356	<.0001
Arrest Charge				
Violent Crime				
Violent Felony	.1478	.1193	.2272	<.0001
Murder, Rape, Robbery	.0581	.0391	.1110	<.0001
Aggravated Assault	.0853	.0867	.0812	<.0001
Simple Assault	.2144	.2434	.1335	<.0001
Property Crime				
Burglary	.0206	.0125	.0433	<.0001
Larceny	.0738	.0659	.0959	<.0001
MV Theft	.0067	.0060	.0087	<.0001
Arson	.0006	.0003	.0014	<.0001
Fraud	.0696	.0763	.0507	<.0001
Other Crime				
Weapons	.0515	.0502	.0552	<.0001
Sex Offenses	.0089	.0086	.0096	.0009
Prostitution	.0139	.0161	.0078	<.0001
DUI	.0475	.0615	.0084	<.0001
Other	.1375	.1433	.1216	<.0001
Gun Charge	.0335	.0213	.0674	<.0001
Drug Crime				
Drug Felony	.1411	.1175	.2067	<.0001
Drug Misdemeanor	.1142	.1156	.1105	<.0001

Continued on next page

Table 1 – continued from previous page

	Full Sample	Judge Releases	Judge Detains	P-value
Defendant Priors				
FTAs	2.093	1.305	4.288	<.0001
Felony Arrests	3.177	2.119	6.127	<.0001
Felony Convictions	.6157	.3879	1.251	<.0001
Misdemeanor Arrests	5.119	3.349	10.06	<.0001
Misdemeanor Convictions	3.122	1.562	7.473	<.0001
Violent Felony Arrests	1.017	.7084	1.879	<.0001
Violent Felony Convictions	.1521	.1007	.2955	<.0001
Drug Arrests	3.205	2.144	6.163	<.0001
Felony Drug Convictions	.2741	.1778	.5429	<.0001
Misdemeanor Drug Convictions	1.049	.5408	2.465	<.0001
Gun Arrests	.2194	.1678	.3632	<.0001
Gun Convictions	.0462	.0362	.0741	<.0001

Notes: The top panel reports the observed crime rate for the riskiest 1% of defendants by the algorithm s predicted risk, for different measures of crime using algorithms trained on different crime measures. The first row shows base rates for each type of crime across the columns. In the second row we train the algorithm on failure to appear (FTA) and show for the 1% defendants with highest predicted risk who are observed to commit each different form of crime across the columns. The remaining rows show the results for the top 1% predicted riskiest for an algorithm trained on different forms of crime. The bottom panel shows the potential gains of the algorithmic re-ranking release rule versus the judges (at the judges observed release rate) for each measure of crime shown across the rows, for an algorithm trained on each measure of crime shown in each row.

Table 2: Linear Projection of Prediction Function

Fraction of ML Prediction Variance Explained by Linear Projection					
Adjusted R^2	0.455	0.493	0.468	0.470	0.514
F-Tests of Joint Significance of Categorical Controls					
Control Dummies			*p-values on F-tests*		
Arrest County	<.0001	<.0001	<.0001	<.0001	<.0001
Arrest Month	<.0001	<.0001	<.0001	<.0001	<.0001
Detailed Arrest Charge		<.0001			<.0001
Detailed Prior Arrest			<.0001		<.0001
Detailed Prior Conviction				<.0001	<.0001
Selected Coefficients					
Age	−0.003	-0.004	-0.003	-0.003	-0.003
	(0.00002)	(0.00002)	(0.00002)	(0.00002)	(0.00002)
Current Arrest					
Number of		-0.00001			-0.00001
Arrest Charges		(0.00002)			(0.00002)
Felony		-0.046			-0.047
		(0.082)			(0.080)
Misdemeanor		-0.018			-0.021
		(0.082)			(0.080)
Violent Felony		-0.024			-0.025
		(0.001)			(0.001)
Drug Charge		0.022			0.021
		(0.0004)			(0.0004)
Firearm		-0.020			-0.018
		(0.001)			(0.001)

Continued on next page

Table 2 – continued from previous page

Selected Coefficients					
Defendant Priors					
FTAs	0.020	0.019	0.023	0.023	0.023
	(0.0001)	(0.00005)	(0.0001)	(0.0001)	(0.0001)
Felony Arrests			-0.004		-0.001
			(0.0001)		(0.0001)
Felony Convictions				-0.009	-0.005
				(0.0002)	(0.0002)
Misdemeanor Arrests			-0.001		0.002
			(0.00004)		(0.0001)
Misdemeanor Convictions				-0.003	-0.004
				(0.00004)	(0.0001)
Violent Felony Arrests			0.002		-0.002
			(0.0002)		(0.0002)
Violent Felony Convictions				0.011	0.015
				(0.0005)	(0.001)
Drug Arrests			0.002		-0.003
			(0.0001)		(0.0001)
Drug Convictions				0.005	0.008
				(0.0001)	(0.0002)
Firearm Arrests			-0.004		-0.004
			(0.0004)		(0.0004)
Firearm Convictions				-0.013	-0.008
				(0.001)	(0.001)
Observations	221,876	221,876	221,876	221,876	221,876

Notes: The top panel reports the observed crime rate for the riskiest 1% of defendants by the algorithm s predicted risk, for different measures of crime using algorithms trained on different crime measures. The first row shows base rates for each type of crime across the columns. In the second row we train the algorithm on failure to appear (FTA) and show for the 1% defendants with highest predicted risk who are observed to commit each different form of crime across the columns. The remaining rows show the results for the top 1% predicted riskiest for an algorithm trained on different forms of crime. The bottom panel shows the potential gains of the algorithmic re-ranking release rule versus the judges (at the judges observed release rate) for each measure of crime shown across the rows, for an algorithm trained on each measure of crime shown in each row.

Table 3: Comparing Machine Learning to Logistic Regression

Predicted Risk Percentile	ML/Logit Overlap	Average Observed Crime Rate for Cases Identified as High Risk by:				
		Both ML & Logit	ML Only	Logit Only	All ML Cases	All Logit Cases
1%	30.6%	.6080	.5440	.3996	.5636	.4633
		(.0309)	(.0209)	(.0206)	(.0173)	(.0174)
5%	59.9%	.4826	.4090	.3040	.4531	.4111
		(.0101)	(.0121)	(.0114)	(.0078)	(.0077)
10%	65.9%	.4134	.3466	.2532	.3907	.3589
		(.0067)	(.0090)	(.0082)	(.0054)	(.0053)
25%	72.9%	.3271	.2445	.1608	.3048	.2821
		(.0038)	(.0058)	(.0049)	(.0032)	(.0031)

Notes: The table above shows the results of fitting a machine learning (ML) algorithm or a logistic regression to our training dataset, to identify the highest-risk observations in our test set. Each row presents statistics for the top part of the predicted risk distribution indicated in the first column: top 25% (N=20,423); 10% (8,173); 5% (4,087); and 1% (818). The second column shows the share of cases in the top X% of the predicted risk distribution that overlap between the set identified by ML and the set identified by logistic regression. The subsequent columns report the average crime rate observed among the released defendants within the top X% of the predicted risk distribution as identified by both ML and logit, ML only, and logit only, and all top X% identified by ML (whether or not they are also identified by logistic regression) and top X% identified by logit.

Table 4: The Cost Of Misranking

	Additional Jailings		Judges *Relative to Most Lenient Quintile*		Algorithm *To Achieve Judge's* Δ *Crime*	Δ *Jail*
	All Could Come From Percentile	Percent Actually From Percentile	Δ Jail	Δ Crime	Δ Jail	Δ Crime
Second Quintile	11.98	.332	.066	-.099	.028	-.201
Third Quintile	14.10	.298	.096	-.137	.042	-.269
Fourth Quintile	18.56	.318	.135	-.206	.068	-.349
Fifth Quintile	28.45	.396	.223	-.307	.112	-.498

Notes: This table reports the results of contrasting the cases detained by the second through fifth most lenient quintile judges compared with the most lenient quintile judges. The first column shows from where in the predicted risk distribution each less-lenient quintile s judges could have drawn their marginal detainees to get from the most lenient quintile s release rate down to their own release rate if judges were detaining in descending order of risk. The second column shows what share of their marginal detainees actually come from that part of the risk distribution. The fourth and fifth columns show the increase in detention rates and decrease in crime rates for each of the less-lenient quintile judges observed decisions relative to the most lenient quintile. The fifth column shows the increase in the jail rate that would be required to reach each quintile s reduction in crime rate if we jailed in descending order of the algorithm s predicted risk, while the final column shows the reduction in crime that could be achieved if we increased the jail rate by as much as the judge quintile shown in that row.

Table 5: Policy Simulation Under Different Assumptions

	Assume $y = min(1, \alpha\hat{y})$ for Additional Releases Beyond Most Lenient Judge Quintile's Release Rate						
	Value of α						
	1	1.25	1.5	2	3	$\cdots$	∞
Algorithm's Crime Rate	.0854	.0863	.0872	.0890	.0926		.1049
at Judge's Jail Rate	(.0008)	(.0008)	(.0008)	(.0009)	(.0009)		(.0009)
Percentage Reduction	-24.68%	-24.06%	-23.01%	-21.23%	-18.35%		-14.39%
Algorithm's Jail Rate	.1531	.1590	.1642	.1733	.1920		.2343
at Judge's Release Rate	(.0011)	(.0011)	(.0011)	(.0011)	(.0012)		(.0013)
Percentage Reduction	-41.85%	-40.13%	-38.37%	-34.87%	-29.36%		-18.51%

Notes: In this table we examine the sensitivity of the potential gains in our policy simulation of an algorithmic release rule that re-ranks all defendants by predicted crime risk. We examine the potential gains of the algorithm relative to the judges assuming that the actual crime rate among defendants who the judges jailed and the algorithm releases would be some multiple of the algorithm s predicted crime rate for those defendants (with each defendant s likelihood of crime capped at a probability of 1). As we move across the columns we increase this multiple. The first row shows the crime rate if we jail at the judge s rate but detain in descending order of the algorithm s predicted risk, with percentage gain relative to the judges underneath. The second row shows the reduction in jail rates that could be achieved at the judge s crime rate if we detained in descending order of the algorithm s predicted risk.

Table 6: Predictive Performance on Other Outcomes

Panel A: Outcomes for the 1% Predicted Riskiest

		Outcome Algorithm Evaluated On				
		Failure to Appear	Any Other Crime	Violent Crime	Murder Rape and Robbery	All Crimes
	Base Rate	.1540	.2590	.0376	.0189	.3295
Outcome Algorithm Trained On	Failure to Appear	.5636 (.0173)	.6271 (.0169)	.0611 (.0084)	.0477 (.0075)	.7641 (.0148)
	Any Other Crime	.4425 (.0174)	.7176 (.0157)	.1015 (.0106)	.0672 (.0088)	.7910 (.0142)
	Violent Crime	.2531 (.0152)	.6296 (.0169)	.2225 (.0145)	.1394 (.0121)	.6736 (.0164)
	Murder, Rape and Robbery	.2628 (.0154)	.6222 (.0170)	.1944 (.0138)	.1357 (.0120)	.6797 (.0163)
	All Crimes	.5000 (.0175)	.7127 (.0158)	.0831 (.0097)	.0660 (.0087)	.8117 (.0137)

Continued on next page

Table 6 – continued from previous page

Panel B: Effect of Re-Ranking on Other Outcomes

		Outcome Algorithm Evaluated On				
		Failure to Appear	Any Other Crime	Violent Crime	Murder Rape and Robbery	All Crimes
	Base Rate	.1540	.2590	.0376	.0189	.3295
Outcome Algorithm Trained On	Failure to Appear	.0854	.1697	.0235	.0121	.2135
		(.0008)	(.0011)	(.0005)	(.0003)	(.0012)
	Percentage Gain	-24.68%	-11.07%	-15.03%	-13.27%	-12.05%
	Any Other Crime	.0965	.1571	.0191	.0082	.2084
		(.0009)	(.0011)	(.0004)	(.0003)	(.0012)
	Percentage Gain	-14.96%	-17.67%	-30.9%	-40.9%	-14.15%
	Violent Crime	.1106	.1734	.0157	.0059	.2263
		(.0009)	(.0011)	(.0004)	(.0002)	(.0013)
	Percentage Gain	-2.514%	-9.098%	-43.17%	-57.21%	-6.76%
	Murder, Rape and Robbery	.1096	.1747	.0158	.0059	.2272
		(.0009)	(.0011)	(.0004)	(.0002)	(.0013)
	Percentage Gain	-3.39%	-8.42%	-42.79%	-57.31%	-6.413%
	All Crimes	.0913	.1583	.0201	.0090	.2069
		(.0009)	(.0011)	(.0004)	(.0003)	(.0012)
	Percentage Gain	-19.47%	-17.04%	-27.51%	-35.12%	-14.75%

Notes: The top panel reports the observed crime rate for the riskiest 1% of defendants by the algorithm s predicted risk, for different measures of crime using algorithms trained on different crime measures. The first row shows base rates for each type of crime across the columns. In the second row we train the algorithm on failure to appear (FTA) and show for the 1% defendants with highest predicted risk who are observed to commit each different form of crime across the columns. The remaining rows show the results for the top 1% predicted riskiest for an algorithm trained on different forms of crime. The bottom panel shows the potential gains of the algorithmic re-ranking release rule versus the judges (at the judges observed release rate) for each measure of crime shown across the rows, for an algorithm trained on each measure of crime shown in each row.

Table 7: Racial Fairness

Release Rule	Crime Rate	Drop Relative to Judge	Percentage of Jail Population Black	Hispanic	Minority
Distribution of Defendants (Base Rate)			.4877	.3318	.8195
Judge	.1134 (.0010)	0%	.573 (.0029)	.3162 (.0027)	.8892 (.0018)
Algorithm					
Usual Ranking	.0854 (.0008)	-24.68%	.5984 (.0029)	.3023 (.0027)	.9007 (.0017)
Match Judge on Race	.0855 (.0008)	-24.64%	.573 (.0029)	.3162 (.0027)	.8892 (.0018)
Equal Release Rates for all Races	.0873 (.0008)	-23.02%	.4877 (.0029)	.3318 (.0028)	.8195 (.0023)
Match Lower of Base Rate or Judge	.0876 (.0008)	-22.74%	.4877 (.0029)	.3162 (.0027)	.8039 (.0023)

Notes: Table reports the potential gains of the algorithmic release rule relative to the judge at the judge s release rate with respect to crime reductions and share of the jail population that is black, Hispanic or either black or Hispanic. The first row shows the share of the defendant population overall that is black or Hispanic. The second row shows the results of the observed judge decisions. The third row shows the results of the usual algorithmic re-ranking release rule, which does not use race in predicting defendant risk and makes no post-prediction adjustments to account for race. In the fourth row we adjust the algorithm s ranking of defendants for detention to ensure that the share of the jail population that is black and Hispanic under the algorithmic release rule are no higher than those under current judge decisions. The next row constraints the algorithmic release rule s jail population to have no higher share black or Hispanic than that of the general defendant pool, while the final row constrains the algorithm s jail population to have no higher share black or Hispanic than either the judge decisions or the overall defendant pool.

Table 8: Robustness Checks

	Predictably Riskiest 1%	*To Achieve Judge's*	
		Crime Rate	*Release Rate*
	Crime Rate	Release Rate	Crime Rate
Full Data	.5636	.1531	.0854
	(.0149)	(.0011)	(.0008)
By Borough	.5361	.1543	.086
	(.015)	(.0011)	(.0008)
By Quarter-Year	.556	.1539	.0854
	(.0149)	(.0011)	(.0008)
By Borough-Quarter-Year	.5533	.1558	.0861
	(.0149)	(.0011)	(.0008)
Train on 2008 - 2012	.6274	.1335	.0851
Test on 2013	(.0237)	(.0017)	(.0014)

Notes: Table reports various robustness checks for the re-ranking policy simulation. Each row represents a different robustness check. For each robustness check, in the first column, the predictably riskiest 1% is displayed. The second column displays the release rate needed to achieve the judge's crime rate. The third column displays the crime rate achieved at the judge's release rate. The first row shows the full sample results from the rest of the paper. The second row constrains the release rule to release the same fraction as judges do in each borough. The third row constrains the release rule to release the same fraction as judges do in each quarter-year. The fourth row constrains the release rule to release the same fraction as judges do in each borough-quarter-year cell. The final row trains the algorithm using 2008-2012 data and evaluates its on 2013 data.

Table 9: Judges Versus the Predicted Judge

	Judges *Relative to Most Lenient Quintile*		Algorithm *To Achieve Judge's*	
			Δ *Crime*	Δ *Jail*
	Δ Jail	Δ Crime	Δ Jail	Δ Crime
Second Quintile	.066	-.099	.047	-.134
Third Quintile	.096	-.137	.068	-.188
Fourth Quintile	.135	-.206	.106	-.254
Fifth Quintile	.223	-.307	.166	-.399

Notes: This table replicates the comparison of the algorithmic release rule to the decisions of less lenient quintile judges, but now using an algorithmic release rule based on a model that predicts the release decisions of the judges (our predicted judge model). The first and second columns show the difference in jail rates and crime rates between the 2nd through 5th most lenient quintile judges compared to the most lenient quintile. The third column shows the increase in the jail population that would be required to meet the judges drop in crime if we jailed people in descending order of our prediction that the judges release a case. The fourth column shows the decline in crime that could be achieved if we increased the jail rate the same as the judges do, but detain people in ascending order of the judge predicted release probabilities.

Table 10: Results on National Data Set

Panel A: Predictable Risk Among Released		
	Crime Rate	Release Rate
Riskiest 1%	.7655	.5252
	(.0196)	(.0424)
Average	.3062	.6155
	(.0039)	(.0041)

Panel B: Policy Simulation for Re-Ranking Tool		
	Crime Rate	Release Rate
Judge	.1885	.6155
	(.0033)	(.0041)
ML at Judge Release Rate	.1531	
	(.0031)	
ML at Judge Crime		.7097
		(.0039)

Notes: The table above presents results from applying the predictions of a machine learning algorithm to a national dataset of 151,461 felony defendants assembled by the US Department of Justice (DOJ) for 40 urban counties over a 20 year period. The first panel reports crime rates (crimes per released defendant) by predicted risk among the set of defendants released by the judge. In the first row we identify the top 1% of the predicted risk distribution within each county-year cell and then report the observed rate of crime, which is defined here as either failing to appear in court or re-arrest for a new offense. The second row shows the average crime rate within the set of released defendants. In the bottom panel we report the results of our policy simulation of re-ranking, where we construct a release rule that detains people in descending order of predicted crime risk. We first present the results of the judge s decision the release rate, and the crime rate (defined now as crimes per defendant who passed through bond court), followed by the crime rate that results from the algorithm s rule evaluated at the judge s release rate, followed by the release rate that could be achieved by the algorithmic rule evaluated at the same crime rate that the judges achieve.

A Empirical evidence for conditionally random assignment of cases to judges in New York City

Several of the analyses reported in our paper take advantage of quasi-random assignment of cases to judges in the New York City dataset. This appendix provides details on our statistical tests for whether this assignment process is actually as-good-as random.

As noted in the data section we construct 577 borough, year, month and day of week "cells" in the New York City data where we have at least five judges, which together account for 56.5% of our sample. The summary statistics for this sample are presented in Table A1.

[Table 11 about here.]

To define judge leniency, within each cell we calculate the release rate of each judge and then divide cases up into "judge leniency quintiles" based on the leniency of the judge that hears their case.

We examine balance across these leniency quintiles in the projection of the FTA variable onto the baseline characteristics. That is, we regress the FTA rate of defendant (i) whose case is heard by judge (j) in cell (c), y_{ijc}, against the defendant's baseline characteristics, x_{ijc}, and then retain the predicted value, $\hat{y}_{ijc}$. This essentially creates an index of all the baseline defendant background and case characteristics, weighted in proportion to the strength of their relationship to the main outcome we are examining in our analysis (failure to meet pre-trial release conditions).

Then separately for each borough, year, month and day of week cell, we regress this predicted value against a set of indicators for within-cell judge leniency quintile, Q_{jc}, and calculate the F-test statistic for the null hypothesis that the judge leniency indicators are jointly zero. Call this F_c.

$$\hat{y}_{ijc} = \beta_0 + \beta_1 Q_{jc} + \epsilon_{ijc}$$

We then randomly permute the judge leniency quintiles across cases M=1,000 times within each cell. This provides us with a distribution of F-test statistics calculated under the null hypothesis of no relationship between judge leniency and defendant characteristics, F^*_{ck} for k=1,...M. Then for each cell we calculate:

$$P_c = \frac{1}{M} \sum_k 1(F^*_{ck} > F_c)$$

If defendant characteristics were systematically related to judge leniency, we would expect to see a concentration of our F-test statistics F_c with low p-values. Yet the first panel of Appendix Figure A1 shows that the histogram of P_c values across the 577 cells in our analysis sample does not show unusual mass at low p-values. The subsequent panels show this is similar in each of the individual boroughs in New York, not just citywide.

While these analyses suggest balance in average defendant characteristics, in principle one might worry about the lingering possibility of imbalance at the high end of the $\hat{y}$ distribution which might be particularly relevant for our comparison of inter-judge detention differences on the margin. One way to address this concern is to identify the $\hat{y}$ threshold that corresponds to the full-sample judge release rate of 73.7%, or $P[\hat{y}_{ijc} < \hat{y}^*]] = .737$. When we construct an indicator for each observation in our sample, $D_{ijc} = 1$ if $\hat{y}_{ijc} < \hat{y}^*$, and re-do our balance test as described above, the results are similar. This remains true if we set an even higher $\hat{y}$ threshold to identify cases in the top decile of the predicted risk distribution instead.

B Appendix Tables and Figures

[Figure 15 about here.]

[Table 12 about here.]

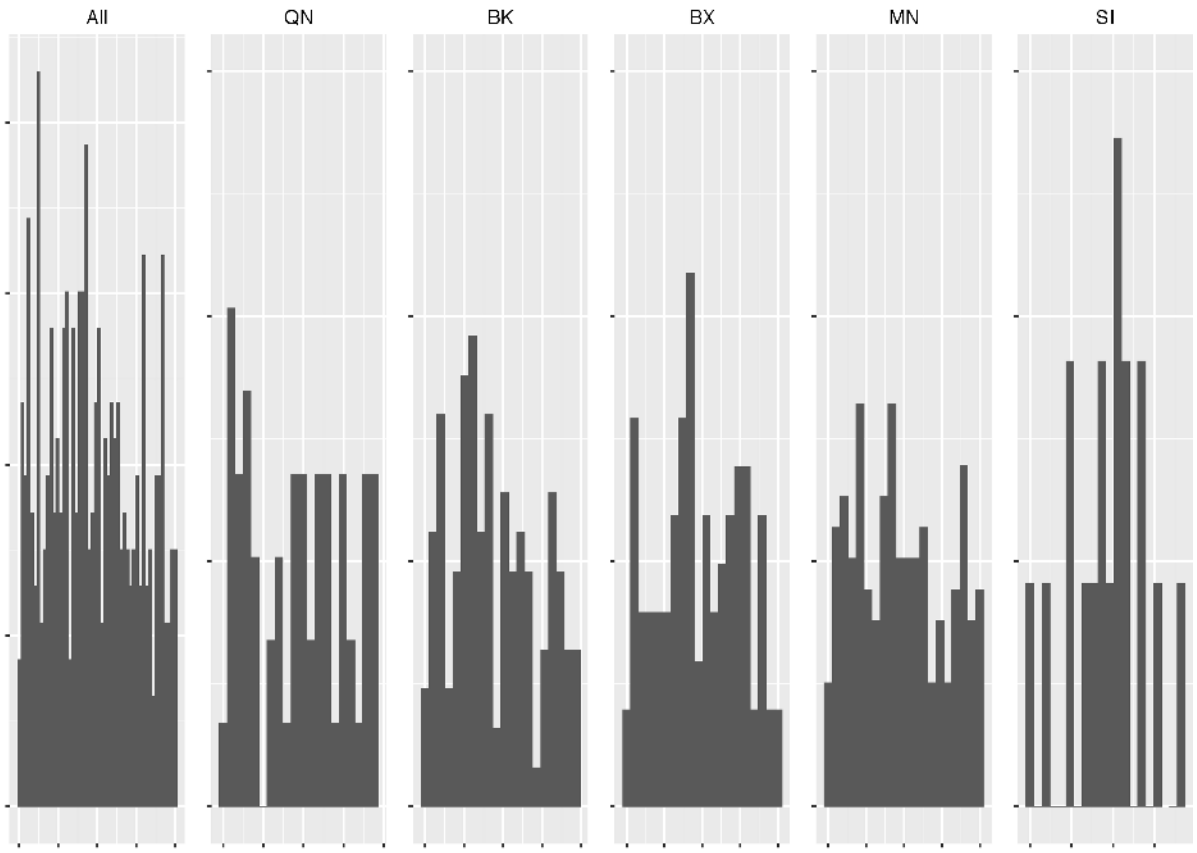

Figure A1: Balance check: Distribution of p-values for Placebo Tests

Table A1: Summary Statistics for Contraction Subsample

	Full Subsample	Judge Releases	Judge Detains	P-value
Sample Size	313,601	230,704	82,897	
Release Rate	.7357	1.000	.00	
Outcomes				
Failure to Appear (FTA)	.1512	.1512		
Arrest (NCA)	.2587	.2587		
Violent Crime (NVCA)	.0370	.0370		
Murder, Rape, Robbery (NMRR)	.0187	.0187		
Defendant Characteristics				
Age	31.97	31.30	33.83	<.0001
Male	.8319	.8087	.8967	<.0001
White	.1207	.1336	.0849	<.0001
African American	.4913	.4603	.5775	<.0001
Hispanic	.3352	.3413	.3180	<.0001
Arrest County				
Brooklyn	.2844	.2823	.2900	<.0001
Bronx	.2279	.2237	.2395	<.0001
Manhattan	.2570	.2461	.2875	<.0001
Queens	.2013	.2161	.1600	<.0001
Staten Island	.0295	.0318	.0230	<.0001
Arrest Charge				
Violent Crime				
Violent Felony	.1493	.1208	.2287	<.0001
Murder, Rape, Robbery	.0592	.0401	.1123	<.0001
Aggravated Assault	.0858	.0872	.0818	<.0001
Simple Assault	.2133	.2425	.1321	<.0001
Property Crime				
Burglary	.0205	.0127	.0421	<.0001
Larceny	.0724	.0644	.0947	<.0001
MV Theft	.0065	.0058	.0085	<.0001
Arson	.0006	.0003	.0014	<.0001
Fraud	.0701	.0766	.0518	<.0001
Other Crime				
Weapons	.0518	.0502	.0562	<.0001
Sex Offenses	.0090	.0088	.0097	.0155
Prostitution	.0139	.0161	.0079	<.0001
DUI	.0474	.0615	.0081	<.0001
Other	.1385	.1446	.1215	<.0001
Gun Charge	.0337	.0214	.0678	<.0001
Drug Crime				
Drug Felony	.1427	.1189	.2088	<.0001
Drug Misdemeanor	.1125	.1138	.1087	.0001

Continued on next page

Table A1 – continued from previous page

	Full Subsample	Judge Releases	Judge Detains	P-value
Defendant Priors				
FTAs	2.0928	1.3007	4.2971	<.0001
Felony Arrests	3.182	2.118	6.141	<.0001
Felony Convictions	.6206	.3906	1.261	<.0001
Misdemeanor Arrests	5.127	3.34	10.10	<.0001
Misdemeanor Convictions	3.128	1.555	7.506	<.0001
Violent Felony Arrests	1.018	.7069	1.883	<.0001
Violent Felony Convictions	.1536	.1014	.2987	<.0001
Drug Arrests	3.205	2.142	6.162	<.0001
Felony Drug Convictions	.2759	.1785	.5468	<.0001
Misdemeanor Drug Convictions	1.048	.5392	2.465	<.0001
Gun Arrests	.2200	.1685	.3635	<.0001
Gun Convictions	.0467	.0365	.0750	.0001

Table A2: Summary Statistics for National Sample

	Full Sample	Judge Releases	Judge Detains	P-value
Sample Size	140,538	85,189	55,349	
Release Rate	.6062	1.0000	.0000	
Outcomes				
Failure to Appear (FTA)	.2065	.2065		
Arrest (NCA)	.1601	.1601		
Violent Crime (NVCA)	.0179	.0179		
Murder, Rape, Robbery (NMRR)	.0094	.0155	.0000	
Defendant Characteristics				
Age	30.3691	30.1134	30.7626	<.0001
Male	.8300	.7948	.8842	<.0001
White	.2457	.2702	.2080	<.0001
African American	.3832	.3767	.3933	<.0001
Hispanic	.2188	.1927	.2590	<.0001
Arrest Charge				
Violent Crime				
Violent Felony	.2488	.2215	.2909	
Murder, Rape, Robbery	.0908	.0615	.1359	<.0001
Aggravated Assault	.1191	.1211	.1160	.0039
Other Violent	.0389	.0389	.0390	.9165
Property Crime				
Burglary	.0866	.0682	.1150	<.0001
Larceny	.0934	.1033	.0780	<.0001
MV Theft	.0306	.0237	.0412	<.0001
Fraud	.0288	.0374	.0154	<.0001
Other Crime				
Weapons	.0312	.0328	.0287	<.0001
DUI	.0296	.0352	.0210	<.0001
Gun Charge	.0000	.0000	.0000	
Other	.0344	.0343	.0345	.8020
Drug Crime				
Drug	.3471	.3635	.3219	
Drug Felony	.3471	.3635	.3219	<.0001
Drug Sale	.1702	.1740	.1643	<.0001
Drug Possession	.1769	.1895	.1576	<.0001
Drug Misdeamenor	.0000	.0000	.0000	

Continued on next page

Table A2 – continued from previous page

	Full Sample	Judge Releases	Judge Detains	P-value
Defendant Priors				
FTAs	.3125	.2586	.3955	<.0001
Felony Arrests	2.9355	2.2686	3.9618	<.0001
Felony Convictions	1.1862	.8170	1.7544	<.0001
Misdemeanor Arrests	3.1286	2.6030	3.9375	<.0001
Misdemeanor Convictions	1.6808	1.2591	2.3298	<.0001
Violent Felony Convictions	.1046	.0735	.1526	<.0001

Notes: This table shows descriptive statistics overall and by judge release decision for the 140,538 cases that serve as our national analysis dataset. The probability values at right are for pair-wise comparison of the equivalence of the mean values for the released versus detained defendants.