HUMAN DECISIONS AND MACHINE PREDICTIONS

JON KLEINBERG, HIMABINDU LAKKARAJU, JURE LESKOVEC, JENS LUDWIG, SENDHIL MULLAINATHAN

February 2, 2021

Motivation

- Many important decisions hinge on a prediction: managers assess future productivity for hiring; lenders forecast repayment; doctors form diagnostic and prognostic estimates
- This raises the question, Could we use statistically driven predictions to improve decision making in these prediction policy problems
- We study one example, significant in its own right, to understand the promise of using machine learning to improve - Bail decisions

Motivation

- Each year in the United States, the police arrest over 10 million people.
- Soon after arrest, a judge decides where defendants will await trial, at home or in jail. By law, this decision should be based solely on a prediction: what will the defendant do if released? Will they flee or commit a new crime?
- In principle an algorithm could also make these predictions.
- To answer this the researchers build an algorithm, based on bail past decisions, and then suggest a method to evaluate its efficiency

Empirical Strategy

- The empirical analysis consists of of two steps: train an algorithm, and then evaluate its performance.
- In the first step the construct a gradient boosted decisions trees to fit some function $m(x)$ that spits out probabilities $P(Y = 1|x)$ to commit a crime
- The algorithm AUC is 0.707, implying its doing a good job predicting crime **for those who were released by judges** (Standard procedure in evaluating ML algorithms)

Empirical Strategy

- Let X be observed characteristics and Z characteristics observed by the judge and w is unobserved variable affecting judge decision, but not the risk. Assume that

$$E(y|X,Z) = x + z$$

- The judges payoff function is given by

$$\pi^j(y,R) = \underbrace{-a_j yR}_{\text{Crime Cost}} - \underbrace{b_j(1-R)}_{\text{Incarceration Cost}},$$

- let the decision rule of the judge be

$$\rho^j(x,z,w) = 1 \text{ if and only if } h_j(x,z,w) < \kappa_j \equiv \frac{b_j}{a_j}$$

- Assume that judges draw cases from the same pool

Empirical Strategy

- The basic question we address is whether a given algorithm's predictions m(x) can improve upon judicial predictions

$$\Pi^j\left(\rho^d\right) - \Pi^j\left(\rho^j\right) = -a_j \underbrace{\left(\bar{R}^d E\left[y \mid \rho^d = 1\right] - \bar{R}^j E\left[y \mid \rho^j = 1\right]\right)}_{\Delta \text{ Crime}}$$

$$- b_j \underbrace{\left(\bar{R}^j - \bar{R}^d\right)}_{\Delta \text{ Release}},$$

- While we can change the threshold of the algorithm to zero out the second term, the first term poses a selection issue, as the difference between the two risks is given by

$$-\underbrace{E\left[y \mid \rho^d = 0, \rho^j = 1\right]}_{\text{Measured}} + \underbrace{E\left[y \mid \rho^d = 1, \rho^j = 0\right]}_{\text{Unmeasured}}$$

- It also not likely that we can assume selection on observables here

Empirical Strategy

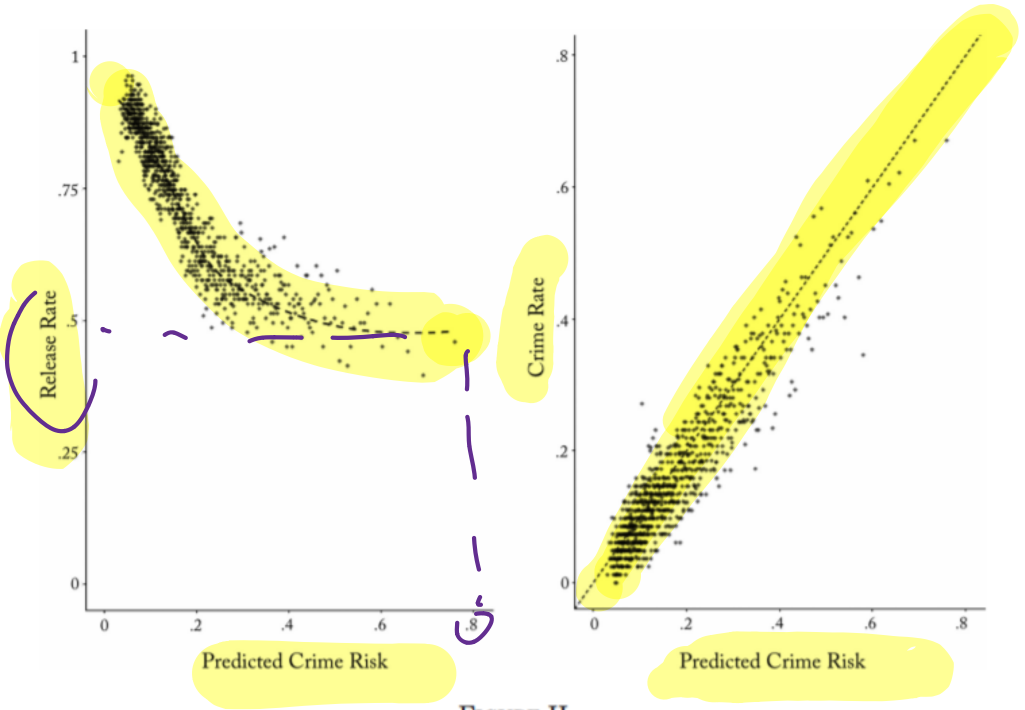


FIGURE II

How Machine Predictions of Crime Risk Relate to Judge Release Decisions and Actual Crime Rates

The figure 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 I). Both panels show the algorithm's predicted crime risk (defined here as predicted risk for failure to appear, or FTA) on the *x*-axis: each point represents one of 1,000 percentile bins. The left panel shows the release rate on the *y*-axis; the right panel shows the realized crime risk on the *y*-axis.

Empirical Strategy

- We see that there's correlation between the algorithm predicted risk and judges decisions
- We can see that the algorithm is doing a relatively good job in predicting crime - so selection is not the whole story here
- The issue is that we can't quantify whether the algorithm helps without known the utility function of the judges

Empirical Strategy

- We see that there's correlation between the algorithm predicted risk and judges decisions
- We can see that the algorithm is doing a relatively good job in predicting crime - so selection is not the whole story here
- The issue is that we can't quantify whether the algorithm helps without known the utility function of the judges

Empirical Strategy

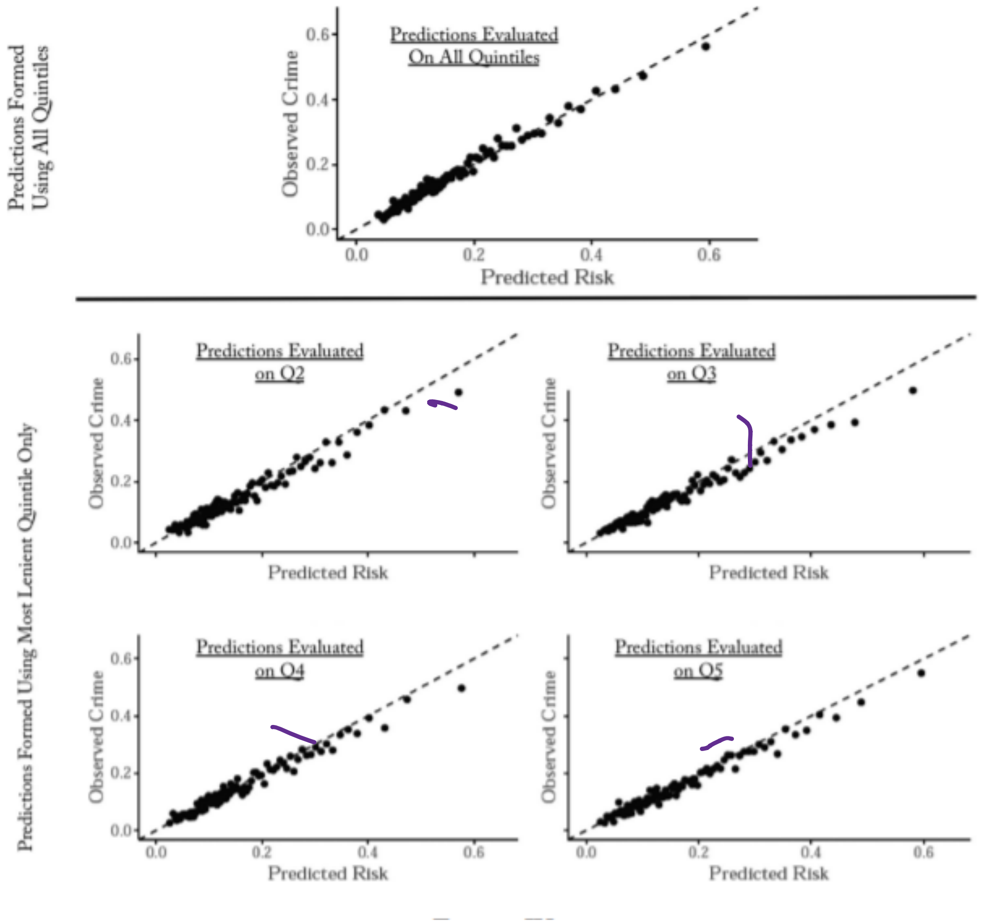
- To overcome this issue, the authors suggest using the different leniency of judges
- To do so they impose some additional assumptions
  - Judges are randomly assigned
  - There's heterogeneity in leniency
  - Judges ability to select on unobservables is the same. Let $\bar{z}^j(x, l)$ be the average unobservables for judge with release l, for cell x. This assumption implies that

$$(\forall l, x) : \bar{z}^1(x, l) = \bar{z}^2(x, l)$$

- We can examine a weaker assumption

$$\bar{z}_1\left(x, \bar{R}^1(x)\right) = \bar{z}_2\left(x, \bar{R}^2(x)\right)$$

Empirical Strategy



- With these assumptions we can explore the following decision rule

$$\rho^C: \text{ Release if and only if } \rho^1 = 1 \text{ and } m(x) < k\,,$$

for some constant k.

- Notice that we have

$$E\left[y \mid \rho^C = 1\right] = E\left[y \mid \rho^1 = 1, m(x) < k\right]$$

- Which allows us to measure how helpful the algorithm by observing

$$\Pi^2\left(\rho^C\right) - \Pi^2\left(\rho^2\right) = a_2\left(E\left[y \mid \rho^C = 1\right] - E\left[y \mid \rho^2 = 1\right]\right)$$

$$-b_2\left(\bar{R}^2 - \bar{R}^C\right)$$

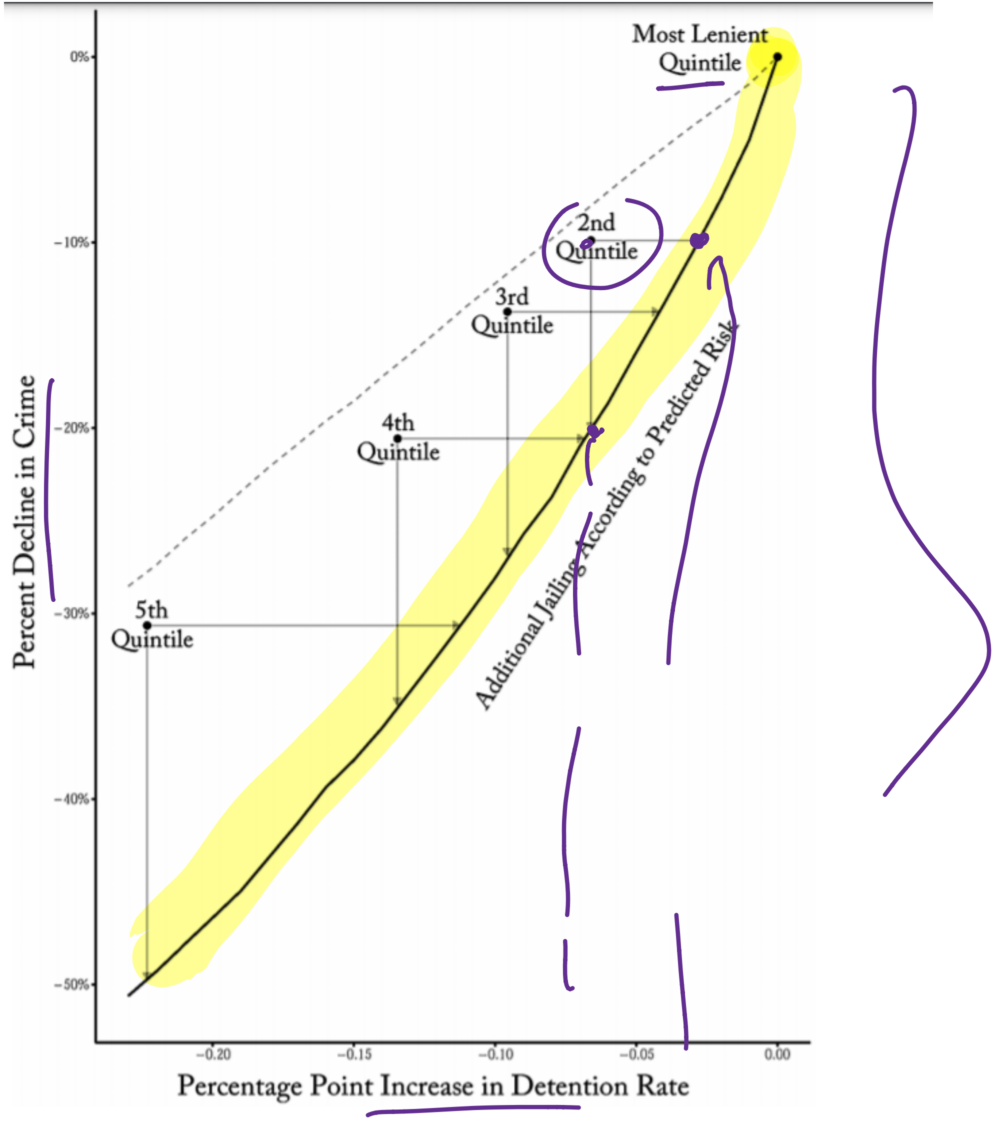
Empirical Strategy

- To overcome this issue, the authors suggest using the different leniency of judges
- To do so they impose some additional assumptions
  - Judges are randomly assigned
  - There's heterogeneity in leniency
  - Judges ability to select on unobservables is the same. Let $\bar{z}^j(x, l)$ be the average unobservables for judge with release l, for cell x. This assumption implies that

$$(\forall l, x) : \bar{z}^1(x, l) = \bar{z}^2(x, l)$$

- We can examine a weaker assumption

$$\bar{z}_1\left(x, \bar{R}^1(x)\right) = \bar{z}_2\left(x, \bar{R}^2(x)\right)$$


Most Lenient Quintile
2nd Quintile
3rd Quintile
4th Quintile
5th Quintile
Additional Jailing According to Predicted Risk
Percent Decline in Crime
0%
−10%
−20%
−30%
−40%
−50%
−0.20
−0.15
−0.10
−0.05
0.00
Percentage Point Increase in Detention Rate

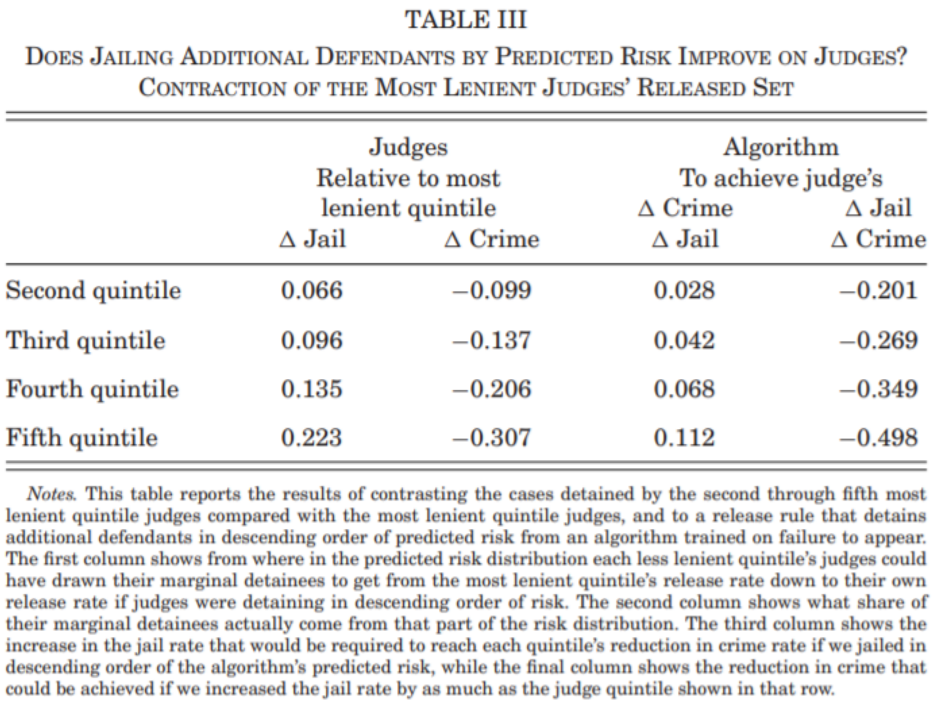


TABLE III

DOES JAILING ADDITIONAL DEFENDANTS BY PREDICTED RISK IMPROVE ON JUDGES? CONTRACTION OF THE MOST LENIENT JUDGES' RELEASED SET

	Judges: Relative to most lenient quintile		Algorithm: To achieve judge's	
	Δ Jail	Δ Crime	Δ Crime: Δ Jail	Δ Jail: Δ Crime
Second quintile	0.066	–0.099	0.028	–0.201
Third quintile	0.096	–0.137	0.042	–0.269
Fourth quintile	0.135	–0.206	0.068	–0.349
Fifth quintile	0.223	–0.307	0.112	–0.498

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, and to a release rule that detains additional defendants in descending order of predicted risk from an algorithm trained on failure to appear. 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 third 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.