Human Decisions and Machine Predictions

Daniel Montero Rivas

Faculty of Economics,
University of Cambridge

February 13, 2026

Background and Motivation I

- If ML is surprisingly effective at a variety of tasks traditionally associated with human intelligence, why not examine it in the context of judicial decisions?
- Judges decide whether defendants must wait in jail while their legal fate is being decided.
- What will the defendant do if released?
- Is it possible to understand and improve judges' decisions?

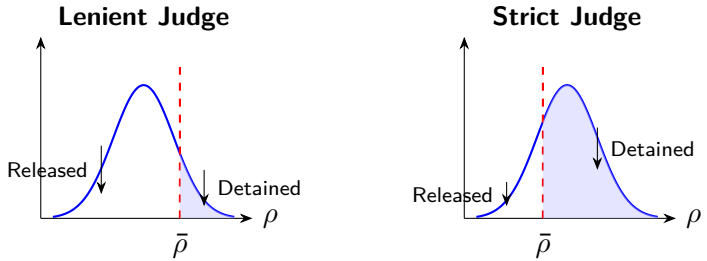
Background and Motivation II

- Trained gradient-boosted decision trees machine learning algorithm on defendant characteristics (input) to predict crime risk (output) in probabilistic terms.
- $758,027$ defendants who were arrested in New York City between $2008 - 2013$.
- Data is about defendants who were released pre-trial and if so, whether they went on to commit a new crime(s).
- But, what if defendants were sent to jail in the meantime?
- Counterfactual (inference problem): What jailed defendants would have done if released?

Background and Motivation III

- Judges' decisions $\neq$ algorithm's predictions.
- $\Rightarrow$ defendants flagged by the algorithm as high risk are treated by the judge as if they were low risk.
- Defendants the algorithm predicted to be risky do in fact commit many crimes.
- Reasons: judges' mis-prediction or high risk threshold for detention.
- Data only on judges' decisions, not on their predictions of risk. Therefore, the latter reason can only be evaluated indirectly.

Background and Motivation IV (a)



ρ represents risk/propensity score. The threshold $\bar{\rho}$ determines detention decisions.
Lenient judges have higher thresholds (fewer detentions), strict judges have lower thresholds (more detentions).

Background and Motivation IV (b)

- Thus, we can see if judges' implicit risk rankings match the algorithm's by looking at who they jail as they become stricter.
- Empirically, which judge hears a case is essentially an arbitrary consequence of who was working in the courtroom when the case was heard.
- $\Rightarrow$ **as-good-as randomly assignment** to judges to overcome inference problem created by not knowing what the jailed defendants would have done if released.

Background and Motivation V

- Exercise shows: judges are not simply setting a high threshold for detention but are mis-ranking defendants.
- Stricter judges do not simply jail the riskiest defendants but the marginal defendants they select to detain are drawn from throughout the entire predicted risk distribution.
  - Judges in the second quintile of leniency could have accomplished their additional jailings by simply detaining everyone in the riskiest 12% of the defendant risk distribution, but only 33.2% of their additional jailings come from this high-risk tail, and the remaining jailings are from lower-risk people.
- Mis-rankings are costly.
  - Relative to most lenient judges, stricter judges increase jailings by 13.0% and reduce crime rates by 18.7%.
  - If detained in order of predicted risk: (1) same reduction in crime with a 6.3% increase in jailings; (2) same jailing rate, crime could have been reduced by 32.9%.
  - Large welfare gains from using algorithmic predictions in the ranking and release process.

Background and Motivation VI: Counter-factuals to scope the size of potential gains

1. Re-ranking = shuffling of judge decisions (jailing some people the judges released and releasing some they jailed).
   - The counterfactual of jailing someone who had been released is easy (no crimes) but, what crimes might the jailed commit if they had been released instead?
   - Solution: impute outcomes for these defendants using outcomes of other defendants with similar observables who judges did release.
   - At the same release rate as judges, the algorithm could produce 24.7% fewer crimes (at the same crime rate observed, the algorithm could jail 41.8% fewer people).
   - Problem: **strong 'selection of observables' assumption** while unobservable variables seen by judges could bias the results.

Background and Motivation VII: Counter-factuals to scope the size of potential gains

- What if judges and society care about outcomes other than crime? E.g., racial equity (not explicit input).
  - Algorithm then reduces crime at the expense of other goals.
  - Race may correlate with other inputs.
  - Appropriately done re-ranking simultaneously reduce jailings, crime and racial disparities.

Background and Motivation VIII: Counter-factuals to scope the size of potential gains

- What if judges have a more complicated set of preferences over different kinds of crimes than assumed? E.g., violent crimes.
  - Algorithm may be reducing overall crime while increasing violent crimes judges care most about.
  - However, similar reductions across crimes. Different types of crimes correlated enough.

Background and Motivation IX: Why are judges mis-predicting?

- Judicial decisions too variable.
- Trained algorithm to predict whether judges will release a defendant with a given set of characteristics or not $\neq$ judge's decision $\Rightarrow$ unobserved variables reflect private info.
  - To test it, **release rule** based on the predicted judge, which orders defendants by the algorithm's predicted probability that the judge will release them, and then release the defendants in this order.
  - Predicted judge outperforms actual judges by a wide margin.
  - Unobservables create noise (mis-prediction), not signal (private info.).

Background and Motivation X: Similar patters

- Previous findings not unique to New York.
- With broader geographic coverage and different outcome variables:
  - Judges releasing predictably very risky defendants.
  - Algorithm could reduce crime by 18.8% holding the release rate constant.
  - Algorithm could jail 24.5% fewer people, holding the crime rate constant.

Data & Context

- Judges decide whether 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 $\Rightarrow$ defendant's crime risk.
- All arrests in NYC between November 1, 2008 and November 1, 2013 ($1,460,462$ cases).
  - For each defendant, the judge has information about instant offense and rap sheet + demeanor and what they wear.
  - Dataset: age, outcome of each case, FTA, pre-trial release, any re-arrest prior to resolution of the case.
  - Of the initial sample, $758,027$ were subject to pre-trial release (relevant for the study). Randomly:
    - $203,338$ cases in a "lock box".
    - $554,689$ cases used in the algorithm.

Broad Picture

1. Train a prediction algorithm with defendant features and outcome (training data).
2. Assess accuracy by taking the predictions given by $m(x)$ to new data (hold-out; to avoid over-fitting).

Machine Learning Black Box

ML is fed with:

- Set of input features x and outcome y to be predicted.
- Class of allowable functions m used to build a prediction: $y = m(x)$ s.t. $m(x)$ generate probability values.
- Bernouilli loss function $L(y, \hat{y})$ that quantifies the cost of prediction errors. Goal is to minimize $L(\cdot)$ to guide $m(x)$ to generate accurate predictions out of sample.

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

Algorithm: gradient boosted decision trees

- In standard econometric models, researcher chooses the set of independent variables of the model (pre-specified and usually a small number) and a fixed functional form s.t. the result set of coefficients fit the dependent variable.
  - And they do not change with data size or other features.
- ML relies on a complexity parameter(s) that is measured differently depending on the model chosen.
  - It can be decided based on data themselves (part of the estimation procedure).
  - More complexity fits data better but, we do not want to overfit it. We want a better out of sample fit (logic of the fifth fold next figure).
  - The chosen complexity parameter is the one that yields the lowest prediction loss in the held-out fold.
  - More data trumps better algorithms.

Procedure and Data Use

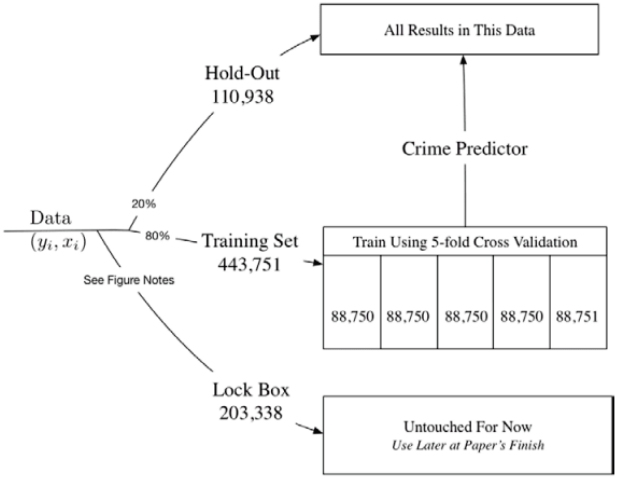


Figure 1: Schematic for Data Flow and use

Evaluating Results

- The algorithm is built on data for released defendants (observed both crime outcomes and case characteristics) to compute each defendant's predicted risk $m(x_i)$.
- And we evaluate the results in test set.
  - Judge's decision $h(x_i, z_i)$ is a function of observables and unobservables z_i, whereas the algorithm is only a function $m(x_i)$ of observables.
  - Interested in decision quality (how well judges sort defendants: $m(x_i)$ VS. $h(x_i, z_i)$) rather than prediction quality (how $m(x_i)$ compares to y_i within the released set).
  - However, only observe release decisions $R_i \in 0, 1$.
  - In the paper, authors compare machine predictions $m(x_i)$ to the predictions implicit in judge release decision assuming judges release in order of their own predictions of defendants: $R_i = f(h(x_i, z_i)) = 1(h(x_i, z_i) < h^*)$.

How risky are the riskiest people judges release

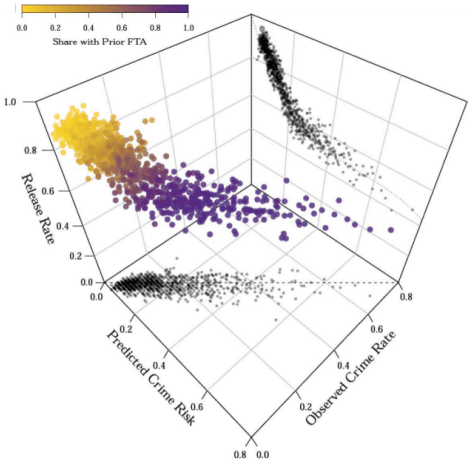


Figure 2: Outcomes for the Released Ranked by Predicted Risk

- Low end of the predicted risk distribution (where most defendants are concentrated), judges release at a rate of over 90%.
- As predicted risk ↑, the judge release rate ↓ ⇒ judges' and algorithm's predictions correlated.
- Disagreement at the high end of the risk distribution ⇒ judges treat many people with high predicted risk as if they were low risk.
- People the algorithm predicts risky are indeed risky ⇒ defendants the judges release do not seem to have unusual unobservables.
- Defendants predicted to be the riskiest 1% by the ML algorithm go on to have an observed crime rate of $\bar{y} = 56.3\%$. Well calibrated.

Mis-rankings or high detention thresholds?

- Judges may place a high cost on jailing defendants, which would lead them to detain only those with even higher risk.
- We do not observe judge predictions of each defendant's risk $h(x_i, z_i)$.
- Looking across caseloads of judges with different release rates allow us to uncover the implicit risk-ordering (risk of the marginal defendants detained). See graph slide 7.
  - This procedure does not rely on judges having similar rank-orderings.
  - This procedure makes a specific assumption about the distribution of unobservables: within each predicted risk bin, different judges have similar distributions of unobservables amongst the released (same average risk).

Mis-rankings or high detention thresholds?

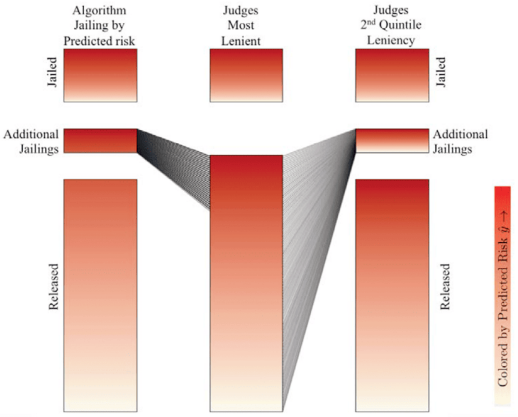


Figure 3: Who is Jailed as Judges Become More Stringent?

- Top box 17.1% of defendants jailed; bottom box 82.9% of defendants released by most lenient.
- LH panel: algorithm would select marginal defendants to detain if judges were detaining defendants in descending order of predicted risk (to the second-most-lenient quintile); RH panel: what judges actually do.
- Instead of selecting all the marginal defendants from the highest-risk 12% of the distribution, only 33.2% come from the riskiest tail.
- **Judges are choosing to jail many low-risk defendants ahead of those with higher predicted risk**.

Mis-rankings or high detention thresholds?

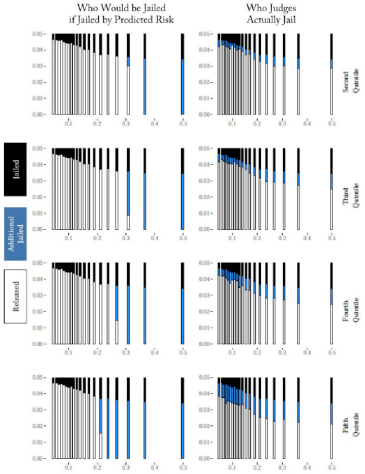


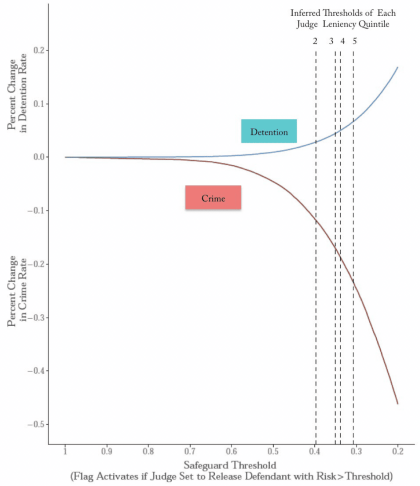
Figure 4: Who do Stricter Judges Jail? Predicted Risk of Marginal Defendants

- Similar results for all judge-leniency quintiles.
- From RH panel, judges select marginal defendants from throughout the predicted-risk distribution.
- Judges do not seem to disagree much with each other about how to rank-order defendants based on their observable characteristics.
- That is, all leniency quintiles behave as the second quintile (previous graph): jailing predictably low-risk individuals while high-risk ones are available.

Who's right?

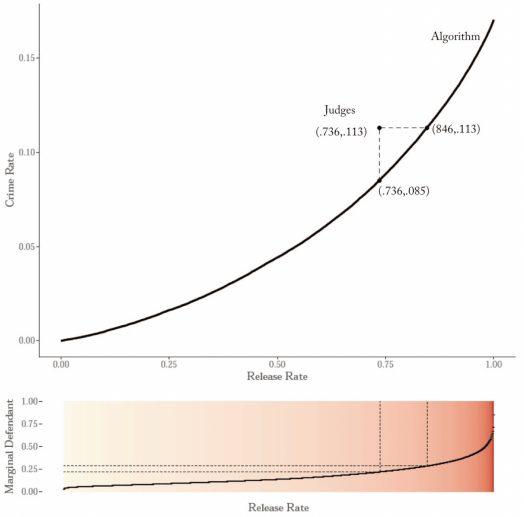
	Additional Jailings		Judges *Relative to Most Lenient Quintile*		Algorithm *To Achieve Judge's* Δ *Crime*	Algorithm *To Achieve Judge's* Δ *Jail*
	All Could Come From Percentile	Percent Actually From Percentile	Δ Jail	Δ Crime	Δ Jail	Δ Crime
Second Quintile	11.98	.332	.066	-.099	.028	-.201
Third Quintile	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

Contraction



- We found judges mistakenly release predictably-risky defendants.
- Policy: warning when such an error is about to happen ("driver assist").
- This policy *contracts* the release set.
- Problem: contraction limits us to jailing more people; do not take advantage of the algorithm's ability to select people for release.
- Need to set a threshold risk that activates the warning.
- Policy could create steep crime reductions for modest jailing increases.

Alternative policy: re-ranking



- This is an "auto-pilot" policy that sets any jail rate.
- Rank-order all defendants according to predicted risk and make both jail and release recommendations to judges.
- Release rule: 40% working sample is training data, 40% imputation, 20% test; train GBDTs to get crime rate predictions; rank-order defendants in test set; for a release rate x, algorithm releases the predictably riskiest fraction x of defendants.
- Dominance argument since we do not know judges' and society's utility functions.

Confounders

1. **Omitted payoff bias**: judges might have additional objectives beyond the outcome the algorithm is predicting, such as risk of re-arrest (and not only flight risk) or racial equity.
   - Algorithm focused on FTA risk would reduce both FTA rates and the overall re-arrest rates for every crime among released defendants.
   - Algorithm achieves the same share minority jailed as by current judge decisions without information on race or ethnicity.
2. Algorithm may outperform judges due to some constraints that bind judges' decisions such as jail capacity.
   - However, after accounting for this concern, large social-welfare gains from releasing defendants using the algorithm rather than judges' predictions.
3. **Algorithm (in)stability**: changing over time in ways that attenuate the potential gains of the algorithm relative to the judge decisions (over-state potential gains from adopting an algorithmic release rule).
   - Yet no signs of instability.

Conclusions

- Reducing jail and prison populations without increasing crime is a key policy priority.
- Instead of focusing on causal questions, the paper focuses on improved prediction.
- Identifying high risk defendants reduces crime rates by 25%, holding release rates constant; or reduces pre-trial jailing rates by 40% with no increase in crime.
- To solve the Data $\longrightarrow$ Prediction $\longrightarrow$ Decision, ML requires a search of the prediction function with the greatest prediction accuracy given the data: Data $\longrightarrow$ Prediction.
- Be careful with how predictions translate to decisions.

Thank you for listening !

Daniel Montero Rivas