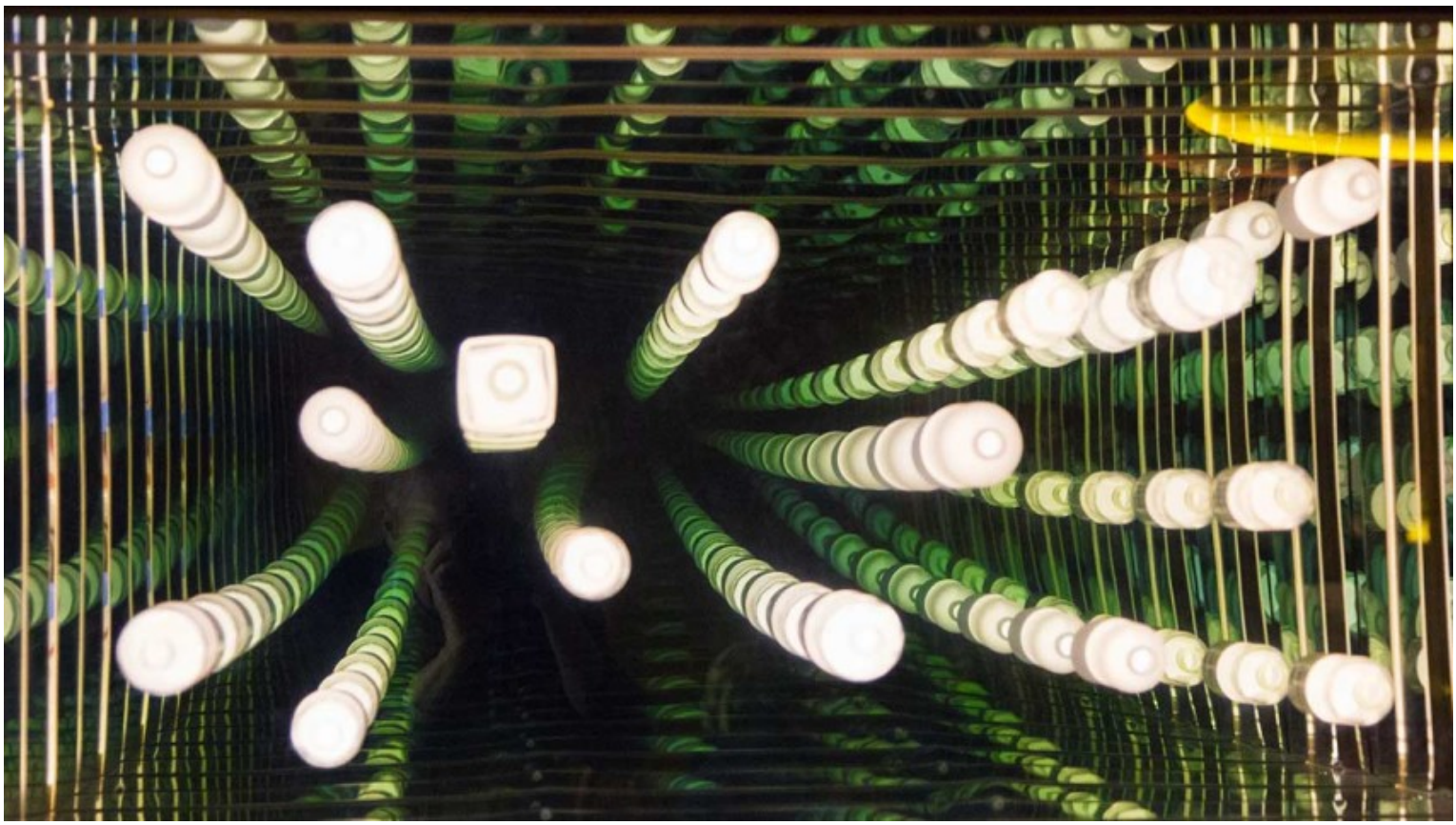
Harvard Business Review

ANALYTICS

A Guide to Solving Social Problems with Machine Learning

by Jon Kleinberg, Jens Ludwig, and Sendhil Mullainathan

DECEMBER 08, 2016



It's Sunday night. You're the deputy mayor of a big city. You sit down to watch a movie and ask Netflix for help. ("Will I like Birdemic? Ishtar? Zoolander 2?") The Netflix recommendation algorithm predicts what movie you'd like by mining data on millions of previous movie-watchers using sophisticated machine learning tools. And then the next day you go to work and every one of your agencies will make hiring decisions with little idea of which candidates would be good workers; community college students will be largely left to their own devices to decide which courses are too

hard or too easy for them; and your social service system will implement a reactive rather than preventive approach to homelessness because they don't believe it's possible to forecast which families will wind up on the streets.

You'd love to move your city's use of predictive analytics into the 21st century, or at least into the 20th century. But how? You just hired a pair of 24-year-old computer programmers to run your data science team. They're great with data. But should they be the ones to decide which problems are amenable to these tools? Or to decide what success looks like? You're also not reassured by the vendors the city interacts with. They're always trying to up-sell you the very latest predictive tool. Decisions about how these tools are used seem too important for you to outsource, but raise a host of new issues that are difficult to understand.

This mix of enthusiasm and trepidation over the potential social impact of machine learning is not unique to local government or even to government: non-profits and social entrepreneurs share it as well. The enthusiasm is well-placed. For the right type of problem, there are enormous gains to be made from using these tools. But so is the trepidation: as with all new "products," there is potential for misuse. How can we maximize the benefits while minimizing the harm?

In applying these tools the last few years, we have focused on exactly this question. We have learned that some of the most important challenges fall within the cracks between the discipline that builds algorithms (computer science) and the disciplines that typically work on solving policy problems (such as economics and statistics). As a result, few of these key challenges are even on anyone's radar screen. The good news is that many of these challenges, once recognized, are fairly straightforward to solve.

We have distilled what we have learned into a "buyer's guide." It is aimed at anyone who wants to use data science to create social good, but is unsure how to proceed.

How machine learning can improve public policy

First things first: There is always a *new* "new thing." Especially in the social sector. Are these machine learning tools really worth paying attention to?

Yes. That's what we've concluded from our own proof-of-concept project, applying machine learning to a dataset of over one million bond court cases (in joint work with Himabindu Lakkaraju and Jure Leskovec of Stanford University). Shortly after arrest, a judge has to decide: will the defendant await their legal fate at home? Or must they wait in jail? This is no small question. A typical jail stay is between two and three months. In making this life-changing decision, by law, the judge has to make a prediction: if released, will the defendant return for their court appearance, or will they skip court? And will they potentially commit further crimes?

We find that there is considerable room to improve on judges' predictions. Our estimates show that if we made pre-trial release decisions using our algorithm's predictions of risk instead of relying on judge intuition, we could reduce crimes committed by released defendants by up to 25% without having to jail any additional people. Or, without increasing the crime rate at all, we could jail up to 42% fewer people. With 12 million people arrested every year in the U.S., this type of tool could let us reduce jail populations by up to several hundred thousand people. And this sort of intervention is relatively cheap. Compared to investing millions (or billions) of dollars into more social programs or police, the cost of statistically analyzing administrative datasets that already exist is next-to-nothing. Plus, unlike many other proposals to improve society, machine learning tools are easily scaled.

By now, policymakers are used to hearing claims like this in sales pitches, and they should appropriately raise some skepticism. One reason it's hard to be a good buyer of machine learning solutions is that there are so many overstated claims. It's not that people are intentionally misstating the results from their algorithms. In fact, applying a known machine learning algorithm to a dataset is often the most straightforward part of these projects. The part that's much more difficult, and the reason we struggled with our own bail project for several years, is accurately evaluating the potential impact of any new algorithm on policy outcomes. We hope the rest of this article, which draws on our own experience applying machine learning to policy problems, will help you better evaluate these sales pitches and make you a critical buyer as well.

Look for policy problems that hinge on prediction

Our bail experience suggests that thoughtful application of machine learning to policy *can* create very large gains. But sometimes these tools are sold like snake oil, as if they can solve *every* problem.

Machine learning excels at predicting things. It can inform decisions that hinge on a prediction, and where the thing to be predicted is clear and measurable.

For Netflix, the decision is what movie to watch. Netflix mines data on large numbers of users to try to figure out which people have prior viewing histories that are similar to yours, and then it recommends to you movies that these people have liked. For our application to pre-trial bail decisions, the algorithm tries to find past defendants who are like the one currently in court, and then uses the crime rates of these similar defendants as the basis for its prediction.

If a decision is being made that already depends on a prediction, why *not* help inform this decision with more accurate predictions? The law *already* requires bond court judges to make pre-trial release decisions based on their predictions of defendant risk. Decades of behavioral economics and social psychology teach us that people will have trouble making accurate predictions about this risk - because it requires things we're not always good at, like thinking probabilistically, making attributions, and drawing inferences. The algorithm makes the same predictions judges are already making, but better.

But many social-sector decisions do not hinge on a prediction. Sometimes we are asking whether some new policy or program works - that is, questions that hinge on understanding the causal effect of something on the world. The way to answer those questions is not through machine learning prediction methods. We instead need tools for causation, like randomized experiments. In addition, just because something is predictable, that doesn't mean we are comfortable having our decision depend on that prediction. For example we might reasonably be uncomfortable denying welfare to someone who was eligible at the time they applied just because we predict they have a high likelihood to fail to abide by the program's job-search requirements or fail a drug test in the future.

Make sure you're comfortable with the outcome you're predicting

Algorithms are most helpful when applied to problems where there is not only a large history of past cases to learn from but also a clear outcome that can be measured, since measuring the outcome concretely is a necessary prerequisite to predicting. But a prediction algorithm, on its own, will focus relentlessly on predicting the outcome you provide as accurately as possible at the expense of everything else. This creates a danger: if you care about *other* outcomes too, they will be ignored. So even if the algorithm does well on the outcome you told it to focus on, it may do worse on the other outcomes you care about but didn't tell it to predict.

This concern came up repeatedly in our own work on bail decisions. We trained our algorithms to predict the overall crime rate for the defendents eligible for bail. Such an algorithm treats every crime as equal. But what if judges (not unreasonably) put disproportionate weight on whether a defendant engages in a very serious violent crime like murder, rape, or robbery? It might *look* like the algorithm's predictions leads to "better outcomes" when we look at overall rates of crime. But the algorithm's release rule might actually be doing worse than the judges with respect to serious violent crimes specifically. The possibility of this happening doesn't mean algorithms can't still be useful. In bail, it turns out that different forms of crime are correlated enough so that an algorithm trained on just one type of crime winds up out-predicting judges on almost every measure of criminality we could construct, including violent crime. The point is that the outcome you select for your algorithm will define it. So you need to think carefully about what that outcome is and what else it might be leaving out.

Check for bias

Another serious example of this principle is the role of race in algorithms. There is the possibility that any new system for making predictions and decisions might exacerbate racial disparities, especially in policy domains like criminal justice. Caution is merited: the underlying data used to train an algorithm may be biased, reflecting a history of discrimination. And data scientists may sometimes inadvertently report misleading performance measures for their algorithms. We should take seriously the concern about whether algorithms might perpetuate disadvantage, no matter what the other benefits.

Ultimately, though, this is an empirical question. In our bail project, we found that the algorithm can actually *reduce* race disparities in the jail population. In other words, we can reduce crime, jail populations *and* racial bias - all at the same time - with the help of algorithms.

This is not some lucky happenstance. An appropriate first benchmark for evaluating the effect of using algorithms is the existing system - the predictions and decisions already being made by humans. In the case of bail, we know from decades of research that those human predictions can be biased. Algorithms have a form of neutrality that the human mind struggles to obtain, at least within their narrow area of focus. It is entirely possible—as we saw—for algorithms to serve as a force for equity. We ought to pair our caution with hope.

The lesson here is that if the ultimate outcome you care about is hard to measure, or involves a hard-to-define combination of outcomes, then the problem is probably not a good fit for machine learning. Consider a problem that *looks* like bail: Sentencing. Like bail, sentencing of people who have been found guilty depends partly on recidivism risk. But sentencing also depends on things like society's sense of retribution, mercy, and redemption, which cannot be directly measured. We intentionally focused our work on bail rather than sentencing because it represents a point in the criminal justice system where the law explicitly asks narrowly for a prediction. Even if there is a measurable single outcome, you'll want to think about the other important factors that aren't encapsulated in that outcome - like we did with race in the case of bail - and work with your data scientists to create a plan to test your algorithm for potential bias along those dimensions.

Verify your algorithm in an experiment on data it hasn't seen

Once we have selected the right outcome, a final potential pitfall stems from how we measure success. For machine learning to be useful for policy, it must accurately predict "out-of-sample." That means it should be trained on one set of data, then tested on a dataset it hasn't seen before. So when you give data to a vendor to build a tool, withhold a subset of it. Then when the vendor comes back with a finished algorithm, you can perform an independent test using your "hold out" sample.

An even more fundamental problem is that current approaches in the field typically focus on performance measures that, for many applications, are inherently flawed. Current practice is to report how well one's algorithm predicts only among those cases where we can observe the

outcome. In the bail application this means our algorithm can only use data on those defendants who were released by the judges, because we only have a *label* providing the correct answer to whether the defendant commits a crime or not for defendants judges chose to release. What about defendants that judges chose not to release? The available data cannot tell us whether they would have reoffended or not.

This makes it hard to evaluate whether any new machine learning tool can actually improve outcomes relative to the existing decision-making system — in this case, judges. If some new machine learning-based release rule wants to release someone the judges jailed, we can't observe their "label", so how do we know what would happen if we actually released them?

This is not merely a problem of academic interest. Imagine that judges have access to information about defendants that the algorithm does not, such as whether family members show up at court to support them. To take a simplified, extreme example, suppose the judge is particularly accurate in using this extra information and can apply it to perfectly predict whether young defendants re-offend or not. Therefore the judges release only those young people who are at zero risk for re-offending. The algorithm only gets to see the data for those young people who got released - the ones who never re-offend. Such an algorithm would essentially conclude that the judge is making a serious mistake in jailing so many youthful defendants (since none of the ones in its dataset go on to commit crimes). The algorithm would recommend that we release far more youthful defendants. The algorithm would be wrong. It could inadvertently make the world worse off as a result.

In short, the fact that an algorithm predicts well on the part of the test data where we can observe labels doesn't *necessarily* mean it will make good predictions in the real world. The best way to solve this problem is to do a randomized controlled trial of the sort that is common in medicine. Then we could directly compare whether bail decisions made using machine learning lead to better outcomes than those made on comparable cases using the current system of judicial decision-making. But even before we reach that stage, we need to make sure the tool is promising enough to ethically justify testing it in the field. In our bail case, much of the effort went into finding a "natural experiment" to evaluate the tool.

Our natural experiment built on two insights. First, within jurisdictional boundaries, it's essentially random which judges hear which cases. Second, judges are quite different in how lenient they are. This lets us measure how good judges are at selecting additional defendants to jail. How much crime reduction does a judge with a 70% release rate produce compared to a judge with an 80% release rate? We can also use these data to ask how good an algorithm would be at selecting additional defendants to jail. If we took the caseload of an 80% release rate judge and used our algorithm to pick an additional 10% of defendants to jail, would we be able to achieve a lower crime rate than what the 70% release rate judge gets? That "human versus machine" comparison doesn't get tripped up by missing labels for defendants the judges jailed but the algorithm wants to release, because we are only asking the algorithm to recommend additional detentions (not releases). It's a comparison that relies only on labels we already have in the data, and it confirms that the algorithm's predictions do indeed lead to better outcomes than those of the judges.

It can be misguided, and sometimes outright harmful, to adopt and scale up new predictive tools when they've only been evaluated on cases from historical data with labels, rather than evaluated based on their effect on the key policy decision of interest. Smart users might go so far as to refuse to use any prediction tool that does not take this evaluation challenge more seriously.

Remember there's still a lot we don't know

While machine learning is now widely used in commercial applications, using these tools to solve policy problems is relatively new. There is still a great deal that we don't yet know but will need to figure out moving forward.

Perhaps the most important example of this is how to combine human judgment and algorithmic judgment to make the best possible policy decisions. In the domain of policy, it is hard to imagine moving to a world in which the algorithms actually *make* the decisions; we expect that they will instead be used as decision aids.

For algorithms to add value, we need people to actually use them; that is, to pay attention to them in at least some cases. It is often claimed that in order for people to be willing to use an algorithm, they need to be able to really understand how it works. Maybe. But how many of us know how our cars

work, or our iPhones, or pace-makers? How many of us would trade performance for understandability in our own lives by, say, giving up our current automobile with its mystifying internal combustion engine for Fred Flintstone's car?

The flip side is that policymakers need to know when they should override the algorithm. For people to know when to override, they need to understand their comparative advantage over the algorithm - and vice versa. The algorithm can look at millions of cases from the past and tell us what happens, on average. But often it's only the human who can see the extenuating circumstance in a given case, since it may be based on factors not captured in the data on which the algorithm was trained. As with any new task, people will be bad at this in the beginning. While they should get better over time, there would be great social value in understanding more about how to accelerate this learning curve.

Pair caution with hope

A time traveler going back to the dawn of the 20th century would arrive with dire warnings. One invention was about to do a great deal of harm. It would become one of the biggest causes of death—and for some age groups the biggest cause of death. It would exacerbate inequalities, because those who could afford it would be able to access more jobs and live more comfortably. It would change the face of the planet we live on, affecting the physical landscape, polluting the environment and contributing to climate change.

The time traveler does not want these warnings to create a hasty panic that completely prevents the development of automobile transportation. Instead, she wants these warnings to help people skip ahead a few steps and follow a safer path: to focus on inventions that make cars less dangerous, to build cities that allow for easy public transport, and to focus on low emissions vehicles.

A time traveler from the future talking to us today may arrive with similar warnings about machine learning and encourage a similar approach. She might encourage the spread of machine learning to help solve the most challenging social problems in order to improve the lives of many. She would also remind us to be mindful, and to wear our seatbelts.

Jon Kleinberg is a professor of computer science at Cornell University and the coauthor of the textbooks *Algorithm Design* (with Éva Tardos) and *Networks, Crowds, and Markets* (with David Easley).

Jens Ludwig is the McCormick Foundation Professor of Social Service Administration, Law and Public Policy at the University of Chicago.

Sendhil Mullainathan is a professor of economics at Harvard University and the coauthor (with Eldar Shafir) of *Scarcity: Why Having Too Little Means So Much*.