McKinsey
& Company

Public & Social Sector Practice

When governments turn to AI: Algorithms, trade-offs, and trust

Artificial intelligence can help government agencies solve complex public-sector problems. For those that are new at it, here are five factors that can affect the benefits and risks.

by Anusha Dhasarathy, Sahil Jain, and Naufal Khan

© Gorodenkoff/Getty Images

October 2020

As artificial intelligence (AI) and machine learning gain momentum, an increasing number of government agencies are considering or starting to use them to improve decision making. Additionally, COVID-19 has suddenly put an emphasis on speed. In these uncharted waters, where the tides continue to shift, it's not surprising that analytics, widely recognized for its problem-solving and predictive prowess, has become an essential navigational tool. Some examples of compelling applications include those that identify tax-evasion patterns, sort through infrastructure data to target bridge inspections, or sift through health and social-service data to prioritize cases for child welfare and support, or predicting the spread of infectious diseases. They enable governments to perform more efficiently, both improving outcomes and keeping costs down.

The most pressing aspects of adopting such solutions are generally well known. Algorithms should be accurate and consciously checked for unintended bias.[1] Others are less so. Algorithms must be stable, meaning that small changes to their input don't meaningfully change their output. They should be explainable, especially in the public sector, where myriad stakeholders will review every step.[2] And to ensure successful adoption, public-sector users should pay particular attention to how AI solutions are deployed, given public-sector managers generally have less authority and operational control to compel adoption than private-sector ones. While all these factors are relevant to every public-sector entity, they aren't necessarily relevant in the same way.

Getting the right balance is essential not only to minimize the risks but also to build a proper business case for the investment, and to ensure that taxpayer dollars are well spent. Below, we'll explore each of these five dimensions—accuracy, fairness, explainability, stability, and adoption—as they apply to the public sector.

Accuracy

When it comes to algorithms, public-sector users could measure performance in terms of better decision making. Since there are typically many possible measures and probabilistic outcomes, it's unlikely that an algorithm will forecast every one of them precisely. Users could start with identifying which ones are most likely to lead to the best decisions for the situation. We recommend focusing on two or three measures that truly matter for the specific use case. Consider the following examples:

— *Prioritizing investments in road work by analyzing traffic-bottlenecks.* When funds are scarce, government officials may prefer to reduce the number of false positives—spending money to repair roads with fewer bottlenecks—relative to false negatives, that is, missing a road that has bottlenecks. Spending money on roads that don't need repairs wastes taxpayer money and, potentially, strains public coffers. In contrast, while missing a road with a bottleneck delays resolution of the issue, it is likely until the next round of assessment and prioritization.

— *Deciding where to focus tax audits.* Tax officials may want to optimize for focusing on only the most likely tax evaders—given the potential consequences of falsely tagging someone as a high risk for evasion.

— *Deciding which students get scholarship money based on probability to graduate.* When the rank order of students determines scaled scholarship amounts, the order in which students rank could matter more than the absolute probabilistic score that the individual student receives from the model—in this instance, the likelihood of graduation. In such cases, school administrators would care most about predicting the correct ranking order of the students than the accuracy of the probabilistic outcome by itself.

[1] See, for instance, Kate Crawford, "The hidden biases in big data," Harvard Business Review, April 1, 2013, hbr.org.
[2] See, for example, Michael Chui, James Manyika, and Mehdi Miremadi, "What AI can and can't do (yet) for your business, McKinsey Quarterly, January 2018, McKinsey.com.

When it comes to algorithms, public-sector users could measure performance in terms of better decision making.

One word of caution: ensure that a clear baseline accuracy for decision making exists before implementing an algorithm, whether based on historical human decisions, rudimentary scoring, or criteria-based approaches that were being used. Knowing when the algorithm performs well and when it does not, relative to the baseline, is helpful both for making a case to use it as well as to establish incentives for continued improvement of the algorithm.

In our experience, machine learning can significantly improve accuracy relative to most traditional decision-making processes or systems. Its value can come from better resource-allocation decisions, such as matching the right types of rehabilitation programs in a corrections facility to the prisoners most likely to benefit from them. But it can also be valuable for improving efficiency, such as helping public-health case workers prioritize the right cases, as well as effectiveness, such as knowing which school programs are most effective at minimizing drop-outs.

Fairness

There are many ways to define a fair algorithm, or "algorithmic fairness."[3] The notion reflects an interest in bias-free decision making or, when protected classes of individuals are involved, in avoiding disparate impact to legally protected classes.[4] There is extensive literature on bias in algorithms and how this could manifest. Common issues include some kinds of bias in the data sets and distortions in the algorithm's analytical technique—or in how humans interpret the data.

A critical first step is to establish what fairness means in the specific context of the use case—that is, what are the protected classes and what are the metrics for fairness. There are a few ways to measure and address fairness, not all of which may be equally effective in each instance:

— *Willful blindness.* One approach that is commonly used is to build a kind of blindness into the algorithm, so that it treats subgroups the same regardless of traditional distinctions between them, such as race, gender, or other socioeconomic factors.

  For example, if a school uses an algorithm to identify students at risk of dropping out, educators could deploy a model that uses gender-masked or gender-neutral records to identify those at the greatest risk. Yet even that kind of approach can be naive if it doesn't account for cross-correlated variables—such as postal codes that could imply race, education level, or gender. Such an approach could lead to unfair outcomes or cause issues with the sample data used to train the model itself. It ends up creating an algorithm that is merely unaware without any consideration to fairness.

— *Demographic or statistical parity.* Another way to address fairness is to ensure statistical parity in the decisions being enabled or in the outcomes—for example, by selecting an equal share of people from both protected and nonprotected groups. One way to achieve this would be to set different thresholds for different groups to ensure parity in the outcomes for each group.

[3] Gal Yona, "A gentle introduction to the discussion on algorithmic fairness," Towards Data Science, October 5, 2017, towardsdatascience.com.
[4] These vary from place to place, but typically include, for example, race, gender, age, religion, and sexual orientation.

An example of the latter would be an algorithm written to apply different credit-score thresholds for different demographic groups, in order to select the same proportion of applicants from each. However, this approach requires someone to constantly verify and modify the thresholds—and often may not account for underlying differences in the subgroups. It is usually effective only when someone cares about a single measure of fairness, in this case, an equal share of loan-approval outcomes across gender types.

— *Predictive equality.* Possibly the most balanced approach to address fairness is to not force it in the decision outcome, but rather in the algorithm's performance (or accuracy) across different groups. (For more, see sidebar "Ensuring fairness.") In this definition, fairness means that the algorithm is not disproportionately better or worse off in how decisions are being made for specific subgroups. That means, for example, that the error rates or prevalence of false positives or false negatives for each group is the same—while accounting for variations in the underlying population. In our loan-applicant example, this means that we may not approve an equal share of loan applicants across genders, but the percent of approved applicants who end up defaulting (that is, the false positives) would be the same across genders. In other words, we are not disproportionately favoring or affecting either gender as we are making the same rate of mistakes or errors in our selection.

Ensuring fairness

Fairness through predictive equality can be achieved through a set of nuanced debiasing practices used in the field of data science. Some of these practices include the following:

— Identify the specific subgroups or protected classes that are relevant.

— Identify the set of metrics that define fairness, and any implicit hierarchy within those, if you have more than one.

— Evaluate the training data set for adequacy across subpopulations or protected classes—and collect more data where needed.

— Identify features such as zip codes, income levels, or other socioeconomic data that are correlated with the protected-class variables or groups—and either remove or adapt them. Advanced methods could use machine learning to identify how biased the model is; as an example, if removing race from a model does not change the outcomes at all, then potentially other variables are strongly cross-correlated.

— Evaluate fairness outcomes for different model types, across different time periods. Consider if specific models for different classes or subgroups may be needed (or thresholds or adjustments may be required).

Fairness through predictive equality can be achieved through a set of nuanced debiasing practices used in the field of data science.

We should note that fairness may come at a cost of lower accuracy. For example, we may find an algorithmic model is highly accurate for a population overall—but not for some subsets of the population where there is less data. In the case of education systems, changes in the demographics of a population could render models of behavior moot, if the models are based on historical data. Put another way, the model might be more accurate for historically dominant groups and less accurate for others.

There can be a trade-off between higher overall accuracy at the cost of poorer, less-fair performance for some and more fairness (by removing certain features) at the cost of reducing overall accuracy. For example, if certain variables in the underlying data, such as postal codes, are correlated with race in certain geographies, then adding postal codes to the data set used by a model to be more accurate could inadvertently introduce racial bias. Hence, in picking the right model, it is important to look at how algorithms score across the five dimensions we have outlined here.

Explainability

Easily explained algorithms can be critical in encouraging the adoption of an AI application, ensuring that stakeholders understand how and why decisions are reached. In our experience, AI and machine learning are most valuable when used to support, and not substitute for, human decision making—and to enable the same humans to understand the rationale behind the algorithm's recommendations. In our experience, just making a real person available to engage with those affected by consequential decisions can make a difference, even if the decision isn't changed. Many public-sector systems are already designed to enable this, such as judicial hearings and public-comment periods around policy decisions. This combination of "human plus machine" can actually often make substantively better decisions than the machine or the human on their own (see sidebar, "Privacy, integrity, and vulnerability").

This is particularly relevant regarding decisions to allocate a scarce resource, such as when an algorithm's output helps select a limited number of applicants for scholarships, grants, or permits. In extreme cases, a black-box AI application—one that isn't or can't be explained—can potentially cause more harm than help. Machines can make errors and reach rigid conclusions, especially in narrow borderline situations. For example, an algorithm might deny a loan for an applicant with a credit score of 728 when the cutoff is 730. People can only correct errors or make exceptions when they understand how the machine makes decisions.

Like fairness, explainability can also lead to difficult trade-offs. Simpler algorithms using rules-based heuristics or decision trees may be easier to explain, but more nuanced and complex algorithms might

Privacy, integrity, and vulnerability

Another aspect to keep in mind when building explainable algorithms is around data privacy, integrity, and vulnerability. For example, can the algorithm be "hacked" to reverse engineer specific data elements that should remain private? Does the data needed to build algorithms invade people's privacy, especially in light of the European Union's General Data Protection Regulation, for example? Is the data sufficiently protected from internal and external threats? Each of these are extremely important considerations to keep in mind when developing and deploying AI or machine-learning solutions.

be more accurate or less biased. The determinative question is whether it's more important that people understand the rationale behind a decision or more important to be accurate.

The answer is contextual. In some countries, for example, various credit-scoring systems[5] can have wide-ranging implications for an individual's ability to get a loan. In such cases, a more explainable algorithm would give applicants an opportunity to improve their input variables, such as avoiding late payments, to influence their final scores over time. In contrast, if an algorithm accurately identifies patients with high risk of cancer, patients are unlikely to care if the algorithm is easily explained.

Organizations can also consider moving to more complex algorithms once the user base becomes more familiar with and trust is built in the more explainable models.

Stability

Over time, the performance of most algorithms grows unstable, primarily because they were developed using data collected in a world before algorithms were used to make decisions. In addition, sometimes macro changes can affect the relevance of the data that models were trained on. For example, models trained on scenarios before the COVID-19 pandemic may not be relevant going forward. Traditional risk-scoring systems or even human decisions face the same obstacles.

Stability is particularly important in the public sector, where many external factors affect decision making. Consider the example of fraud-mitigation models and public benefits. Fraud patterns evolve very quickly. In addition, changes to benefit requirements can meaningfully affect the kind of fraud that governments experience and the data that the machine-learning model was trained on. For example, during the COVID-19 crisis, many US states experienced a substantial spike in identity fraud related to unemployment benefit claims. A data set created prior to COVID-19 would not have seen this trend. Such changes can create shocks to the system that render historical data less capable of predicting the future—and thereby invalidate traditional heuristics or decision-making rules.

To estimate the frequency at which models should be refreshed, users must understand the speed at which algorithmic performance degrades. One way to do this is to test its performance using backward-looking data over different time spans. If the model performs great on test data that lapsed a year ago but not on data that lapsed two years ago, then retraining the model somewhere between a year and two years will likely help avoid degradation. Ideally, organizations would use such information to develop a cadence of regular testing and retraining to continuously update and rebuild their heuristics.

A great machine-learning model, by itself, is not enough. It often needs to be wrapped in an intuitive user-centric experience and embedded into work flows.

[5] Rachel O'Dwyer, "Algorithms are making the same mistakes assessing credit scores that humans did a century ago," Quartz, qz.com.

However, models may also need to be refreshed in the wake of any major changes to an underlying data set. These might be internal changes, such as the implementation of new policies, or external, such as new legislation.

Adoption

When we think of the potential impact of AI, we think of three big pieces of work: developing the model or algorithm, deriving insight from its output, and adopting its output or recommendations. In the end, a great machine-learning model, by itself, is not enough. It often needs to be wrapped in an intuitive user-centric experience and embedded into work flows, with the use of design thinking and with frontline employees to spur adoption.

Machine-learning algorithms are prone to rejection for the same reasons they deliver great results. That is, they can generate accurate but counterintuitive insights due to the large number of variables and data they use. They go against the grain of traditional heuristics. They challenge the ways things have traditionally been done. And they often require people to give up familiar tools and methods.

Thus, it is critical to plan for, and to incorporate approaches to encourage, adoption from day one. These might include bringing target users into a model's development process from the beginning—or at least soliciting frequent reviews and input along the way. It might also include designing a straightforward way to deliver and consume the model's insights. Consider one organization that successfully implemented advanced analytics models. The response to adoption at this organization was positive because end users were excited by the insights but even more excited by the intuitive user interface. The interface consolidated disparate sources of data—including paper sources—into one easy-to-use, front-end solution. Because their work became less tedious, stakeholders were eager to use both the analytics and the tool.

While important, adoption is where typical analytics teams struggle, whether internally in public-sector agencies or in external partnerships with vendors. Proper adoption requires end-to-end expertise, from use-case articulation to model development, tool development (insight delivery), and, ultimately, change management and operational rollout. The need for these cross-functional skills and expertise makes this last mile often the most challenging one.

Sometimes, in the rush toward employing AI, it is easy to ignore the limitations and risks associated with algorithms. The good news is that these limitations can be understood, managed, and mitigated as needed.

Anusha Dhasarathy is a partner in McKinsey's Chicago office, where **Sahil Jain** is an associate partner and **Naufal Khan** is a senior partner.

The authors wish to thank Ben Cheatham, Michael Chui, Nicolaus Henke, Navjot Singh, and Chris Wigley for their contributions to this article.

Designed by McKinsey Global Publishing