MITSloan
Management Review

What Leaders Should Know About Measuring AI Project Value

Most AI/machine learning projects report only on technical metrics that don't tell leaders how much business value could be delivered. To prevent project failures, press for business metrics instead.

Eric Siegel

• Reprint #65334 • sloanreview.mit.edu

What Leaders Should Know About Measuring AI Project Value

Eric Siegel

Most AI/machine learning projects report only on technical metrics that don't tell leaders how much business value could be delivered. To prevent project failures, press for business metrics instead.

Neil Webb/theispot.com

"AI" can mean many things, but for organizations using artificial intelligence to improve existing, large-scale operations, the applicable technology is machine learning (ML), which is a central basis for — and what many people mean by — AI. ML has the potential to improve all kinds of business processes: It generates predictive models that improve targeted marketing, fraud mitigation, financial risk management, logistics, and much more. To differentiate from generative AI, initiatives like these are also sometimes called *predictive AI* or *predictive analytics*. You might expect that the performance of these predictive ML models — how good they are, and how much value they deliver — would be front and center. After all, generating business value is the whole point.

But you would be wrong. When it comes to evaluating a model, most ML projects report on the wrong metrics — and this often kills the project entirely.

In this article, adapted from *The AI Playbook: Mastering the Rare Art of Machine Learning Deployment*, I'll explain the difference between technical and business metrics for benchmarking ML. I'll also show how to report on performance in business terms, using credit card fraud detection as an example.

Why Business Metrics Must Come First

When evaluating ML models, data scientists focus almost entirely on technical metrics like precision, recall, and *lift*, a kind of predictive multiplier (in other words, how many times better than guessing does the model predict?). But these metrics are critically insufficient. They tell us the *relative performance* of a predictive model — in comparison to a baseline such as random guessing — but provide no

direct reading on the absolute *business value* of a model. Even the most common, go-to metric, *accuracy*, falls into this category. (Also, it's usually impertinent and often misleading.)

Instead, the focus should be on business metrics — such as revenue, profit, savings, and number of customers acquired. These straightforward, salient metrics gauge the fundamental notions of success. They relate directly to business objectives and reveal the true value of the imperfect predictions ML delivers. They're core to building a much-needed bridge between business and data science teams.

Unfortunately, data scientists routinely omit business metrics from reports and discussions, despite their importance. Instead, technical metrics dominate the ML practice — both in terms of technical execution and in reporting results to stakeholders. Technical metrics are pretty much the only kind of metric that most data scientists are trained to work with and most ML tools are programmed to handle.

Data scientists know better but generally don't abide — in good part because ML software tools generally serve up only technical metrics. According to the 2023 Rexer Analytics Data Science Survey, data scientists rank business KPIs, such as ROI and revenue, as the most important metrics yet say technical metrics are the most commonly measured.

The AI industry has this backward. As Katie Malone astutely put it in *Harvard Data Science Review*, "The quantities that data scientists are trained to optimize, the metrics they use to gauge progress on their data science models, are fundamentally useless to and disconnected from business stakeholders without heavy translation."

Fixating on technical metrics doesn't just compromise an ML project's value: Often, this entrenched habit utterly sabotages the project, for two big reasons. First, during model development, the data scientist is benchmarking on metrics that don't directly measure business value — so their model is not maximizing value. If you're not measuring value, you're not pursuing value.

Second, when the data scientist delivers an ML model for deployment, the business stakeholders lack visibility into the potential business value the model could realize. They have no meaningful read on how good the model is. When business leaders ask for straightforward business metrics like profit or ROI, the data scientist is typically ill-equipped to report on these measures. So without a basis for making an informed decision, they typically make the tough choice between authorizing deployment on a leap of faith or, in essence, canceling the project. This latter case of wet feet dominates: Most new ML projects fail to deploy. An IBM Institute for Business Value study found that ROI on enterprisewide AI initiatives averaged just 5.9% as of late 2021 (and that's lower than the cost of capital, meaning you'd be better off just investing the money in the market). Getting the metrics discussion right by including business metrics is central to overcoming the great challenges to launching ML projects.

How to Move From Technical Metrics to Business Metrics

Let's dig a little deeper to see what it takes to measure — and therefore pursue — business value. We often can span a mathematical bridge from technical performance to business performance by incorporating the price you pay when a model predicts wrongly. You incur a misclassification cost for two different kinds of prediction error:

False positive (FP): When a predictive model says "positive" but is wrong. It's a negative case that's been wrongly flagged by the model as positive. Also known as a *false alarm* or a *false flag*.

False negative (FN): When a predictive model says "negative" but is wrong. It's a positive case that's been wrongly flagged by the model as negative.

Accuracy is a blunt instrument. It's one thing to know a model is wrong, say, 12% of the time. That's the same as saying it is correct 88% of the time; that is, it's 88% accurate.

But it's another thing, a much more helpful thing, to separately break down how often it's wrong for positive cases and how often it's wrong for negative cases. Accuracy doesn't do that.

An Example: Fraud Detection Costs

How can you assign a business cost to FP and FN misclassifications? It comes down to how much each kind of error matters. For almost all projects, an FP error matters a different amount than an FN.

Take fraud detection. When your bank's model wrongly blocks your legitimate credit card transaction as if it were fraudulent, you're inconvenienced. That's an FP. This could cost the bank $100 on average, given that you might turn to another card in your wallet — not only for the current purchase but also in general.

The other kind of error is worse. When the bank's model wrongly allows a fraudulent credit card charge to go through, that could cost the bank $500 on average, as the criminal gets away with the contraband. That's an FN.

These FN costs are no small matter. Global payment card fraud losses have surpassed $28 billion annually. The cardholder or an eagle-eyed auditor might notice the bogus charge later, but for card purchases, if it isn't caught by a model on the fly, it's in the wind. In the United States, the bank is usually liable for this loss.

By determining the two misclassification costs, we establish a cost-benefit analysis not only for the entire project but also for each individual decision about whether to hold or authorize a transaction. Then we can add up those individual costs to calculate a KPI for the overall project: cost savings.

Sacrificing a Little Accuracy Makes Sense

With no fraud detection model deployed, a medium-sized regional bank could be losing $50 million per year. Consider a bank that has issued 100,000 credit cards, and each card sees an average of 1,000 transactions per year, with 1 in 1,000 being fraudulent. To summarize:

- Annual transactions: 100 million
- Percentage that are fraudulent: 0.1%
- Annual fraudulent transactions: 100,000
- Cost per fraudulent transaction: $500 (the FN cost)
- Annual loss from fraud: 100,000 × $500 = $50 million

It looks like crime does pay after all. But before you quit your day job to join the ranks of fraudsters, know that fraud detection can improve the situation.

In fact, in the example above, you could save $16 million: The key is developing a fraud detection model that provides an advantageous trade-off between FPs (less costly) and FNs (more costly). For the detailed math, see the sidebar "A Fraud Detection Model's Value: The Math, Explained." As it illustrates, calculating the business value is only a matter of arithmetic.

It turns out that, in general, a fraud detection model can only deliver a cost savings by sacrificing a little accuracy. For example, the model described in the sidebar is 99.8% accurate — slightly lower than the 99.9% accuracy of a "dumb" model that simply assumes every transaction is legitimate (and therefore takes no action to prevent fraud). In this case, a less-accurate model is actually better.

To understand why, just revisit accuracy's fatal flaw: It doesn't distinguish between different kinds of errors, treating FPs and FNs as equally bad. Since it doesn't account for different misclassification *costs*, accuracy oversimplifies for all but very rare ML projects where the costs don't differ. For most projects, accuracy is a red herring.

Beyond delivering business value, fraud detection pursues a societal objective: It fights crime. In the example shown, fraud detection blocks more than half of the attempted fraudulent transactions. In so doing, it meets the expectations of consumers. Although people sometimes bristle at having their behavior predicted by models — electronically pigeonholed to receive bad ads, for example — when it comes to using payment cards, many consumers welcome prediction, gladly withstanding the occasional blocked transaction. Conversely, many consumers want to avoid being charged for a purchase they never made. So

the typical cardholder has an expectation of fraud detection, though they might not be cognizant of it.

Better Decisions, Better Value

By reporting on a fraud detection model's absolute business value — in our example, a cost savings of $16 million — rather than only its relative performance in terms of lift or any other technical metric, business stakeholders are provided with something real to evaluate. They can make an informed decision about whether, how, and when to authorize the ML model's deployment.

It's time for a change: Data scientists must report on business metrics as part of their regular practice. Although bridging the divide from technical to business metrics is rare today, it's a problem that's readily surmountable. You will need leaders and data scientists who are willing to rethink the way they discuss and report on ML projects — and are rewarded for doing so. This way, you will jointly navigate to ML project success.

About the Author

Eric Siegel is a consultant and a former Columbia University and University of Virginia Darden School of Business professor. He is the founder of Machine Learning Week and author of *The AI Playbook: Mastering the Rare Art of Machine Learning Deployment* (MIT Press, 2024) and *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die* (Wiley, 2013).

A Fraud Detection Model's Value: The Math, Explained

Let's dive into the math behind the fraud detection example — knowing that, although this monumental step is only a matter of arithmetic, it's some very particular arithmetic.

If the bank is willing to treat 2 of every 1,000 attempted transactions (0.2%) as potentially fraudulent — deciding in real time to hold the transaction and possibly inconveniencing the customer — then the onus is on a fraud detection model to flag which transactions should be held.

Let's assume the model attains a lift of 300 — which means that, among the targeted 0.2%, fraud occurs 300 times more often than average. This technical metric is telling us that the model has done a relatively good job identifying certain transactions that are more likely to be fraudulent. Three hundred is a higher lift than you could hope for when pursuing many other ML use cases.

But lift is always relative to the size of the targeted group. In this case, we care about the lift only among the very small sliver of transactions scored as most likely to be fraudulent — the top 0.2% that will be blocked. We won't block any attempted transactions other than those, so that sliver is all that counts. Given that it's such a small portion, a high lift is feasible: A model can potentially distinguish more risky transactions well enough so that the sliver includes a relatively high concentration of positive cases.

First, we need to calculate how many errors occur, broken into FPs and FNs — how often the model wrongly blocks a legitimate transaction and how often it lets a fraudulent transaction slip by. Here's the breakdown:

- Transactions blocked: 200,000 (2 per 1,000)
- Percentage blocked that are fraudulent: 30% (lift × overall fraud rate = 300 × 0.1%)
- Fraudulent transactions blocked: 60,000 (30% × 200,000)
- FPs (legitimate transactions blocked): 140,000 (200,000 − 60,000)
- FNs (fraudulent transactions allowed): 40,000 (100,000 − 60,000)

This model is often wrong but extremely valuable. When it blocks a transaction, it's usually wrong — only 30% of the blocked transactions are fraud. This isn't unusual. Since fraud is so infrequent, it would be very difficult to correctly detect some cases without also incorrectly flagging legitimate transactions even more often. With legitimate transactions — that is, negative cases — so prevalent, even misclassifying a small portion of them means a lot of FPs.

So the best we can hope for from a model is that it provides an advantageous trade-off between FPs (less costly) and FNs (more costly). To calculate the bottom line, we add up the costs. We've already established the cost for individual errors:

- Cost of an FP: $100 (inconvenience to a customer)
- Cost of an FN: $500 (fraudster gets away with it)

So we need only multiply these costs by how often they're incurred:

- Aggregate FP cost: $14 million (140,000 at $100 each)
- Aggregate FN cost: $20 million (40,000 at $500 each)
- Total cost with fraud detection: $34 million

We've cut fraud losses by $30 million (from $50 million to $20 million) but introduced $14 million in new costs resulting from FPs. Clearly, this is a worthy trade-off.

- Overall cost savings: $16 million ($50 million to $34 million)

MITSloan Management Review

PDFs ■ Reprints ■ Permission to Copy ■ Back Issues