WHITE PAPER

Optimizing for the *wrong number.*

A model that wins on the metric and loses on the outcome is a common and expensive result. Choosing what to optimize is a business decision, not a technical one.

Riverton & Associates, Inc. · 17 March 2026

Executive summary

The most expensive failure in applied machine learning is not a model that performs badly. It is a model that performs well — against a target nobody should have chosen.

This failure is dangerous precisely because it does not look like failure. The metric improves. The validation curve behaves. The team ships on schedule and reports a win. Six months later the business number has not moved, and nobody can explain why, because every technical indicator is still green.

Choosing the objective is the highest-leverage decision in the whole project, and it is the one most often made by default — inherited from a tutorial, a benchmark, or whatever the library optimises out of the box.

The failure looks like success

Consider the shape of it. A team is asked to reduce customer churn. They build a churn classifier, tune it hard, and reach 0.91 AUC — a genuinely good model. It goes into production. Retention runs campaigns against its output. A year later, churn is unchanged.

> Nothing in the modelling was wrong. The model predicts churn accurately. The problem is that accurate churn prediction and reduced churn are different objectives, and the project optimised the first while being paid for the second.

Worse, this failure resists debugging. Nobody investigates a model that is hitting its numbers. The instinct is to tune further — more features, a bigger model, better calibration — which improves the metric again and moves the outcome not at all. Teams can spend a year on that treadmill.

Where the gap comes from

Every metric is a proxy for an outcome. Proxies diverge, and they diverge in specific, recognisable ways.

- **Accuracy on imbalanced problems.** A fraud model that flags nothing is 99.9% accurate. Well known, and still ships regularly.
- **Optimising globally when you act locally.** AUC measures ranking across the whole distribution. If the fraud team can investigate two hundred cases a week, the only region that matters is the top two hundred. A model with better AUC and worse precision in that slice is worse for the business and better on the report.
- **Treating all errors as equal.** Minimising RMSE weights a $400 error on a small account exactly like a $400 error on the account that carries the quarter. Squared error is a mathematical convenience, not a statement about your P&L.
- **The wrong horizon.** Predicting next month's churn when the retention intervention takes a quarter means every correct prediction arrives too late to act on.
- **Predicting the un-actionable.** A precision-tuned churn model concentrates on the customers it is most confident about — usually the ones already gone, having cancelled the auto-renewal and stopped logging in. The customers you can still save sit in the uncertain band the model was tuned to avoid.

- **Proxy metrics with a life of their own.** Click-through optimised hard enough produces clickbait. Engagement optimised hard enough produces outrage. The metric goes up, the business degrades, and the model is working exactly as specified.

The cost matrix nobody wrote down

Underneath most of these is one omission: nobody stated what the errors cost. False positives and false negatives are almost never equally expensive, and the ratio between them determines the correct operating point. That ratio is a business fact. It cannot be derived from the data, and it is not the modeller's to invent.

CHURN EXAMPLE	COST	WHAT IT IS
Retention offer	$200	what a false positive costs
Value of a saved customer	$3,000	what a false negative costs
Ratio	**1 : 15**	—

Illustrative. Ten minutes with the business owner establishes this.

At fifteen to one, you should be willing to accept a great many false positives to avoid one false negative. Yet the default instinct — and the default threshold in most libraries — is to balance the two, and a team asked to "improve precision" will tune away from exactly the customers worth saving. Establishing this ratio changes the threshold, the metric, and sometimes the entire choice of model. It is routinely skipped.

The number has to attach to a decision

A prediction that changes nothing has no value, however accurate it is. Before choosing a metric, three questions should have concrete answers:

- **What action does this change?** If the answer is "it gives us visibility", there is no decision, and therefore nothing to optimise toward.
- **Who takes that action, and what is their capacity?** This sets the operating point. A team that can work two hundred cases a week defines a top-200 problem, not a whole-distribution problem.
- **How long does the action take to work?** This sets the prediction horizon. A forecast that arrives after the last moment it could have been acted on is a report, not a model.

If those three cannot be answered, the metric cannot be chosen — and the honest response is to stop and answer them, not to pick a default and proceed.

Goodhart's law arrives faster than it used to

When a measure becomes a target, it ceases to be a good measure. Not a new observation, but two things have changed.

First, optimisers are much stronger than they were. A modern model asked to maximise a proxy will find the gap between that proxy and the intended outcome, and it will find it efficiently. Capability makes this worse, not better — a weak model optimising the wrong objective fails harmlessly; a strong one succeeds at the wrong thing.

Second, the loop is faster. Systems retrain on data their own decisions generated, so a proxy mismatch compounds instead of staying constant.

The practical consequence: the more capable the model, the more precisely the objective needs to be stated. Vague targets were survivable when nothing could pursue them effectively.

How to choose the number

- **Start from the decision, not the data.** Write the sentence: *when this number arrives, [who] will do [what] differently.* If it cannot be written, stop here.
- **Write the cost matrix in currency.** With the business owner in the room. Both error types, in money, even approximately — an order of magnitude is enough to move the threshold correctly.

- **Fix the operating point before training.** "We will act on the top 200 per week." That sentence determines which metric is meaningful.
- **Choose the metric that matches it.** If you act on a top slice, measure the top slice. If errors are asymmetric, use a cost-weighted loss. If large accounts matter more, weight by value rather than by count.
- **Add a guardrail.** One metric that must not get worse, chosen to catch the proxy going feral. Optimising engagement, guard on complaints. Optimising recall, guard on the review queue.
- **Agree the counterfactual before you start.** What happens without the model? Without a baseline, any result can be presented as a success.
- **Measure the outcome, not only the metric.** Hold out a control group and report the business number. It is the only measurement that answers the question the project was funded to answer, and the one most often missing.

Who owns the choice

This is the argument in the title, and it is worth stating plainly.

The technical team owns **how** to hit the target: architecture, features, training, validation, deployment. That is their expertise and it should not be second-guessed.

The business owns **what** the target is: which error is more expensive, what capacity exists to act, what horizon is useful, what must not get worse. These are not technical questions. They have no technically correct answer. They are statements about how the organisation makes money and what it is willing to trade.

When the objective is delegated to the technical team — usually by omission rather than decision — it defaults to whatever is conventional. And convention in machine learning is shaped by academic benchmarks, designed for comparability across papers, not for your profit and loss. Accuracy, F1 and AUC are excellent for ranking published results. None of them knows what a false negative costs you.

How to tell it has happened to you

- Model metrics are good and the business metric is flat.
- The output exists and nobody uses it.
- The team is tuning the metric rather than questioning it.
- The metric was chosen before anyone described the decision it supports.
- Nobody can state the cost of a false positive in currency.
- There is no control group, so no one can say what would have happened anyway.

Any two of these together are enough to justify stopping and revisiting the objective. That conversation is uncomfortable — it implies the last several months optimised the wrong thing — and it is far cheaper than another two quarters of tuning.

Conclusion

Most failed machine learning projects do not fail technically. They hit their targets. The target was wrong, the wrongness was invisible because the metric kept improving, and the mistake was made in the first week by someone choosing a default.

Choosing what to optimise is the point where the business decides what it actually wants, in enough detail that a machine can pursue it relentlessly. That is not a task to delegate to whoever happens to be writing the training loop. It belongs to the people who know what the errors cost — and ten minutes of their time, before the modelling starts, is worth more than any amount of tuning afterwards.

Riverton & Associates researches, designs, develops and implements production software and AI systems, starting with what the system is actually for.

`Riverton & Associates, Inc. · Austin, Texas · riverton-assoc.com`