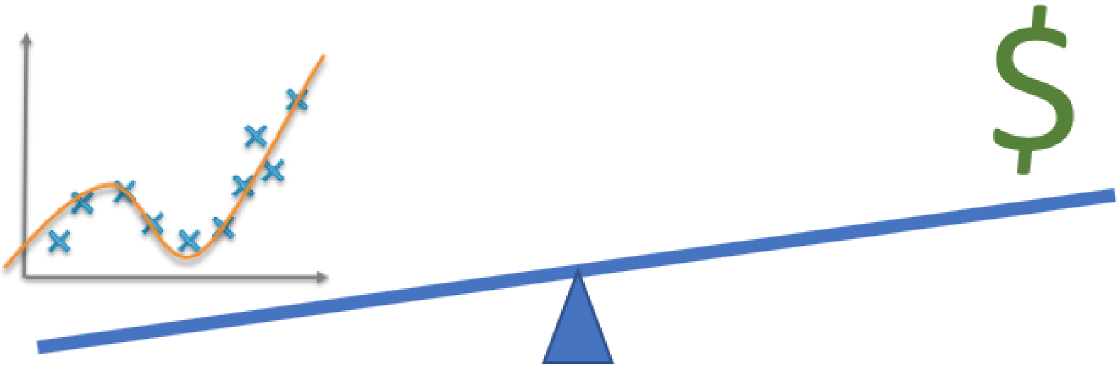
Machine Learning: Balancing model performance with business goals

Stacey Ronaghan
Oct 6, 2018 · 7 min read



This post is designed to give some guidance for evaluating the use of machine learning to solve your business problem.

As a data scientist, I am highly motivated to find the “best” model — how close to perfect can I get my predictions to be? However, more often than not, the incremental gain I’m straining for is not necessary. The success criteria I prioritize isn’t always the same quantifier the business is optimizing for.

For example, if I tell you a client of mine implemented a model with 64.2% accuracy you might be appalled. However, they did not consider the investment in time to improve the model to be necessary. As it was, it helped replaced a predominately manual (and undesirable) work flow that took multiple weeks. The new solution took only couple of days and freed up the team to do more of the challenging tasks they enjoy.

This post will discuss business considerations for evaluating machine learning models. In addition, it provides examples for regression problems (predicting numerical values), classification problems (predicting the class of an item), and recommendations.

The Baseline

The baseline is a metric indicating how successful you are solving the problem today. The goal of using any new solution — whether with machine learning or not — should be to improve on this.

Let's consider the situation of the client I mentioned above…

Example: Currently it takes X hours for employees to do this task manually, costing $Y. They tend to be correct Z% of the time but find the task frustrating.

We have three quantifiable metrics: time, cost and accuracy. There is also the qualitative point indicating the current solution is undesirable to the employees; in some scenarios you may even wish to capture a quantifiable metric for this too.

Goal: Create a solution to remove this task from the employees so they can do work they prefer, improving employee satisfaction whilst saving time and consequently money.

As mentioned above, the business decided that model accuracy was less important than removing unwanted workload and saving time.

So to understand if a model is fit for production, you need to consider and balance multiple metrics. The main three I consider are:

> ***1. Performance***
>
> *This indicates how good the solution is at predicting the correct outcome.*
>
> *The metric itself varies depending on the type of problem. Whichever is chosen for the machine learning model should also be used to calculate the performance baseline.*
>
> ***2. Time***
>
> *This is the duration it takes to complete the task.*

> *For the baseline, this is how long it takes without a machine learning model; whether that is with an alternative software solution or manually.*
>
> ***3. Money***
>
> *This is the monetary impact of the task.*
>
> *For the baseline, this could be related to how much it costs to complete the task or how many sales are made with the current solution.*

As a data scientist I often get very fixated on performance as this is what I have control over. However, in order to get my model used in production, it is important that these other quantifiers are evaluated and communicated. Stakeholders can then make an informed decision on whether to move forward with what I have built.

Example Scenarios

Regression — Predicting House Prices

Let's assume we own a real estate agency; the company has a lot of stock and wants to explore whether machine learning can aid the process of deciding the asking price of each house.

Currently, a human will read the documentation on the property and make an intelligent decision on the value of the house based upon other similar houses recently sold in that area. They will then decide on the offer price based upon their experiences.

We decide to focus on building a model that predicts the value of the house. The model's prediction can then be used by the agent to decide an appropriate offer price.

We are expecting the model to find trends and patterns in the features that relate to price. However, we still appreciate that it may miss nuances that a human can capture — for example, the condition of the property — which is why in this scenario we want them to make the final decision.

Performance

Baseline — How close to the selling price have the previous predictions by agents been?

Metrics — Mean Absolute Error (MAE), Root Mean Square Error (RMSE)

We should calculate the current performance values — how well an agent on average makes a good prediction — and compare against our best model's performance.

Time

Baseline — how long does it take for an agent to make the prediction?

If this is currently a time consuming task for our agents, and using a machine learning model makes it significantly quicker, this alone can justify moving forward with our model.

Money

Baseline — How much does it cost the company to have the agent manually price the house? How much does it cost to be wrong?

Ultimately the agency wants to make the most money. If it's going to cost the company money investing in a machine learning model, they need to understand where they will save money or where they could potentially earn it.

Classification — Predicting Fraudulent Banking Activities

Let's assume we are a community bank and want to explore ways to better prevent fraudulent activities.

Currently, we have simple rules that flag up "suspicious" transactions, such as: above a particular threshold amount, or purchases made out-of-state. For each flagged transaction, an employee reviews the account owners profile and prior transactions to better understand if this seems out-of-character. They then use their best judgement to either allow the transaction or take appropriate actions if they consider it to be fraud.

We wish to build a predictive model to better identify fraud. The model should reduce the number of times our employees review transactions that are not fraudulent whilst ensuring we capture those that are.

As with the prior example, we still need to decide how to use the model in production. One option is to keep the employees in the loop, having them still evaluate the profiles but ensure they have less to review. Alternatively, we trust the model to move straight to the next step in the process: calling the customer to verify the transaction was theirs. This is a decision that will need to be made by the business but the following metrics can help guide that too.

Performance

Baseline — What percentage of transactions flagged as suspicious were in fact fraud? How many of the transactions let through were actually fraudulent?

Metrics — Accuracy, Precision, Recall, F1-score

Understanding how well fraudulent activities are identified today allows us to compare with any model built. The business should evaluate the metrics available and decide which is most important to them — accuracy is the most intuitive but not necessarily the best suited for each problem.

Time

Baseline — What is the time between a transaction taking place and when it is identified by the bank agent as fraud? How long does it take for an agent to confirm with a customer whether a transaction was valid or not?

There may be an opportunity to use a model to minimize the agent's time handling this task. However, delaying the release of funds may also have consequences.

Money

Baseline — How much does it cost the bank to let a fraudulent transaction go through? Conversely, how much money does it cost to follow up with a customer to confirm their transactions?

Fraud obviously costs the business money but it isn't feasible to prevent releasing funds until each transaction has been clarified with the customer. Understanding these associated costs can allow you to decide the appropriate business decision.

Recommendations — Making book recommendations

For this scenario, we own an online book store. We have many repeat customers who log in to the webpage and make purchases. We have historical data of who has purchased what, as well as ratings users have provided for their purchases. We wish to use information to better make recommendations.

Currently we calculate the trending books for each genre: which are the best selling over the past 30 days. The top 20 are displayed as recommendations to the user.

Rather than make the same predictions for every customer purely based upon top sales, we wish to personalize based upon their prior purchases and interests.

As we have historical purchasing and reviews, we have both explicit and implicit data. The ratings are explicit data: the customer is telling us how much like like or dislike an item. Meanwhile, the purchase data is implicit: buying the item implies they like the product but this might not actually hold true.

Performance

Baseline — On average, how many of the recommended books does a customer buy? What was the average rating for books that were recommended?

Metrics (implicit, purchases) — Recall, Precision, Mean Average Precision (MAP)

Metrics (explicit, ratings) — Mean Absolute Error (MAE), Root Mean Square Error (RMSE)

When are are building the model, we are evaluating it's performance with historical data. We may also wish to run this in production on a subset of our users to evaluate its performance in production, this is A/B testing.

Time

Baseline — How long does it take the current recommendation model to display the recommendations?

The recommendation process is already automated but we are looking to replace it. Some recommendation engines may have higher latency (time it takes to return the result) due to the complexity of the algorithm and this should be considered to unsure good user experience.

Money

Baseline — How much money is being made from the current recommendations?

Rather than trying to save money (as in previous examples), here we are trying to make more money. We need to understand want the current financial lift is from the recommendations so we can compare with what we expect the model to achieve.

Machine Learning Data Science Recommendation System Classification

Business Intelligence

About Help Legal