Practical machine learning engineering

Curated guide for building and maintaining solid end-to-end machine learning pipelines

Author: Morales, J.A.
Last update: January 11, 2023

Contents

Introduction 3
Do we really need machine learning? 4
Start measuring before building 4
How to know if you're ready for machine learning 4
Opt for a simple model, with simple features 4
At any point, be clear about what you're testing 4
Don't drop the data, unless noted 4
Heuristics can still be handy 4
Practice 'alerting hygiene' 4
Detect the defects or don't serve 4
How to hear the silent failures 5
References 6

Introduction

Curated guide for building and maintaining solid end-to-end machine learning pipelines.

Do we really need machine learning?

In building a machine learning pipeline, the first question we ask is, “do we really need one?” We want to focus on the problem we’re trying to solve, even if it’s simply wanting to explore a solution or create a proof-of-concept.

Start measuring before building

There are many reasons for doing this. It provides additional information for decisions along the build process. It also provides a pulse on the data – it’ll help capture any changes before, during, and after engineering process.

How to know if you’re ready for machine learning

One sign the project is ready for machine learning is when heuristics begin to get too complex. Simple heuristics work but complicated heuristics can get messy and will hurt more in the long run. At this point, consider machine learning instead. The trifecta you want: unwieldy heuristics + enough data + clear objective = start machine learning.

Opt for a simple model, with simple features

Keeping the first model simple allows you to more efficiently get the infrastructure right. It makes it easier to build up, maintain, and troubleshoot the data and the model.

At any point, be clear about what you’re testing

Test the model, infrastructure, and their integration separately. This helps isolate problems as well as package, prioritize, and plan work.

Don’t drop the data, unless noted

It’s not uncommon to duplicate an existing pipeline as a starting point for a new build. Caveat: the old pipeline might have dropped data that we need in the new pipeline. Make sure it’s noted. This also applies to any filtering and significant data transformations.

Heuristics can still be handy

If we have pre-existing heuristics that already provide good insight or could potentially help us understand better the data better, we can try to use them by turning them into features.

Practice ‘alerting hygiene’

Know the freshness requirements of the system. Get the model out fast – and keep it fresh. This means we want to put actionable alerts and dashboards in place.

Detect the defects or don’t serve

It’s obvious but it’s important to mention. Promote to higher environments only after testing, fixing, and/or noting bugs. Even if it’s a POC project. Yes, release that MVP. But. Don’t serve to the user if you haven’t done your detecting.

How to hear the silent failures

Compared to other system failures, silent errors are tricky to catch. Many mechanisms are in place to catch the loud and noticeable issues. To reduce silent failures however, it's helpful to keep statistics and manually inspect the data on occasion.

…more coming soon.

References

Zinkevich, M. (n.d.) 'Rules of machine learning: Best practices for ML engineering', Google. Available at: https://developers.google.com/machine-learning/guides/rules-of-ml/ (Accessed: 01 April 2019).