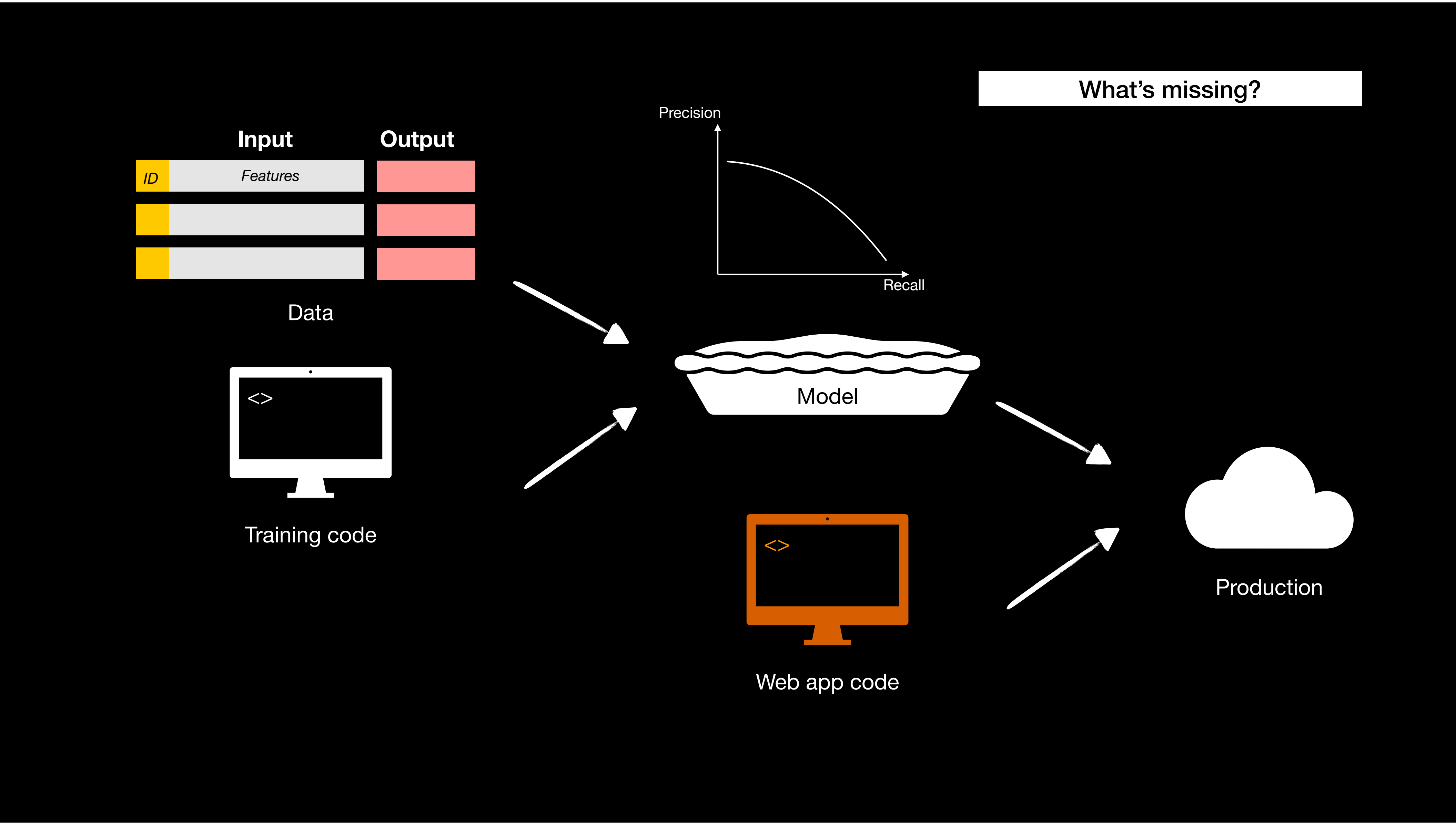
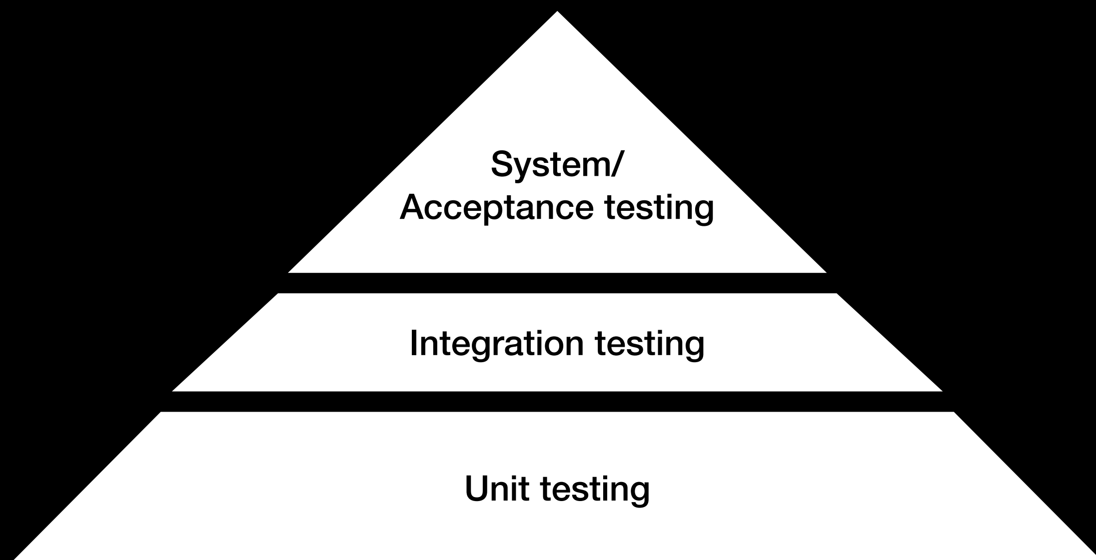
From Machine Learning Models to Systems

Jin Guo

Sept 22, 2020


What’s missing?
Input
Output
ID
Features
Data
Precision
Recall
Model
Training code
Web app code
Production

Testing in Traditional Software Development



What should we test for ML models?

- Algorithmic Correctness (Design and Implementation)

What should we test for ML models?

- Algorithmic Correctness (Design and Implementation)
- Reproducible Training

What should we test for ML models?

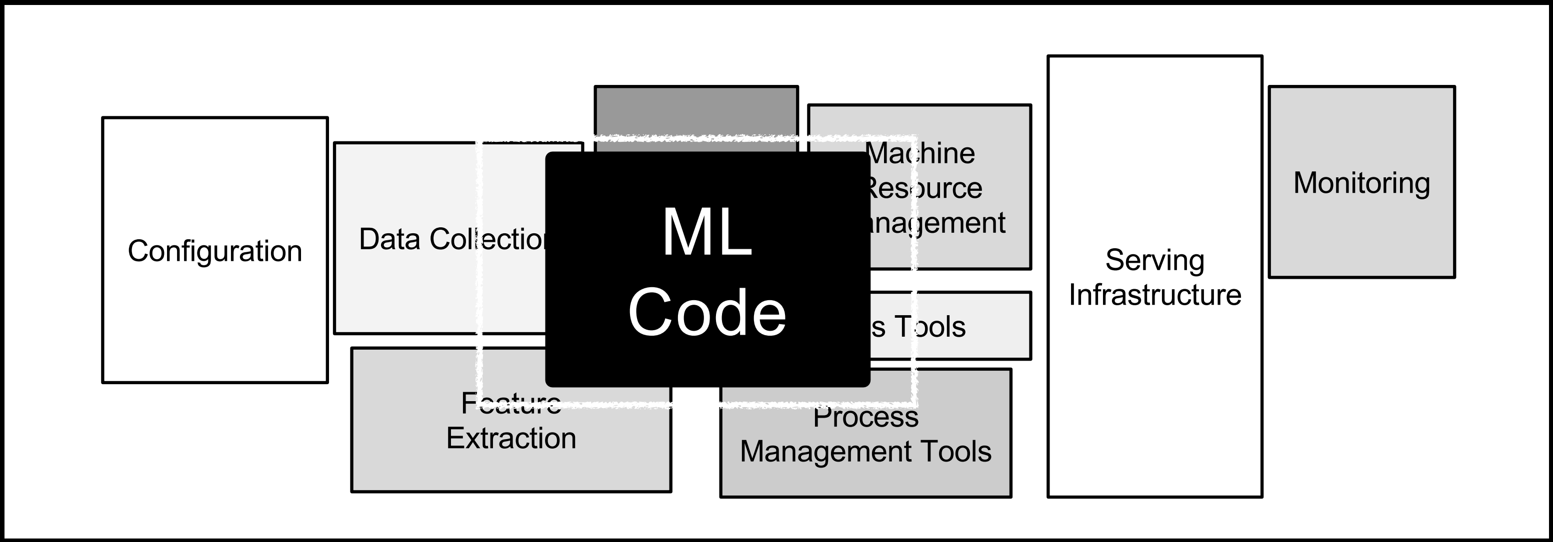
- Algorithmic Correctness (Design and Implementation)
- Reproducible Training
- Model Quality Degradation

What should we test for ML models?

- Algorithmic Correctness (Design and Implementation)
- Reproducible Training
- Model Quality Degradation

Static Model vs Dynamic Model?

The 2-5%

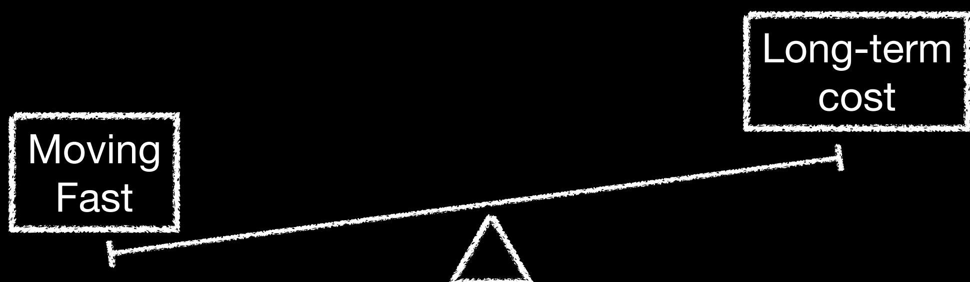


Sculley, David, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. "*Hidden technical debt in machine learning systems*." In Advances in neural information processing systems, pp. 2503-2511. 2015.

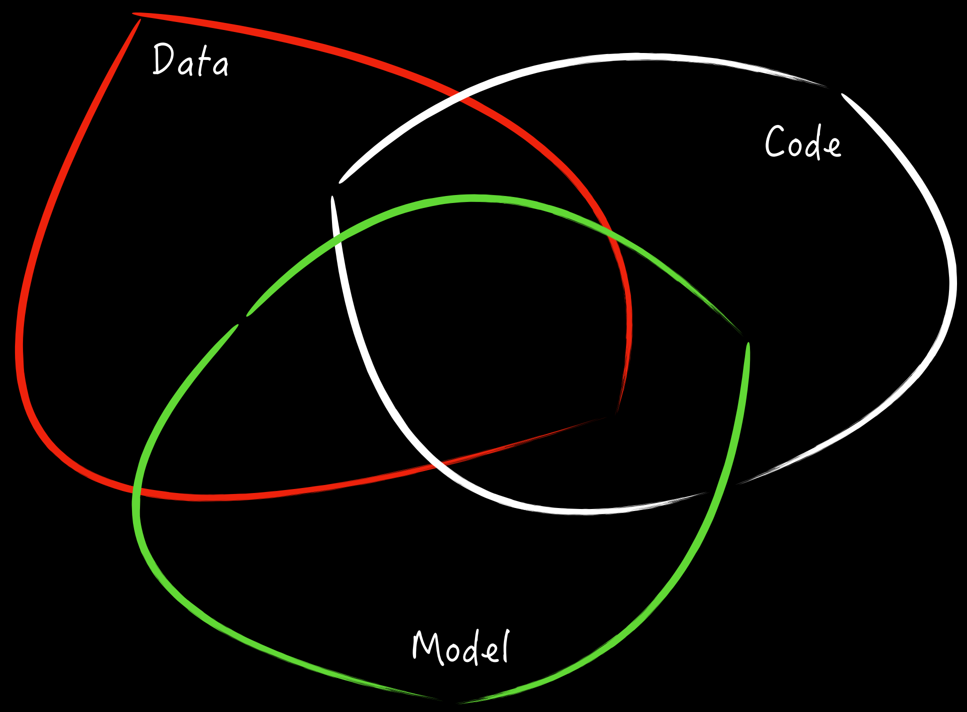
Technical Debt

“not quite right code which we postpone making it right”

Ward Cunningham, “The WyCash Portfolio Management System,”
Proc. OOPSLA, ACM, 1992; http://c2.com/doc/oopsla92.html.



Handle Complexity
Reduce Dependencies
Tests
CI/CD
Documentation
Avoid anti-patterns, code smells


Data
Code
Model

“A system is an interconnected set of elements that is coherently organized in a way that achieves something.”

—Meadows

Why system surprises us?

What Meadows told us in TIS

- Beguiling Events

A system is a big black box
Of which we can't unlock the locks,
And all we can find out about
Is what goes in and what comes out.
Perceiving input-output pairs,
Related by parameters,
Permits us, sometimes, to relate
An input, output and a state.
If this relation's good and stable
Then to predict we may be able,
But if this fails us—heaven forbid!
We'll be compelled to force the lid!

—Kenneth Boulding

Why system surprises us?

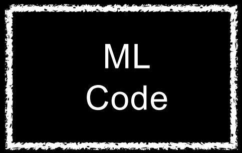
What Meadows told us in TIS

- Linear Minds in a Nonlinear World
  - Linear systems can be taken apart and put them together again— the pieces add up.
  - A nonlinear relationship is one in which the cause does not produce a proportional effect.

Why system surprises us?

What Meadows told us in TIS

- Nonexistent Boundaries
  - Boundaries are of our own making, and that they can and should be reconsidered for each new discussion, problem, or purpose.



Why system surprises us?

What Meadows told us in TIS

- Layers of Limits
  - There are layers of limits around every growing plant, child, epidemic, new product, technological advance, company, city, economy, and population.
  - Insight comes not only from recognizing which factor is limiting, but from seeing that growth itself depletes or enhances limits and therefore changes what is limiting.

Why system surprises us?

What Meadows told us in TIS

- Ubiquitous Delays
  - Most flows in systems have delays—shipping delays, perception delays, processing delays, maturation.
  - When there are long delays in feedback loops, some sort of foresight is essential.

Why system surprises us?
What Meadows told us in TIS

- Bounded Rationality
  - People make quite reasonable decisions based on the information they have. But they don't have perfect information, especially about more distant parts of the system.
  - information flows, goals, incentives, and disincentives can be restructured so that separate, bounded, rational actions do add up to results that everyone desires.

Activity:

- Choose at least two hidden technical debts from the assigned reading and explain them to your room member
- Identify if those technical debts are due to any of the following reasons:
    - Beguiling Events
    - Linear Minds in a Nonlinear World
    - Nonexistent Boundaries
    - Layers of Limits
    - Ubiquitous Delays
    - Bounded Rationality
- Annotate if those technical debt is relevant to Data, Code, or Model.