Characterizing and Detecting Mismatch in ML-Enabled Systems

Problem

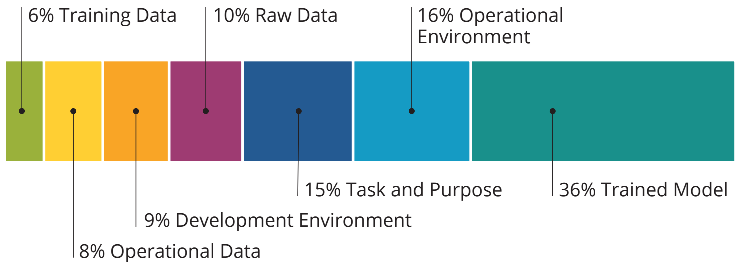
Development, deployment, and operation of ML systems involves three perspectives, often with three completely separate workflows and people: data scientists build the model; software engineers integrate the model into a larger system; and then operations staff deploy, operate, and monitor the system.

Because these perspectives operate separately and often speak different languages, there are opportunities for mismatch between the assumptions made by each perspective with respect to the elements of the ML-enabled system, and the actual guarantees provided by each element.

Solution

Develop descriptors for elements of ML-enabled systems by eliciting examples of mismatch from practitioners; formalizing definitions of each mismatch in terms of data needed to support detection; and identifying potential for using this data for automation of mismatch detection.

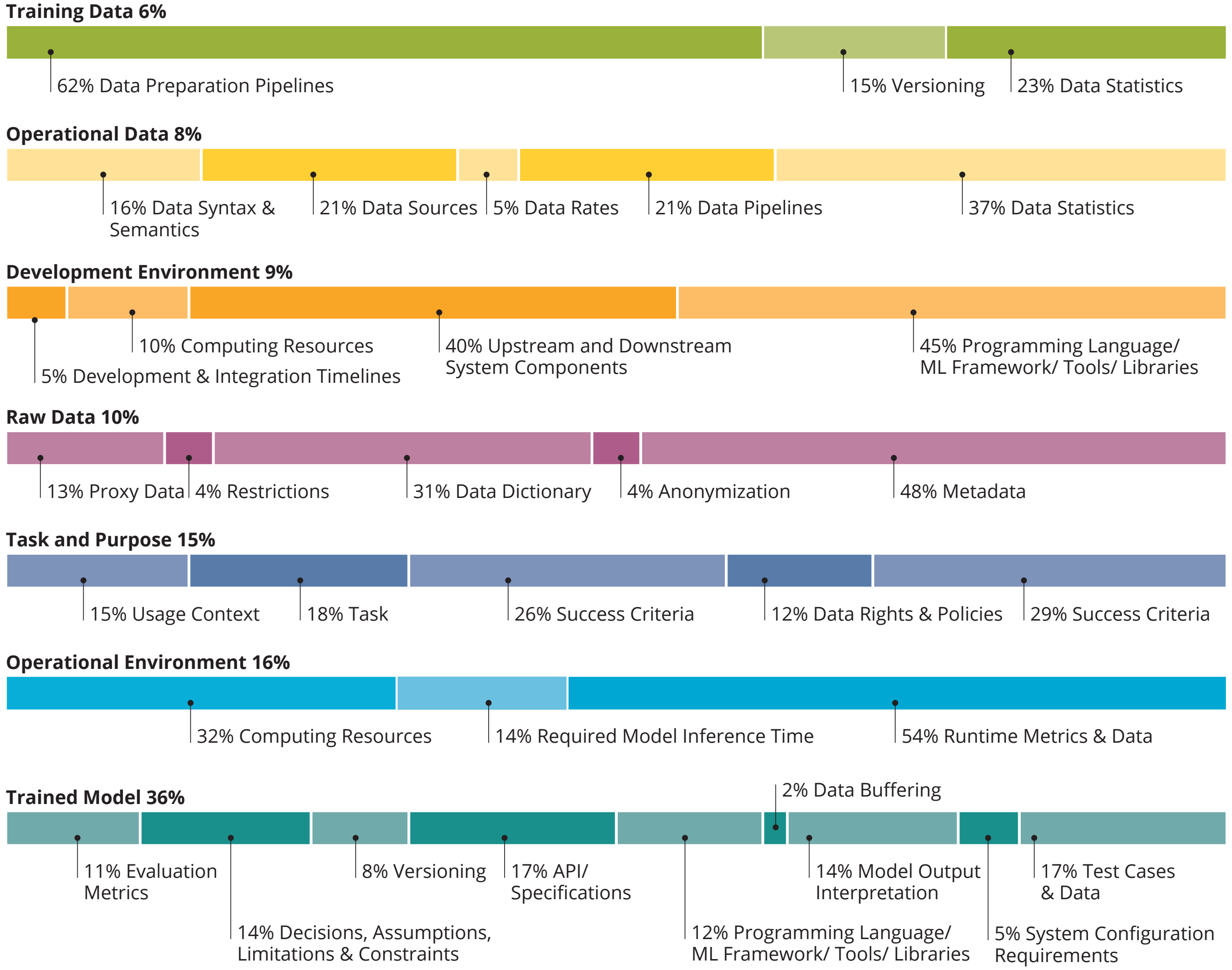
Phase 1: Practitioner interviews to elicit examples of mismatch and their consequences



Resulting Mismatch Categories from Practitioner Interviews

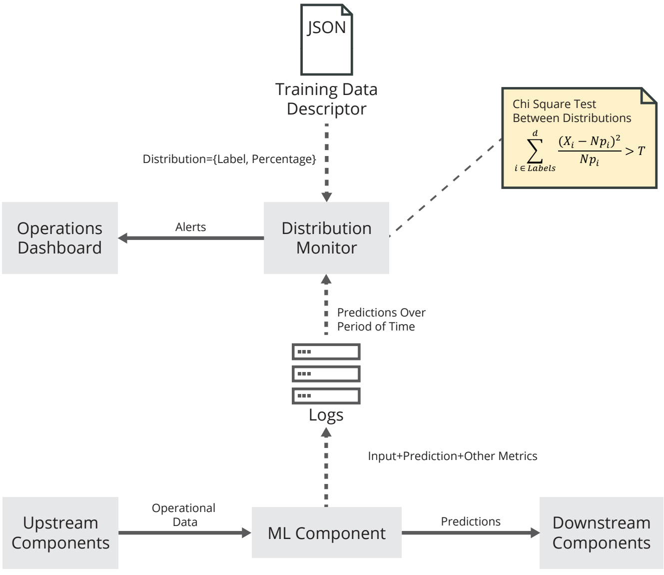
Descriptors for ML system elements **make stakeholder assumptions explicit** and prevent mismatch.

Phase 1 Findings



Training Data mismatches are mostly due to lack of clarity on data preparation pipelines (37%) and lack of data statistics (21%).
Operational Data mismatches are mostly due to lack of data statistics (37%) and lack of clarity on data pipelines (21%).
Development Environment mismatches are mostly due to differences in programming languages ... (45%) and lack of knowledge of upstream and downstream components (40%).
Raw Data mismatches are mostly associated with lack of metadata (48%) and lack of a "data dictionary" (31%).
Task and Purpose mismatches are mostly associated with unknown business goals (29%) or success criteria (26%).
Operational Environment mismatches are mostly associated with unavailable runtime metrics and data (54%) and unawareness of computing resources available for model serving (32%).
Trained Model mismatches are mostly associated with lack of test cases and test data (17%) and lack of model specifications and APIs (17%).

Looking Ahead: Automated Mismatch Detection



Descriptors Being Used for Automated Drift Detection

Principal Investigator, Grace A. Lewis | info@sei.cmu.edu
Stephany Bellomo | Ipek Ozkaya | April Galyardt

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.