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 Findings

<table>
<thead>
<tr>
<th>Training Data 6%</th>
<th>Operational Data 8%</th>
<th>Development Environment 9%</th>
<th>Raw Data 10%</th>
<th>Task and Purpose 15%</th>
<th>Operational Environment 16%</th>
<th>Trained Model 36%</th>
</tr>
</thead>
<tbody>
<tr>
<td>6% Training Data</td>
<td>10% Raw Data</td>
<td>16% Operational Environment</td>
<td>1% Raw Operational Environment</td>
<td>16% Task and Purpose</td>
<td>18% Operational Environment</td>
<td>15% Task and Purpose</td>
</tr>
<tr>
<td>42% Data Preparation Pipelines</td>
<td>16% Data Syntax &amp; Semantics</td>
<td>10% Computing Resources</td>
<td>13% Proxy Data</td>
<td>32% Computing Resources</td>
<td>11% Evaluation Metrics</td>
<td>11% Evaluation Metrics</td>
</tr>
<tr>
<td>15% Versioning</td>
<td>21% Data Sources</td>
<td>40% Upstream and Downstream System Components</td>
<td>4% Restrictions</td>
<td>14% Required Model Inference Time</td>
<td>8% Versioning</td>
<td>14% Decisions, Assumptions, Limitations &amp; Constraints</td>
</tr>
<tr>
<td>23% Data Statistics</td>
<td>5% Data Rates</td>
<td>45% Programming Language/ML Framework/Tools/Libraries</td>
<td>31% Data Dictionary</td>
<td>14% Model Output Interpretation</td>
<td>17% API/Specifications</td>
<td>17% API/Specifications</td>
</tr>
<tr>
<td>5% Data Pipelines</td>
<td>21% Data Pipelines</td>
<td>5% Development &amp; Integration Timelines</td>
<td>4% Anonymization</td>
<td>12% Programming Language/ML Framework/Tools/Libraries</td>
<td>12% API/Specifications</td>
<td>12% Programming Language/ML Framework/Tools/Libraries</td>
</tr>
<tr>
<td>37% Data Statistics</td>
<td></td>
<td></td>
<td>48% Metadata</td>
<td></td>
<td>5% System Configuration Requirements</td>
<td></td>
</tr>
</tbody>
</table>

Resulting Mismatch Categories from Practitioner Interviews

- 15% Task and Purpose
- 16% Operational Environment
- 15% Task and Purpose
- 36% Trained Model
- 16% Operational Environment
- 8% Development Environment
- 15% Task and Purpose
- 18% Operational Environment
- 18% Task
- 26% Success Criteria
- 12% Data Rights & Policies
- 29% Success Criteria
- 32% Computing Resources
- 14% Required Model Inference Time
- 54% Runtime Metrics & Data
- 2% Data Buffering
- 11% Evaluation Metrics
- 14% Decisions, Assumptions, Limitations & Constraints
- 17% API/Specifications
- 14% Model Output Interpretation
- 17% Test Cases & Data
- 5% System Configuration Requirements

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

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

- Distribution Monitor
- Distribution Descriptor
- Distribution Dashboard
- Predictions Over Time
- Upstream Components
- Operational Data
- ML Component
- Predictions
- Downstream Components
- Input Prediction/Other Metrics
- Logs

Descriptors Being Used for Automated Drift Detection

- Chi Square Test between Distributions
- JSON

Characterizing and Detecting Mismatch in ML-Enabled Systems