kolena

Best Practices for Trustworthy AI/ML Model Testing

Written By:
Mohamed Elgendy,
CEO/CO-Founder

Since the release of ChatGPT just over a year ago, companies building new AI / ML models across the industry have been primarily focused on identifying and expanding the limits of what AI can do – what problems it can solve and how it can augment and advance existing capabilities across a vast range of challenges.

As the industry grows, though, companies and researchers are also coming to understand the need for scalable, efficient testing and quality control practices for AI / ML models – both to meet customers' and end users' needs, and to streamline compliance with emerging regulatory requirements.

Pitfalls of Current Testing Practices

For many companies, unfortunately, their first encounter with AI / ML testing is likely to be confusing and time-consuming. Current testing practices are manual and haphazard, often relying on the domain knowledge and intuition of the engineers themselves rather than repeatable, standardized processes.

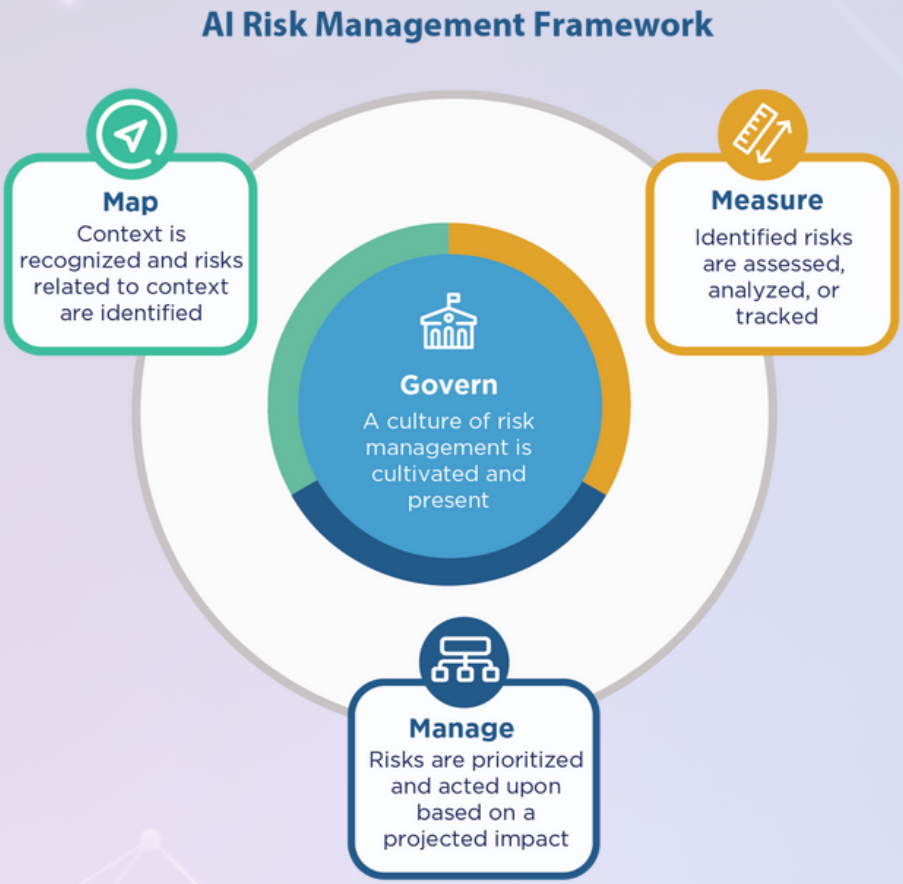
Many teams find that 60% to 80% of their time in developing a model is spent on testing and validating – and they may still find gaps in a model's performance after it's moved into production.

This is due to the 'hidden stratification phenomenon,' in which a model's overall accuracy scores may increase due to strong performance in less important scenarios, while regressing on more important scenarios and tasks. These 'silent regressions' can happen every time a model is updated on new data. (Read more about aggregate metrics and the hidden stratification problem here.)

Current testing practices, then, leave engineers shooting in the dark to improve their models, while undermining businesses' ability to accurately explain the strengths and weaknesses of their ML products. They also make it extremely difficult for product managers to create clearly defined roadmaps – and make customers far more likely to lose trust in AI.

NIST's AI Risk Management Framework

The National Institute of Standards and Technology (NIST) published a risk management framework in January 2023 to provide a systematic, consistent guide to organizations pursuing trustworthy AI.



NIST's AI Risk Management Framework

The framework defines the characteristics of AI trustworthiness as reliability, explainability, and a transparent quality process. These guidelines are already having profound implications for the development and implementation of sound and scalable testing processes across the AI / ML industry.

NIST's framework states that companies must accomplish the following to achieve trustworthy AI:

- Map: Identify model risks and edge cases;
- Manage: Track and manage identified risks and make them visible to all stakeholders;
- Measure: Test and analyze the identified risks.

Best Practices for Systematic and Reliable ML Model Validation

So, how can companies translate NIST's guidance into action? Kolena has developed a three-part approach based on scalable, repeatable best practices that map 1:1 with NIST's framework.

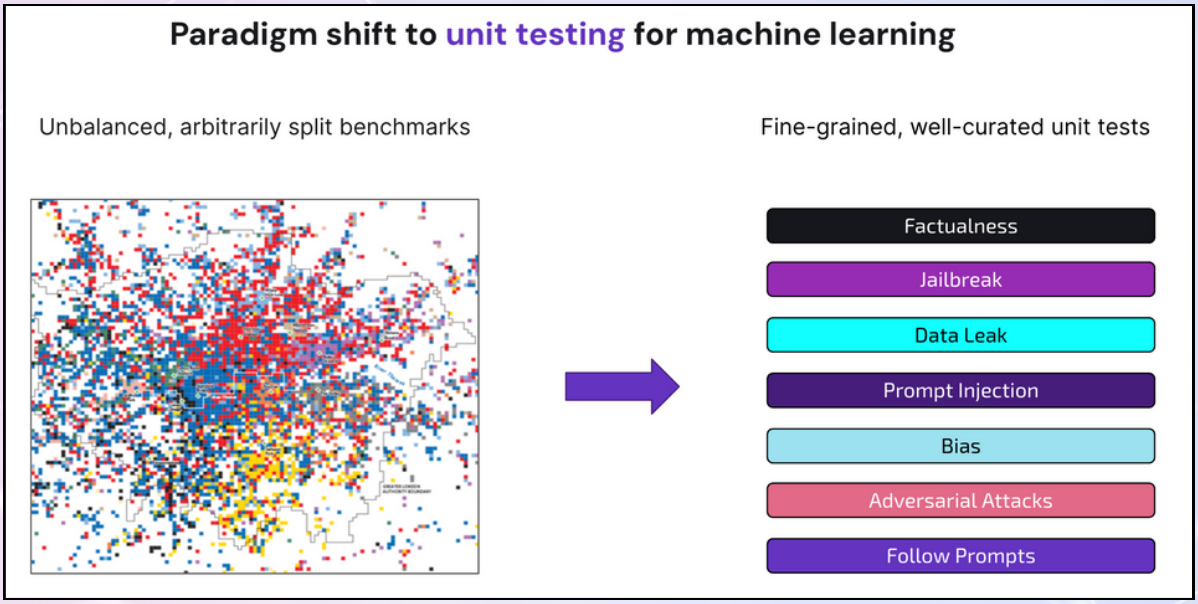
These best practices apply for computer vision models, NLPs, LLMs, and generative models, as well as for multimodal, structured data and all ML use cases:

1.Curate high-fidelity tests (Corresponds to: "Map")

This first discipline within Kolena's best practices approach rests on the principle that properly managing a model's test data is just as important – if not more so – than managing its training data in ensuring reliable and trustworthy behavior.

Training data teaches a model how to generalize and respond, but a company can only measure the accuracy of the model's responses by asking it the right questions and analyzing the resulting test data.

Put simply, everything an organization knows about a model's behavior prior to moving it into production comes from its testing data. Prioritizing and optimizing the use of this data, then, is the first step toward adopting and complying with the NIST framework.



A new paradigm of managing test datasets in the form of fine-grained, well-curated unit tests (testing scenarios).

Unit Testing for Machine Learning: In order to curate high-fidelity tests that will fulfill NIST's guidance of identifying model risks and edge cases, companies first need to implement solutions that allow them to break their testing data down into granular, scenario-level tests, or 'unit tests.'

Rather than testing a model's capabilities based on an engineer's intuition, unit testing allows companies to test performance at a granular, scenario-specific level across categories such as factualness, jailbreak, data leaks, prompt injection, bias and others.

Using this unit testing approach, companies can ensure they are asking their models the right questions to gauge their capabilities in the most important scenarios they will be trusted to perform.

Balanced Distribution: Biased test distribution can produce misleading results. For example, if a model's testing data is overly weighted toward less-important tasks on which the model consistently scores well, but under-samples more important tests on which its performance is lacking, the resulting view of the model's overall accuracy and performance will be skewed and unreliable.

2. Standardize Quality Rubric (Corresponds to: "Manage")

Standardizing the quality assurance process is crucial to ensuring that all defined risks are tracked across the different phases of the ML development cycle to build team alignment and trust. There are three main criteria to standardize:

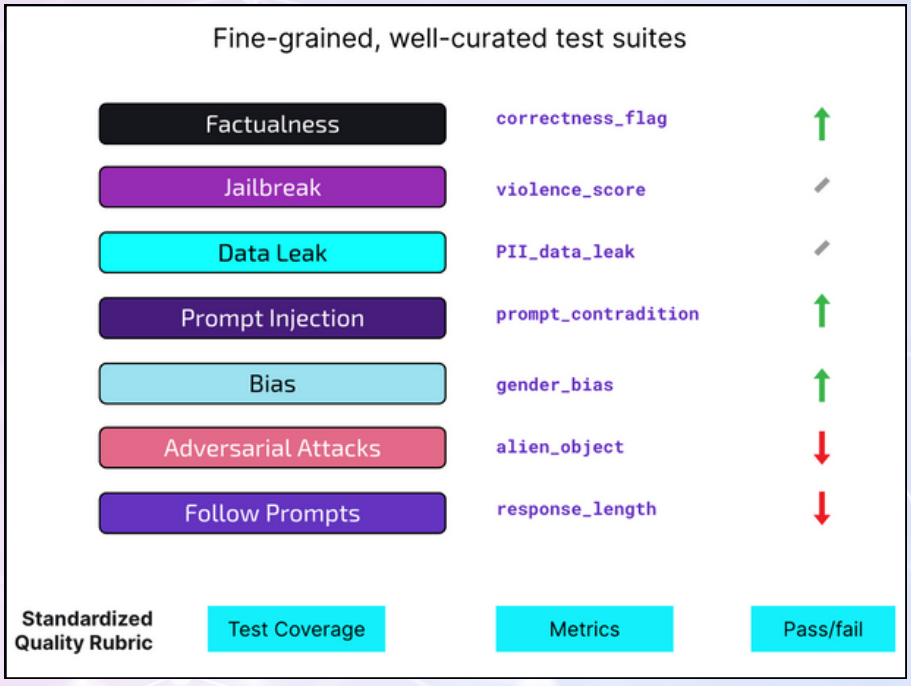
Standardizing Test Coverage: The most commonly asked questions for companies launching new AI / ML products are: "Are we confident that we thoroughly tested our models?" and "What scenarios are we testing against?" Risks identified in Phase 1 should be clearly visible and accessible as companies begin to standardize their test coverage.

This step requires strong and consistent communication between members of a team and across an organization, in order to build and draw upon an institutional knowledge base of key testing scenarios and identified risks. There should be no surprises at any level of an organization as to what tests are being applied to a model; maintaining consistent and well-considered test coverage parameters goes a long way toward building team alignment and trust.

Standardize Product Metrics: Traditional ML metrics such as precision, recall, F1, AUC, BERT, etc. can be misleading when applied to a product-level test. With this in mind, teams will be better served to establish customized, product-level performance metrics (such as, for example, a custom "collision risk" metric for robotics systems, or an "expected revenue" metric for recommender systems) to guide their testing efforts.

Standardizing Success Criteria: Success criteria and pass / fail requirements should also be clearly and consistently applied across teams and organizations, whether this is done by assigning weighted importance scores across a given suite of tests; assigning a passing score on tests that show no regression between each update of the model; applying a basic threshold score; or by other methods.

Breaking the test dataset down into unit tests reflecting the identified product risks allows for setting up specific metrics and pass/fail criteria for each unit test.



3. Product-Level Testing (Corresponds to: “Measure”)

End-to-End System Testing: Models do not function in isolation; in order to provide value to the end customer, most of them are part of a broader system that performs a constellation of complex and inter-related tasks. These systems may incorporate various pre- and post-processing steps and logic, or may see a model connected to a pipeline of other models that handle other elements of the larger task.

In order to properly test and analyze a model’s identified risks – per the NIST framework – companies must have a holistic view of the performance of the entire system, not simply of the core model or models within it.

They can achieve this by, for example, identifying which combination of tests across component models in a larger system produces the most desirable product-level result - not simply by evaluating the efficacy of tests on individual models within the pipeline.

Product-Level Metrics: End-to-end system tests should drive toward a clearly-defined set of performance metrics, similar to model-level metrics such as precision, recall, and others. These metrics, however, should test for product-level outcomes, rather than measuring the isolated performance of individual models with the system.

Example 2 below illustrates this. Although there are multiple discrete models within the computer vision system being tested, the product-level goal is not simply to identify pedestrians or their walking direction - it's to identify and avoid collision risks.

The examples below demonstrate how engineers and teams can define product metrics vs traditional ML metrics.

Example 1: `revenue_expected` metric for recommender systems

```
def expected_revenue(product_price, quantity_remaining, clicking_rate, listing_age, is_holiday):
    sales_rate = clicking_rate * (1 / listing_age)
    if is_holiday:
        sales_rate *= 1.75
    return sales_rate * quantity_remaining * product_price
```

Example 2: `collision_risk` metric for robotics systems

```
def is_collision_risk(sample):
  return (
    sample.is_false_negative
    and sample.distance_meters < 10
  )
```

Conclusion

Although systematic and reliable approaches to testing and validating AI / ML are relatively young - like most technologies in the AI sector - it's already possible to identify and implement core best practices across engineering teams and organizations that will drive vastly superior testing outcomes along with significant time savings.

By adopting the core best practices of using unit testing for machine learning, managing toward a standardized quality rubric, and implementing product-level testing at scale, teams and companies can meet their customers' needs; provide greater visibility and assurance to executives, sales teams and others within their organizations; and - most importantly - create trust among users, regulators and other key stakeholders.

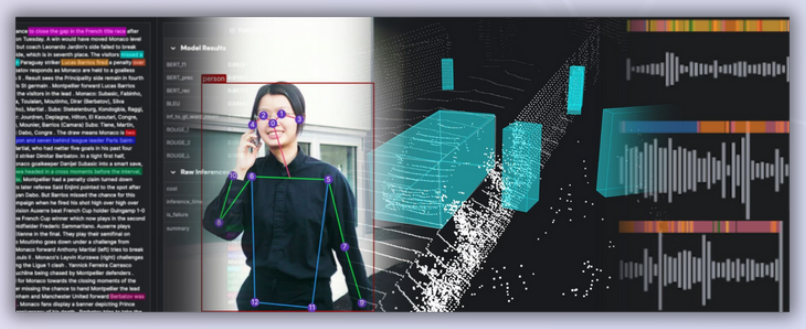
Kolena's AI/ML model testing platform provides a systematic and reliable approach to unit testing. Our interest and expertise in rigorous pre-deployment model testing is something we hope to contribute to NIST and the AI community.

The Testing Platform To Build Trustworthy AI/ML

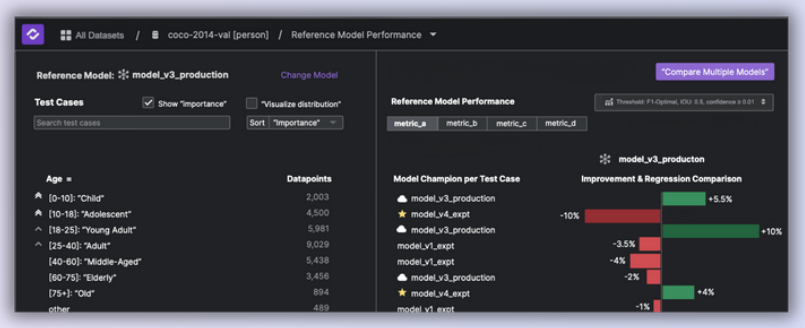
Rigorous • Systematic • Reliable

kolena

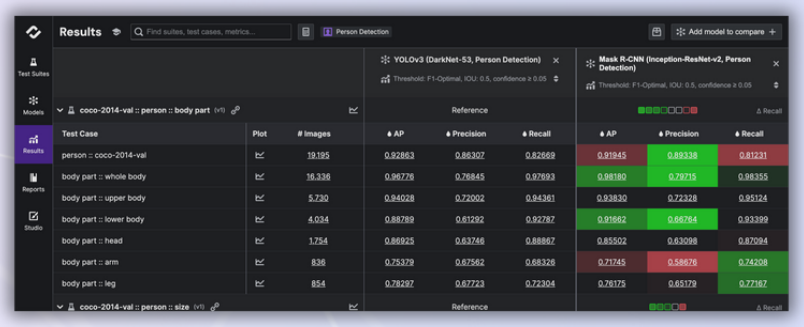
Explore your data and identify blind spots within your domain for crucial test curation.



Optimize your team's QA to foster trust, expand testing scope, and uphold higher quality benchmarks.



Analyze behavioral differences, and evaluate the most fitting model for deployment to production.



Reliable and Systematic Model Testing for Every AI/ML Problem

Computer Vision - Natural Language Processing/Generation
Tabular - Audio - Multimodal - More

Customizable for any workflow, data type, evaluation logic, metric, plots, and report

www.kolena.com | info@kolena.com