Model Testing

ML in Production - Recitation 6

Outline

- What is model testing?
- Why model testing is important?
- Model testing Strategies
- Zeno
- Adatest

What is model testing?

- Model testing in machine learning refers to the process of evaluating the performance of a trained machine learning model on a set of data that is separate from the data used to train the model.
- This process is essential to ensure that the model can generalize well to new, unseen data.

Why is model testing important?

- Detecting model and data drift
- Finding anomalies in dataset
- Detect possible root cause for model failure
- Eliminating bugs and errors
- Reducing false positives and false negatives
- Ensuring robustness of ML model
- Finding new insights within the model

Slicing

- Guide testing by identifying groups and analyzing accuracy of subgroups
- For fairness: gender, country, age groups etc
- For business requirements or cost of mistakes
- Slice test data by population criteria and evaluate interactions
- Identifies problems and plan mitigations, e.g., enhance with more data for subgroup or reduce confidence

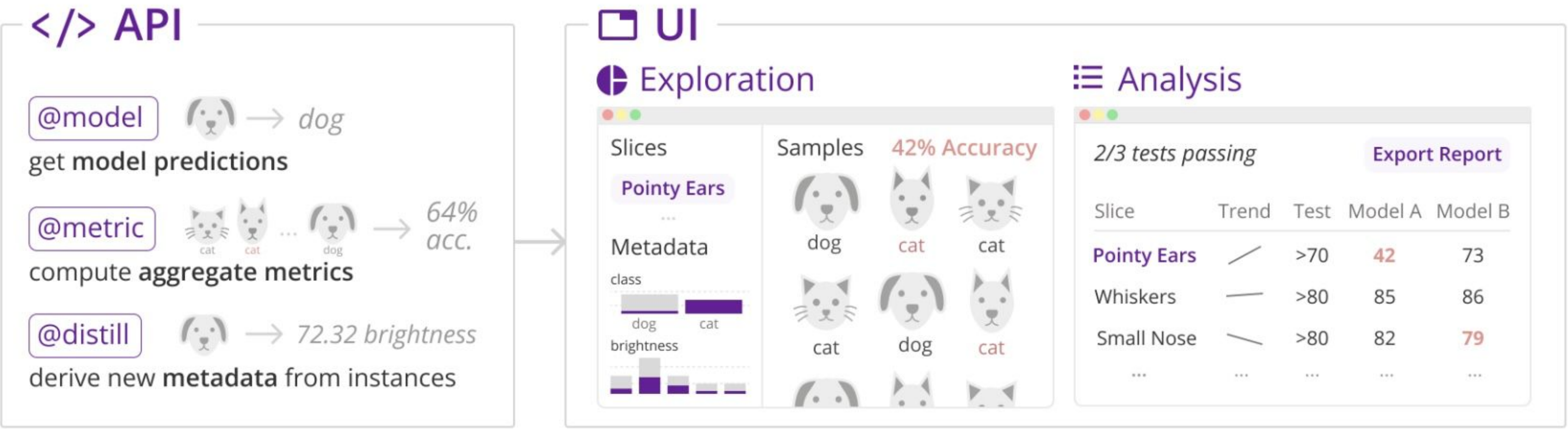
Zeno - A Slicing Tool

Zeno is an interactive platform for exploring and managing your data, debugging your models, and tracking and comparing model performance.

Zeno provides the following features -

1. Explore data and model outputs with customizable views for any data type
2. Interactively test model behaviour
3. Create exportable visualizations and charts comparing models and slices
4. Create custom slicing functions

Zeno Features



</> API

@model → dog

get **model predictions**

@metric → 64% acc.

compute **aggregate metrics**

@distill → 72.32 brightness

derive new **metadata** from instances

UI

Exploration

Slices

Pointy Ears

Metadata

Samples 42% Accuracy

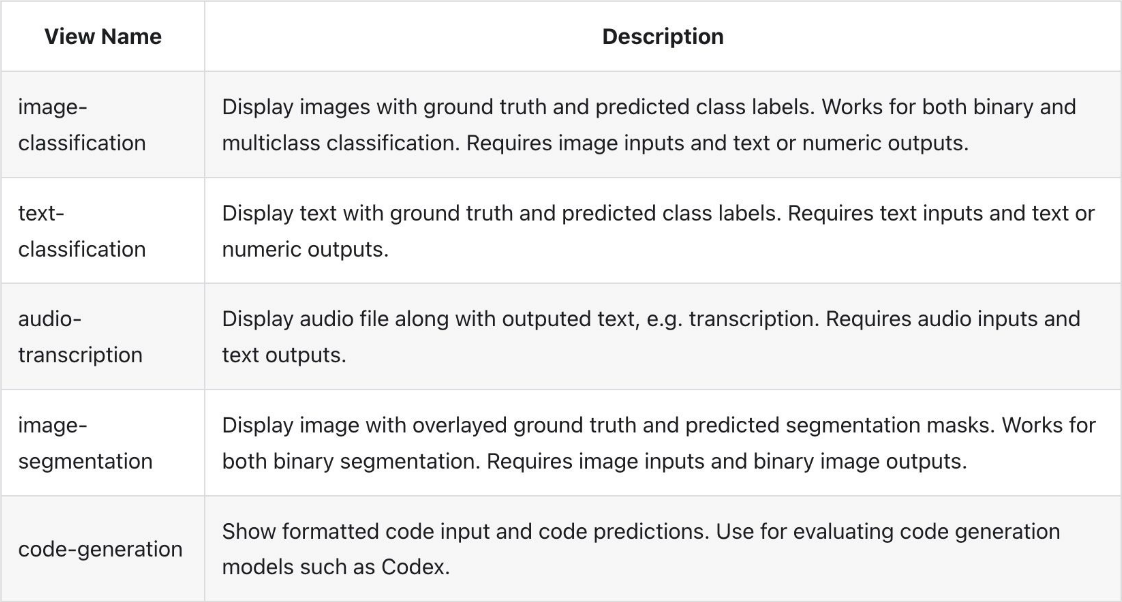
Analysis

2/3 tests passing

Export Report

Slice	Trend	Test	Model A	Model B
Pointy Ears		>70	42	73
Whiskers		>80	85	86
Small Nose		>80	82	79
...	...	...	...	...

Zeno Views



View Name	Description
image-classification	Display images with ground truth and predicted class labels. Works for both binary and multiclass classification. Requires image inputs and text or numeric outputs.
text-classification	Display text with ground truth and predicted class labels. Requires text inputs and text or numeric outputs.
audio-transcription	Display audio file along with outputed text, e.g. transcription. Requires audio inputs and text outputs.
image-segmentation	Display image with overlayed ground truth and predicted segmentation masks. Works for both binary segmentation. Requires image inputs and binary image outputs.
code-generation	Show formatted code input and code predictions. Use for evaluating code generation models such as Codex.

Zeno Activity

Compare 4 simple PyTorch CNN models trained for different number of epochs. Pick a suitable model of your choice. Perform model testing on it and come up with 2 custom slices to potentially identify problems in the model and suggest improvement strategies for them.

Access the demo here.

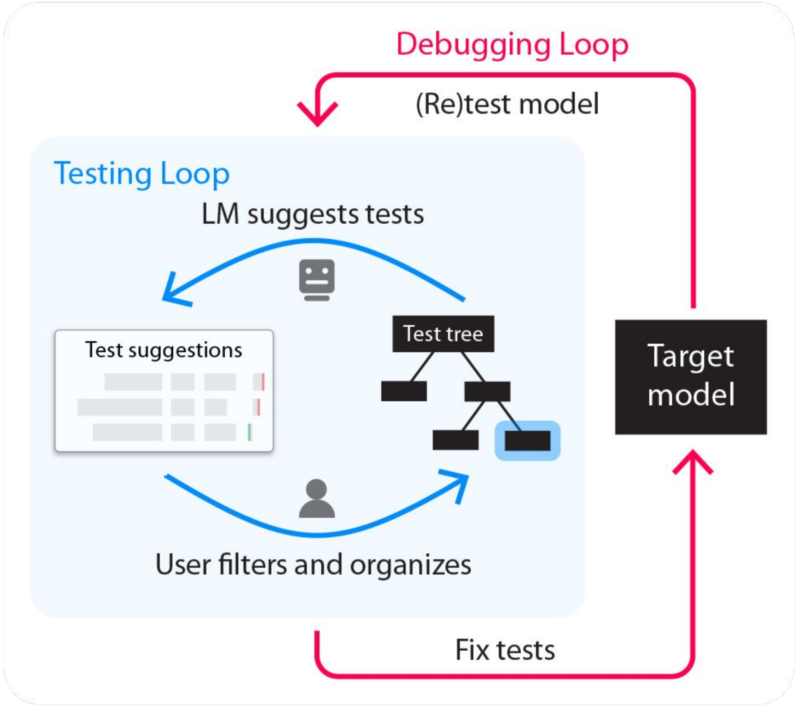
Testing Capabilities

- Capabilities are partial specifications of expected behavior (not expected to always hold)
- Some capabilities correspond to slices of existing test data, for others we may need to create new data
- Encode domain knowledge of the problem to create new data
  1. Capabilities are inherently domain specific
  2. Curate capability-specific test data for a problem

AdaTest - LM Capabilities Testing Tool

- Model testing tool specifically made for language models by Microsoft
- AdaTest uses language models against themselves to build suites of unit tests.
- It is an iterative process between a user and a language model that results in a tree of unit tests specifically adapted to the model you are testing.
- Fixing any failed tests with fine-tuning then leads to an iterative debugging process similar to traditional software development

AdaTest


Debugging Loop
(Re)test model
Testing Loop
LM suggests tests
Test suggestions
Test tree
Target model
User filters and organizes
Fix tests

Testing Capabilities Examples - Sentiment Analysis

- Handles clear positives well
- Handles negation / double negation
- Robustness to typos
- Ignore synonyms and abbreviations
- Person and location names are irrelevant
- Ignore gender

AdaTest Activity

Test a basic open source sentiment analysis model using test cases generated by the language model itself based on the capabilities mentioned in the previous slide. For this demonstration we will be using OpenAI's GPT-3.

Access code for the demo here.

Other Popular Model Testing Tools

1. **Deepchecks** - open-source Python framework for testing ML Models & Data
2. **Drifter-ML** - ML model testing tool specifically written for the Scikit-learn library
3. **Kolena.io** - focuses mostly on the ML unit testing and validation process at scale.

Dive Deeper

Zeno - https://zenoml.com/docs/quickstart

Adatest - https://github.com/microsoft/adatest