# ML Evaluation +  Zeno

Alex Cabrera

October 13, 2022

# Intro

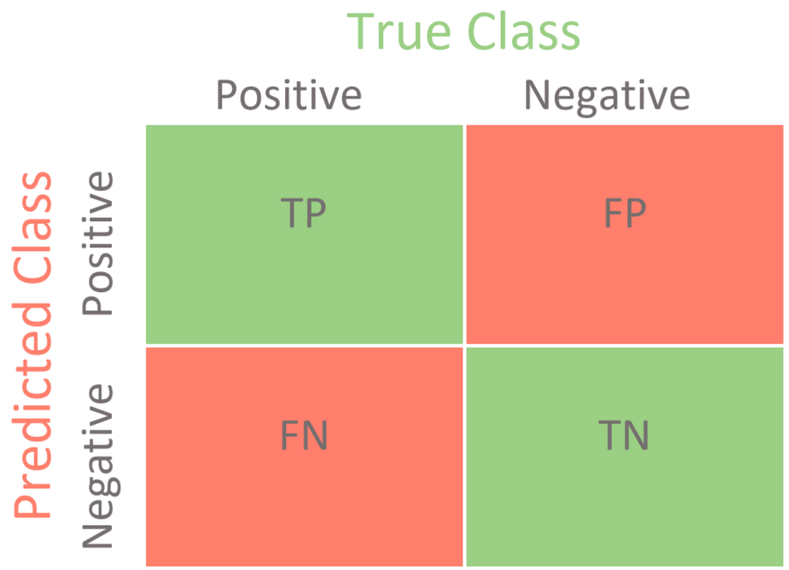
- 4th year PhD student in Human-Computer Interaction
- Undergrad @ Georgia Tech
- Work on tools for machine learning evaluation
  - Fairness, blindspots, etc.
- cabreraalex.com

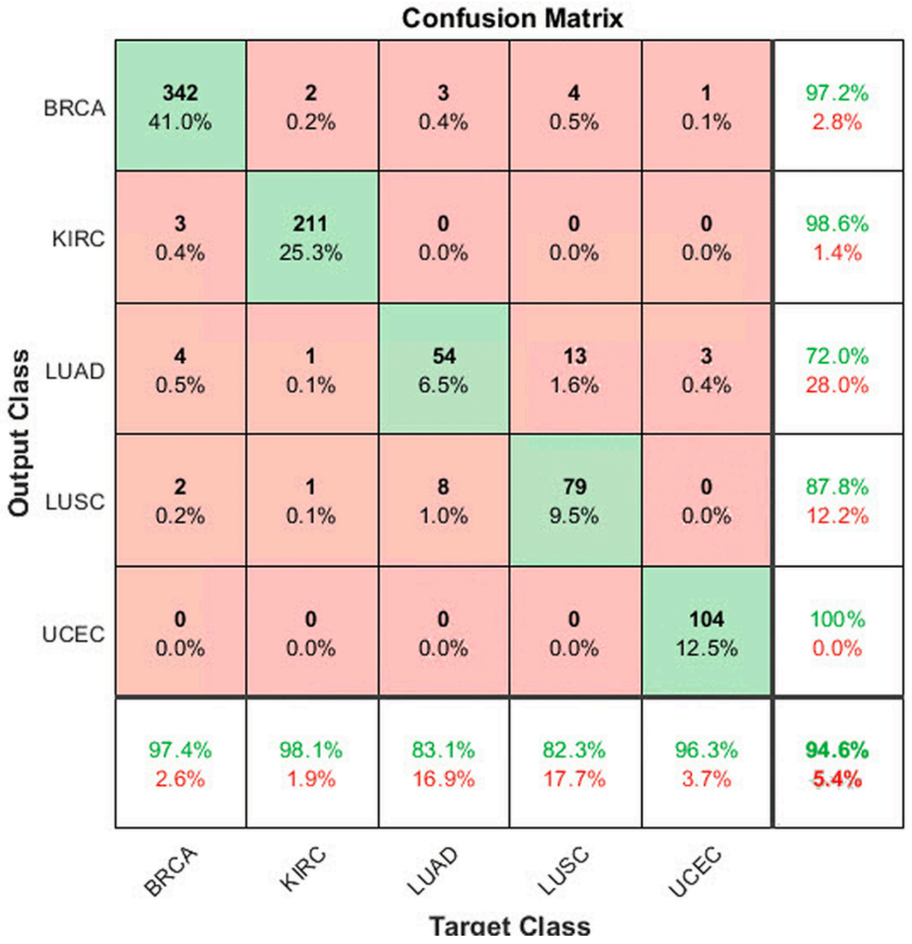
# Overview

1. Output-based Evaluation
2. Input-based (Behavioral) Evaluation
   - Slice-based Analysis
   - Metamorphic Testing
   - Blindspot Discovery
3. Zeno - Interactive ML Evaluation Framework

# Output-based Evaluation

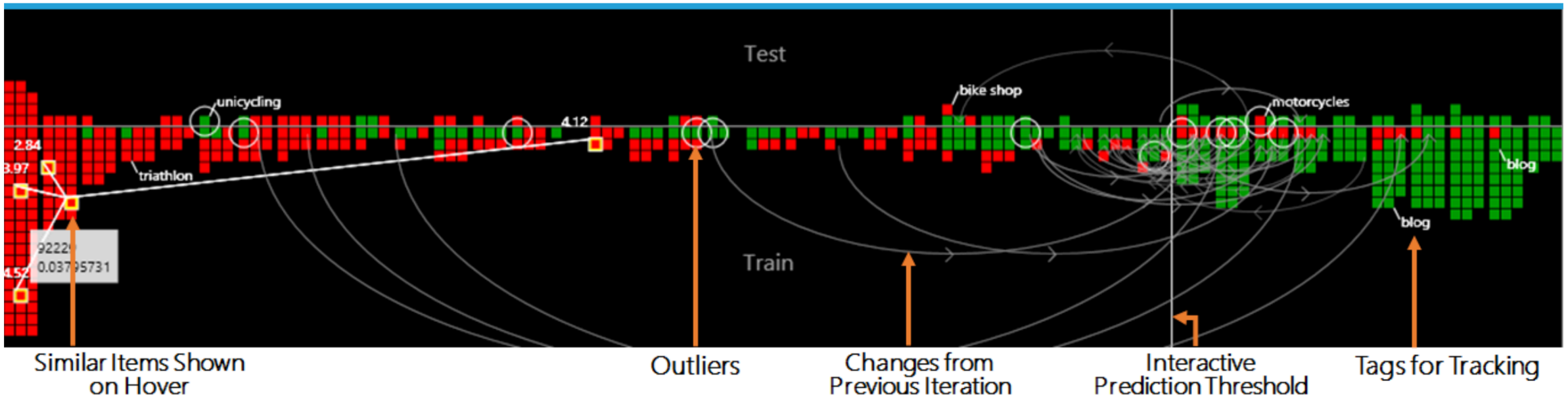
# Confusion Matrix



| | | CONDITION determined by "Gold Standard" | | | |
|---|---|---|---|---|---|
| | TOTAL POPULATION | CONDITION POS | CONDITION NEG | PREVALENCE<br>CONDITION POS / TOTAL POPULATION | |
| TEST OUT-COME | TEST POS | True Pos<br>TP | *Type I Error*<br>False Pos<br>FP | *Precision*<br>Pos Predictive Value<br>PPV = TP / TEST P | False Discovery Rate<br>FDR = FP / TEST P |
| | TEST NEG | *Type II Error*<br>False Neg<br>FN | True Neg<br>TN | False Omission Rate<br>FOR = FN / TEST N | Neg Predictive Value<br>NPV = TN / TEST N |
| | ACCURACY<br>ACC<br>ACC = (TP + TN) / TOT POP | *Sensitivity (SN), Recall*<br>Total Pos Rate<br>TPR<br>TPR = TP / CONDITION POS | *Fall-Out*<br>False Pos Rate<br>FPR<br>FPR = FP / CONDITION NEG | Pos Likelihood Ratio<br>LR +<br>LR + = TPR / FPR | Diagnostic Odds Ratio<br>DOR<br>DOR = LR + / LR - |
| | | *Miss Rate*<br>False Neg Rate<br>FNR<br>FNR = FN / CONDITION POS | *Specificity (SPC)*<br>True Neg Rate<br>TNR<br>TNR = TN / CONDITION NEG | Neg Likelihood Ratio<br>LR -<br>LR - = TNR / FNR | |

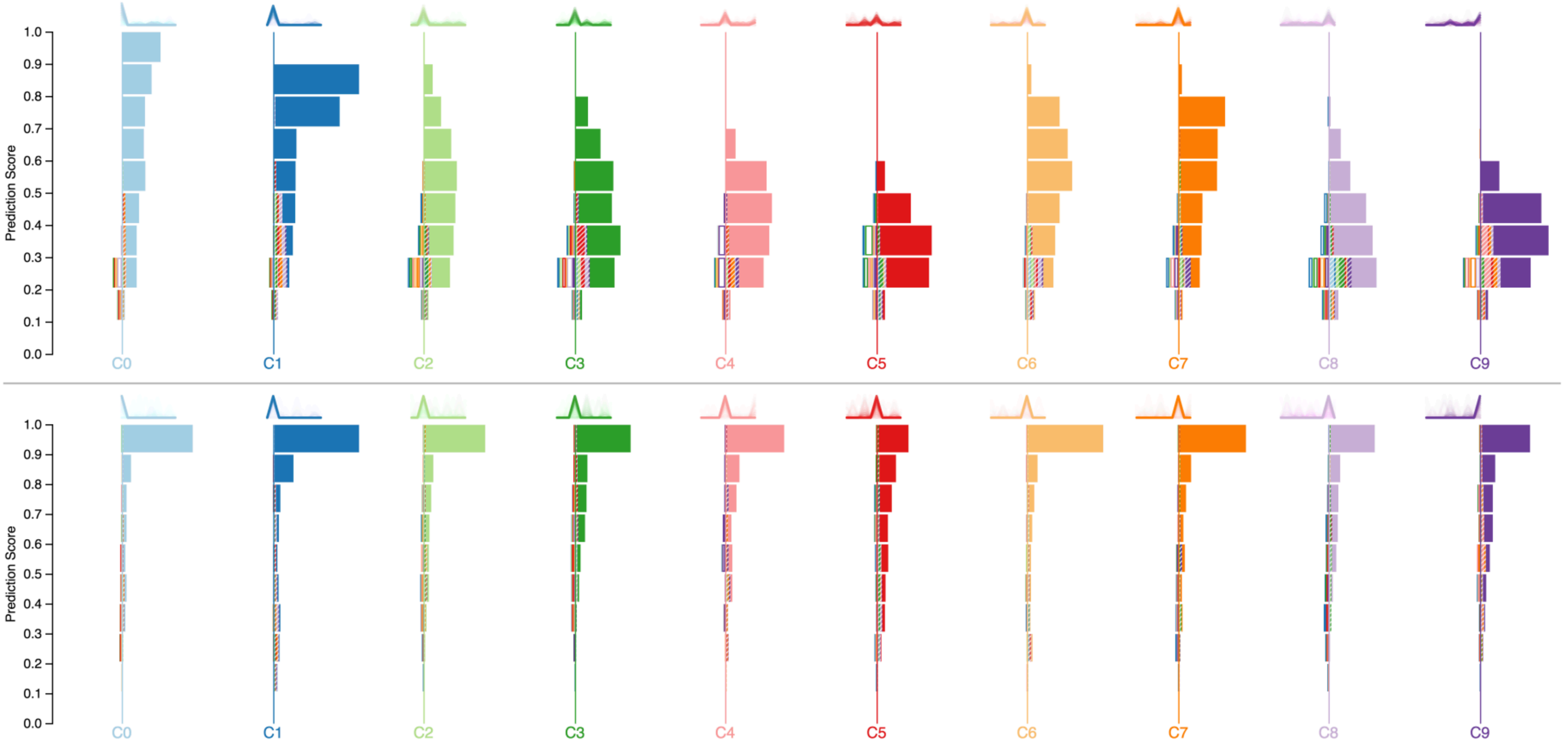
# ModelTracker



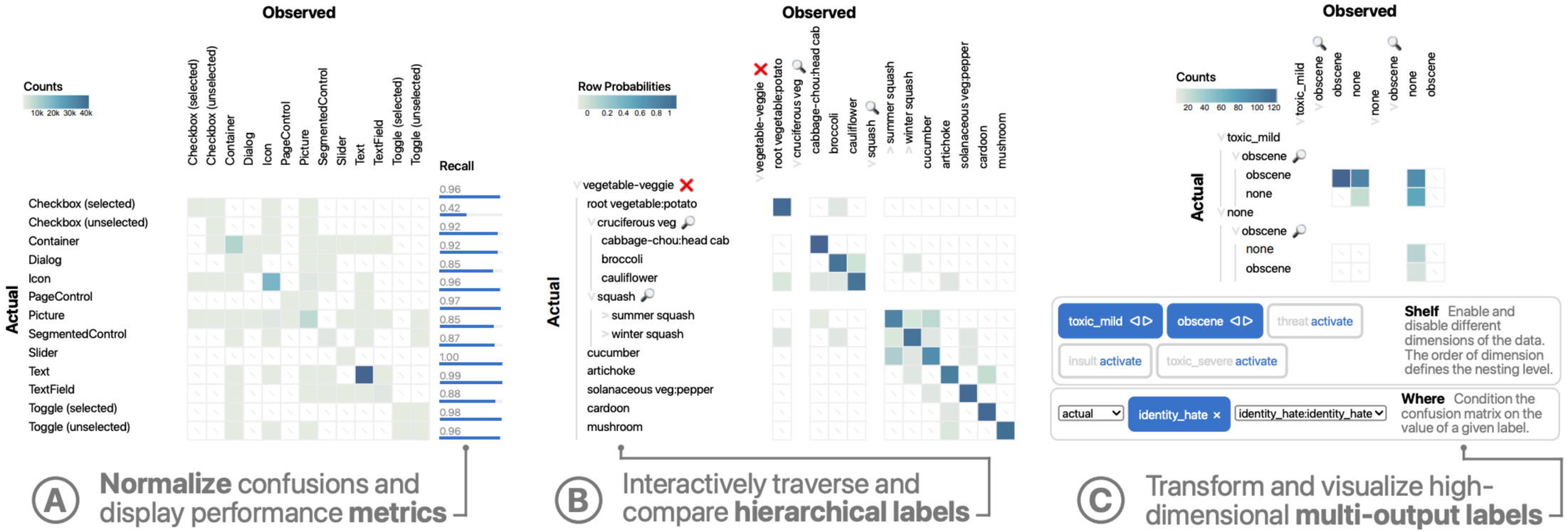

https://dl.acm.org/doi/10.1145/2702123.2702509

# Squares

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/09/squares.VAST_.2016-1.pdf



# Hierarchical Confusion Matrices (Neo)

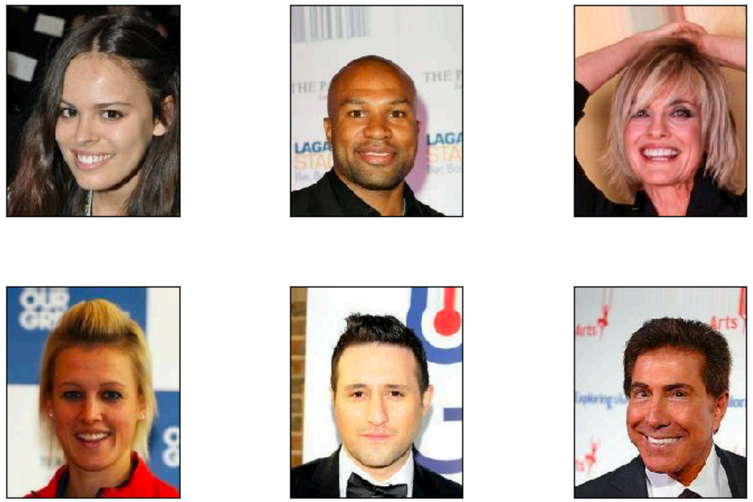



https://machinelearning.apple.com/research/generalizing-confusion-matrix

# Beyond Aggregate Metrics

# Gender Shades

Audit of gender classification models



88.82% accuracy

| | M | F |
|---|---|---|
| **M** | 3,280 | 223 |
| **F** | 543 | 2,810 |

| Gender Classifier | Darker Male | Darker Female | Lighter Male | Lighter Female | Largest Gap |
|---|---|---|---|---|---|
| Microsoft | 94.0% | 79.2% | 100% | 98.3% | 20.8% |
| FACE++ | 99.3% | 65.5% | 99.2% | 94.0% | 33.8% |
| IBM | 88.0% | 65.3% | 99.7% | 92.9% | 34.4% |

http://gendershades.org/overview.html

# COMPAS Recidivism Prediction

Algorithm to predict how likely someone is to recommit a crime if given bail

61% accuracy

**Prediction Fails Differently for Black Defendants**

| | WHITE | AFRICAN AMERICAN |
|---|---|---|
| Labeled Higher Risk, But Didn't Re-Offend | 23.5% | 44.9% |
| Labeled Lower Risk, Yet Did Re-Offend | 47.7% | 28.0% |

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

# Evaluation Beyond Aggregate Metrics

- Evaluating models **beyond aggregate metrics** is essential for responsible ML
- Especially for finding and fixing concerns such as biases and safety issues

**Amazon scraps secret AI recruiting tool that showed bias against women**

The New York Times *Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam*

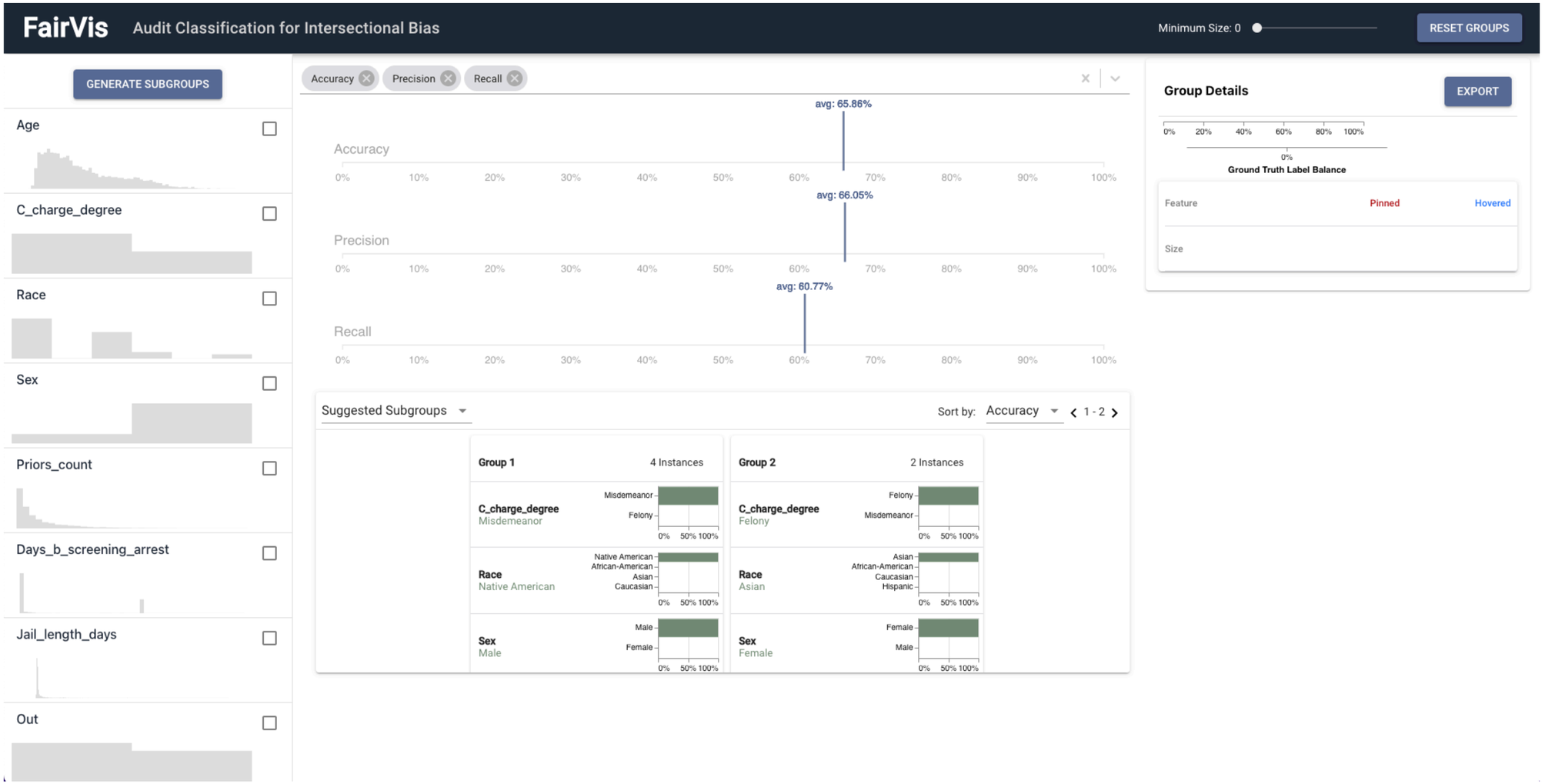
# Slice-based Evaluation

# Slice-based Analysis

- How does a model do across different **subgroups/subsets/slices** defined by the **input features** of my data?
- Often compare slices across different metrics

| Gender Classifier | Darker Male | Darker Female | Lighter Male | Lighter Female | Largest Gap |
|---|---|---|---|---|---|
| Microsoft | 94.0% | 79.2% | 100% | 98.3% | 20.8% |
| FACE++ | 99.3% | 65.5% | 99.2% | 94.0% | 33.8% |
| IBM | 88.0% | 65.3% | 99.7% | 92.9% | 34.4% |

# Auditing for Model Biases with FairVis

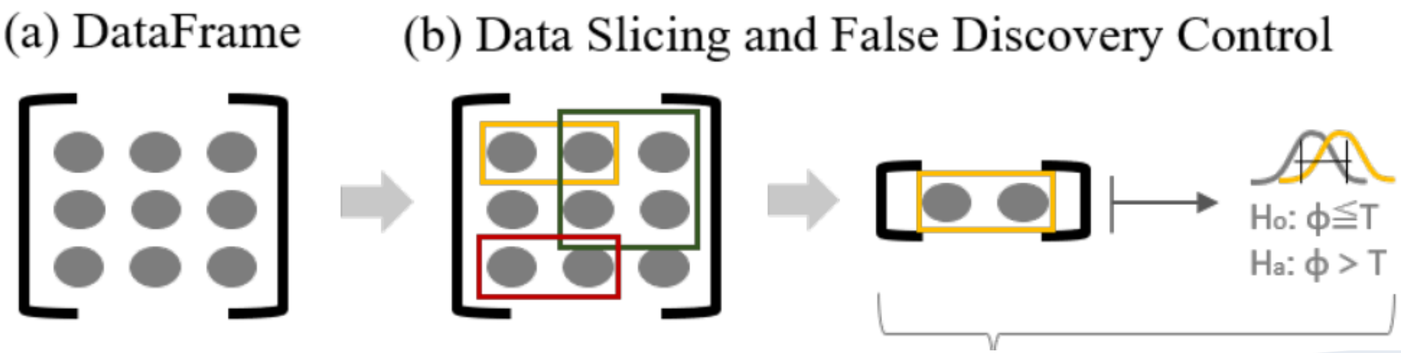



https://poloclub.github.io/FairVis/

# Slice Finder

https://research.google/pubs/pub47966/

- Can we automatically find slices of data with high error?
- **SliceFinder**
  - Start with largest, one-feature slices
  - Compare with child slices for significantly higher error



# Metamorphic Testing

# Metamorphic Testing

- What if we **don’t have sufficient data** to check the types of behaviors we want to check for?
- E.g., want to check how a pedestrian detection performs at night, but don’t have many samples in my test set
- We can instead *generate* instances
- **Metamorphic testing** comes from software engineering - testing that a model behaves as expected when changing the input in specific ways

# CheckList

- Perturb text inputs to audit large language models

**Labels**: positive, negative, or neutral; INV: same pred. (INV) after removals/ additions; DIR: sentiment should not decrease (↑) or increase (↓)

| | Test *TYPE* and Description | Failure Rate (%) | | | | | Example test cases & expected behavior |
|---|---|---|---|---|---|---|---|
| | | | | | | RoB | |
| Vocab.+POS | ***MFT:*** Short sentences with neutral adjectives and nouns | 0.0 | 7.6 | 4.8 | 94.6 | 81.8 | The company is Australian. neutral<br>That is a private aircraft. neutral |
| | ***MFT:*** Short sentences with sentiment-laden adjectives | 4.0 | 15.0 | 2.8 | 0.0 | 0.2 | That cabin crew is extraordinary. pos<br>I despised that aircraft. neg |
| | ***INV:*** Replace neutral words with other neutral words | 9.4 | 16.2 | 12.4 | 10.2 | 10.2 | @Virgin should I be concerned that → when I'm about to fly ... INV<br>@united the → our nightmare continues... INV |
| | ***DIR:*** Add positive phrases, fails if sent. goes down by $> 0.1$ | 12.6 | 12.4 | 1.4 | 0.2 | 10.2 | @SouthwestAir Great trip on 2672 yesterday... You are extraordinary. ↑<br>@AmericanAir AA45 ... JFK to LAS. You are brilliant. ↑ |
| | ***DIR:*** Add negative phrases, fails if sent. goes up by $> 0.1$ | 0.8 | 34.6 | 5.0 | 0.0 | 13.2 | @USAirways your service sucks. You are lame. ↓<br>@JetBlue all day. I abhor you. ↓ |

https://arxiv.org/pdf/2005.04118.pdf

# DeepRoad

https://arxiv.org/pdf/1802.02295.pdf

- We expect computer vision models for self-driving cars to still perform well under different weather conditions
- Use GAN-based model to generate weather-perturbed images


(a) (b)

**Figure 1: Foggy and rainy scenes via DeepTest**

# Blindspot Discovery

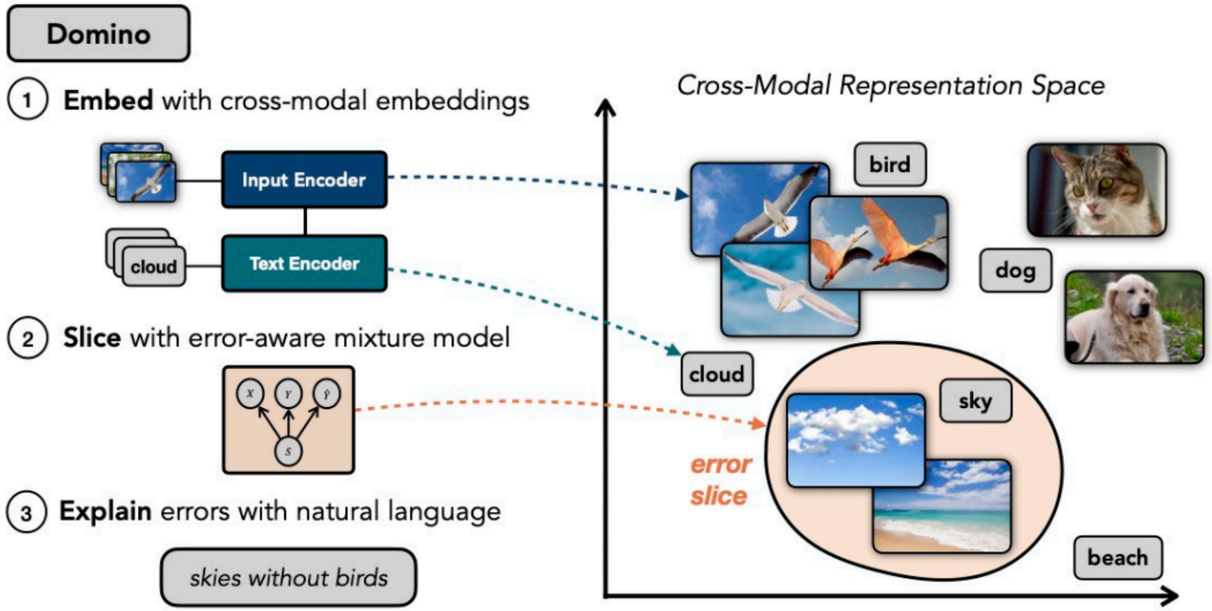
# Blindspot Discovery

- What if we **don’t have metadata** AND **can’t generate data**?
  - Often the case of images, video, audio
- Can we still try to discover model failures?

- **Blindspot Discovery** methods use a model’s representation to find clusters with high error, potential failures.

# Domino

- One example method is **Domino**
- Cluster an embedding representation of the data to find high-error groups
- Use CLIP model to extract natural language descriptions

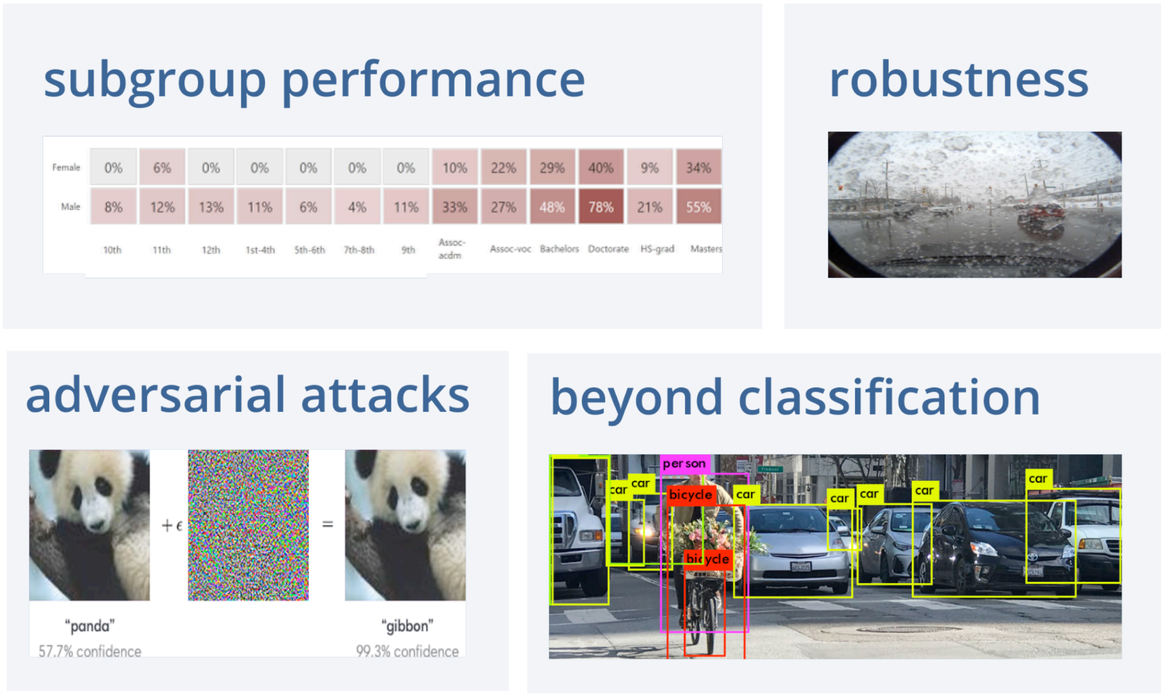



https://arxiv.org/pdf/2203.14960.pdf

# Summary

# Existing Approaches

- Growing number of tools for specific model analyses
  - Fairness toolkits
  - Robustness check tools
  - Adversarial attack libraries
  - Slice-based visualizations




**Despite their proliferation, there has been limited adoption of these tools in real-world ML workflows**

# Interviews with Practitioners

- Conducted **18 need-finding interviews** with ML practitioners to better understand the challenges they face when evaluating their models
- Practitioners evaluate & track model performance on **real-world use cases**
- Mostly an **ad-hoc process**, waiting for failure reports from end users which are then validated in one-off notebooks or scripts
- Existing tools often don’t target relevant use cases or domains, take significant effort to setup, are are hard to reproduce and share.

# Hypothesis

**A unified API and interface for evaluation of machine learning**

- Learn **one evaluation API** applicable to any data or model
- Empower **nontechnical users** to explore model behavior
- **Discover behaviors** using algorithmic and crowdsourced methods.
- Share results with cross-functional teams in a **collaborative** hub.

# Zeno

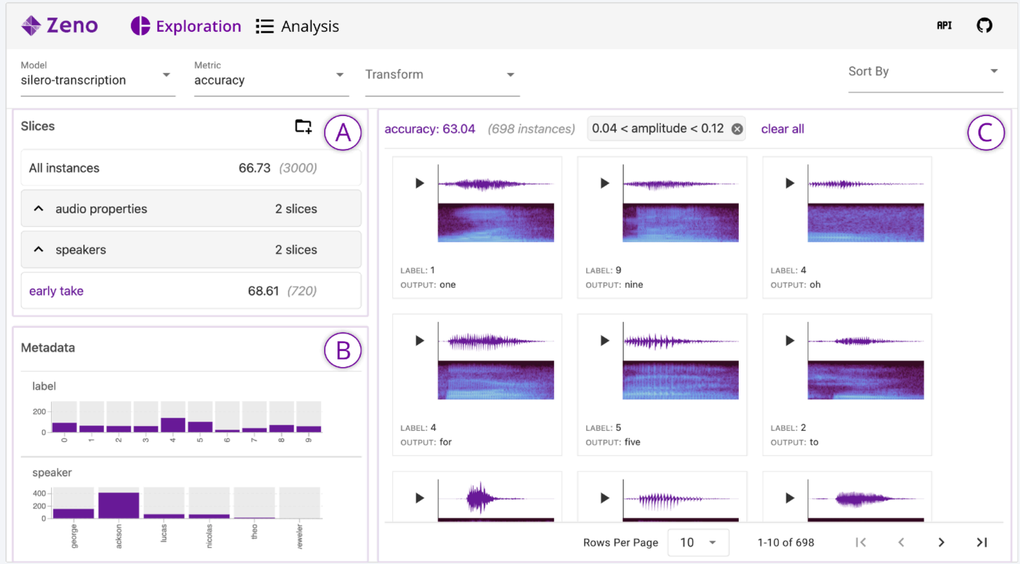
Interactive Framework for Machine Learning Evaluation

# Overview

## Python API

```
@distill
def brightness(df, ops):
  return [lum(r) for r in df[ops.data]]
@metric
def accuracy(df, ops):
  return acc(df[ops.output])
@transform
def rotate(df, ops):
  return [rot(r) for r in df[ops.data]]
```

## Interactive UI



# Python API

- An expressive Python API for **defining evaluation functions**
- Can define a broad range of behaviors, from bias checks to adversarial attacks
- Code-based functions also encourage **reuse and sharing** of tests within and across ML teams

@predict

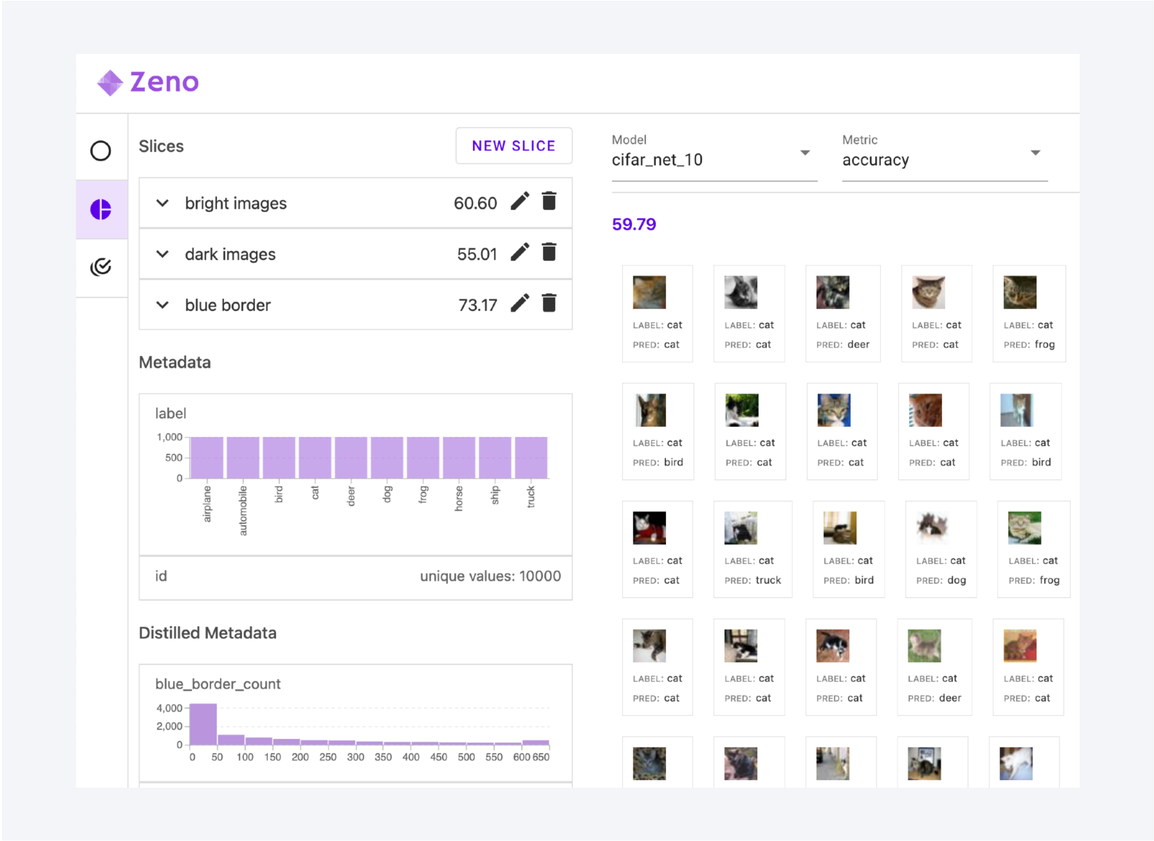
@metric

@distill

@transform

# Interactive UI

- Interactively slice dataset and explore model behaviors
- A modular instance view supports numerous data types and can be extended for specific use cases
- Does not require programming and can be used by diverse stakeholders in cross-functional teams



```
git clone
http://github.com/zeno-ml/example-cifar

pip install -r requirements.txt

zeno zeno.toml
```

# Demo

# Zeno

zenoml.com cabrera@cmu.edu

# Questions?