Machine Learning in Production

Capability Testing of ML Models

Son Nguyen, Ph.D.

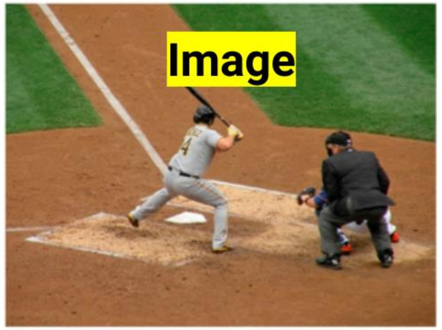
sonnguyen@vnu.edu.vn

Intelligent Software Engineering Group

Software Engineering Department

Programming vs Machine Learning

Programming vs Machine Learning

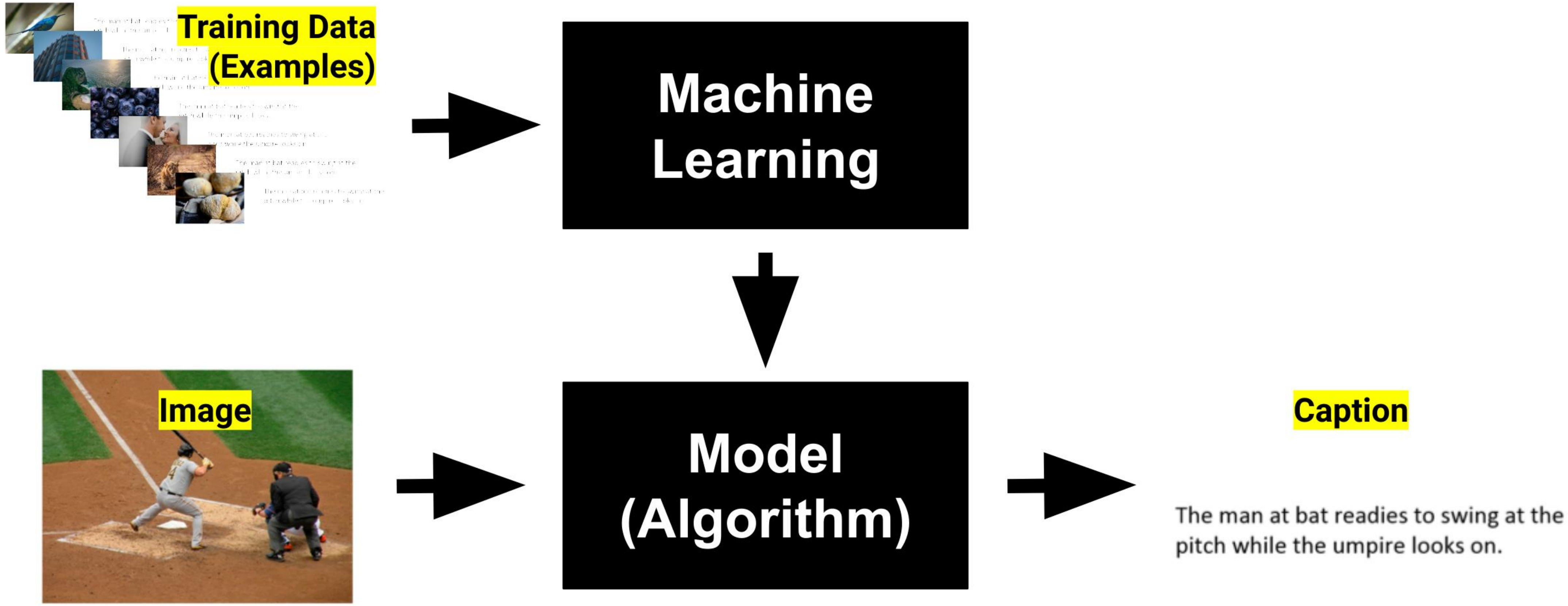


Algorithm

Caption

The man at bat readies to swing at the pitch while the umpire looks on.

Programming vs Machine Learning




THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Evaluating Programs
Evaluating Models

In software implementations,
what is correctness?

Programs’ correctness?

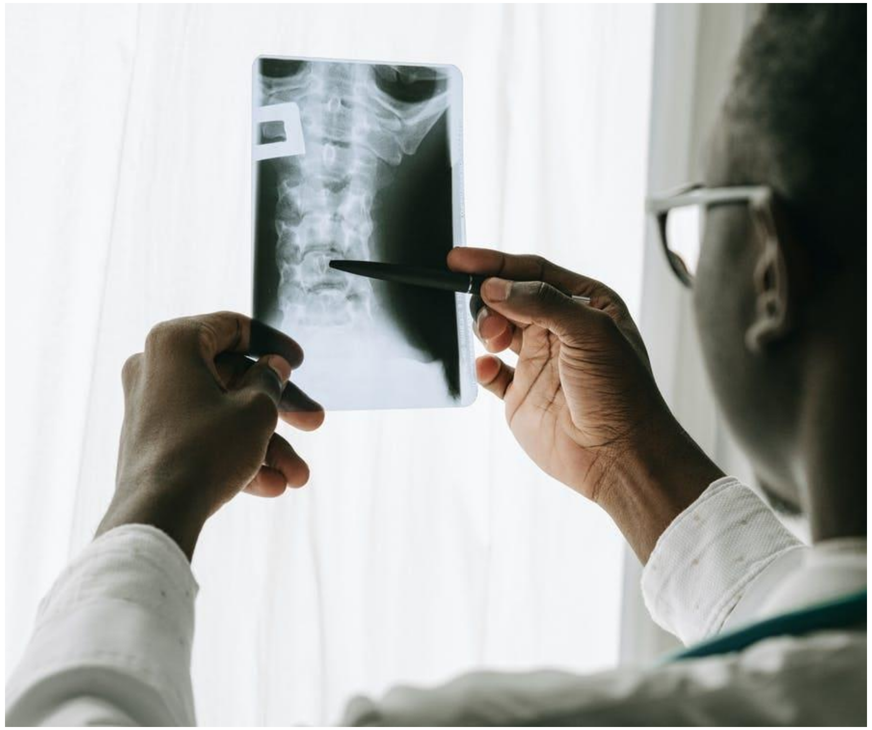
```
/**
* Given a year, a month (range 1–12), and a day (1–31),
* the function returns the date of the following calendar day
* in the Gregorian calendar as a triple of year, month, and day.
* Throws InvalidInputException for inputs that are not valid dates.
*/
def nextDate(year: Int, month: Int, day: Int) = …
```

Analyze

Create tests

```
@Test
void testNextDate() {
        assert nextDate(2010, 8, 20) == (2010, 8, 21);
        assert nextDate(2024, 7, 15) == (2024, 7, 16);
        assert nextDate(2011, 10, 27) == (2011, 10, 28);
        assert nextDate(2024, 5, 4) == (2024, 5, 5);
        assert nextDate(2013, 8, 27) == (2013, 8, 28);
        assert nextDate(2010, 2, 30) throws InvalidInputException;
}
```

Case Study: Cancer Prognosis

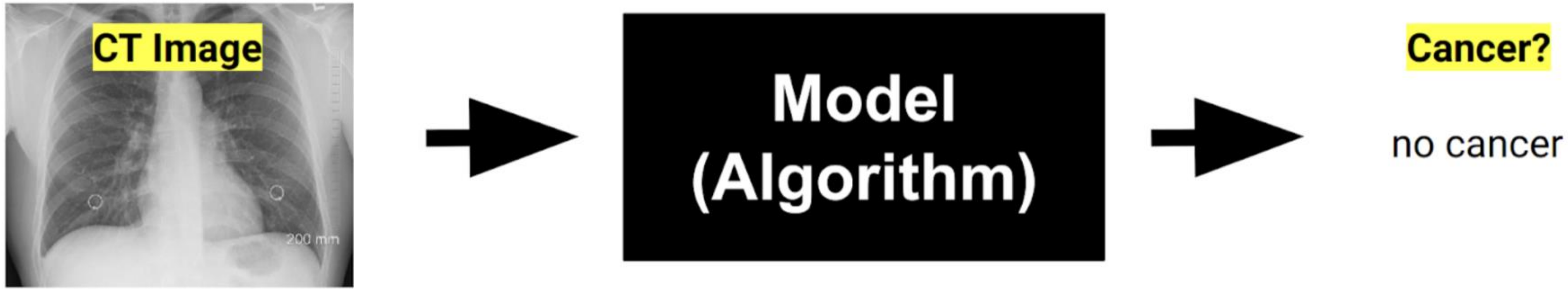


We should stop training radiologists now. It's just completely obvious that within five years, deep learning is going to do better than radiologists.

Geoffrey Hinton, 2016

```
function hasCancer(image: byte[][], age: int, …): boolean
```

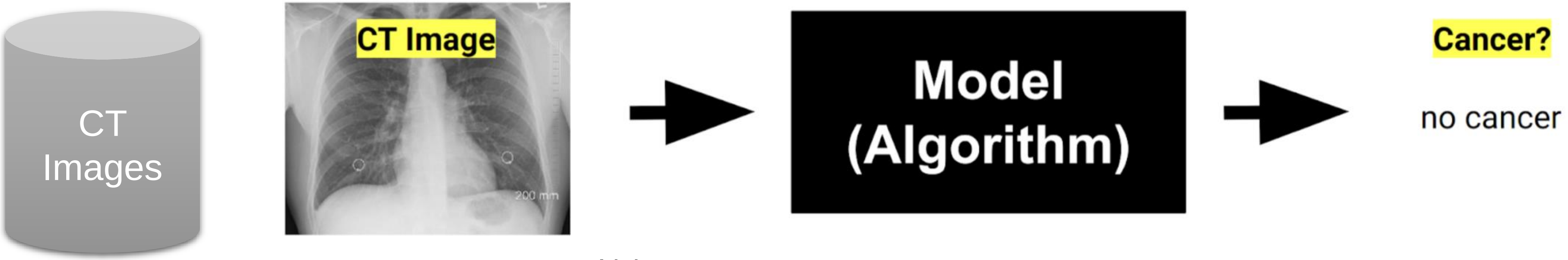
What is models' correctness?




```
/**
* ????
*/
function hasCancer(scan: byte[][]): boolean;
```

We have **no** such specifications!

In ML, what is correctness?




```
/**
* ????
*/
function hasCancer(scan: byte[][]): boolean;
```

Randomly
sample

```
@Test
def testPatient1() {assertEquals(true, hasCancer(loadImage("patient1.jpg")));}
@Test
def testPatient2() {assertEquals(false, hasCancer(loadImage("patient2.jpg")));}
```

How should I think about evaluating models?

…as *correct* or *wrong* or *buggy*”?

It should be about whether they ***fit*** a problem!

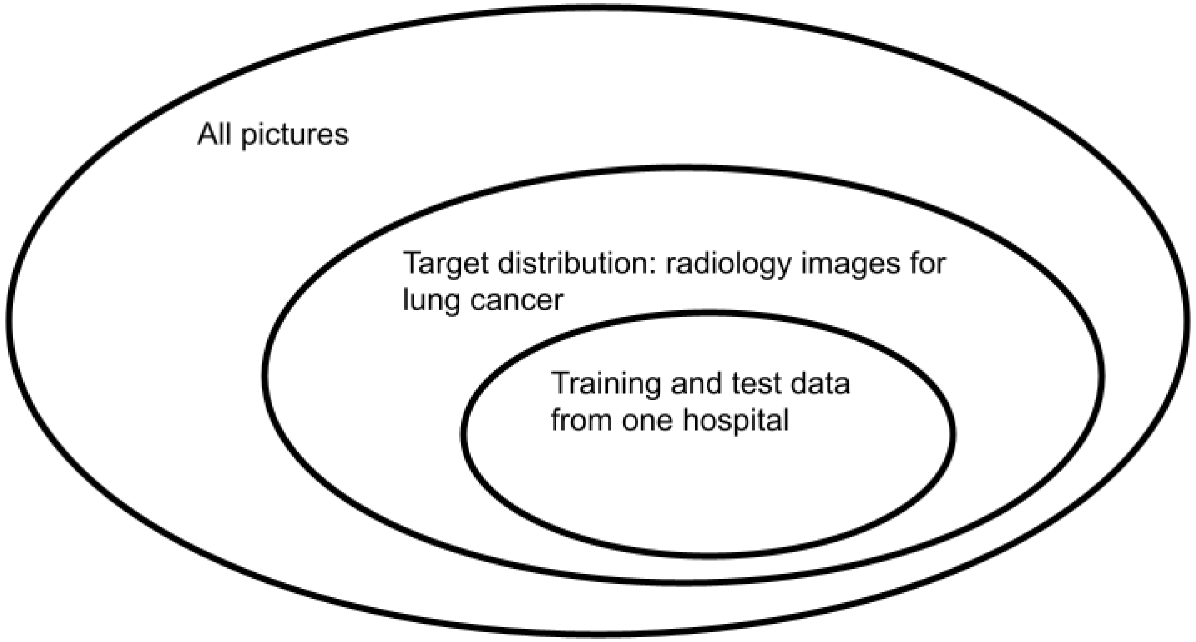
“All models are wrong, but some are useful”

A model, 95% accuracy, may *fit* a problem quite well,
5% of mistakes may be acceptable
for a *useful* solution to a problem

We accept a certain
level of incorrect outputs

Key Assumption in ML Theory and Practice

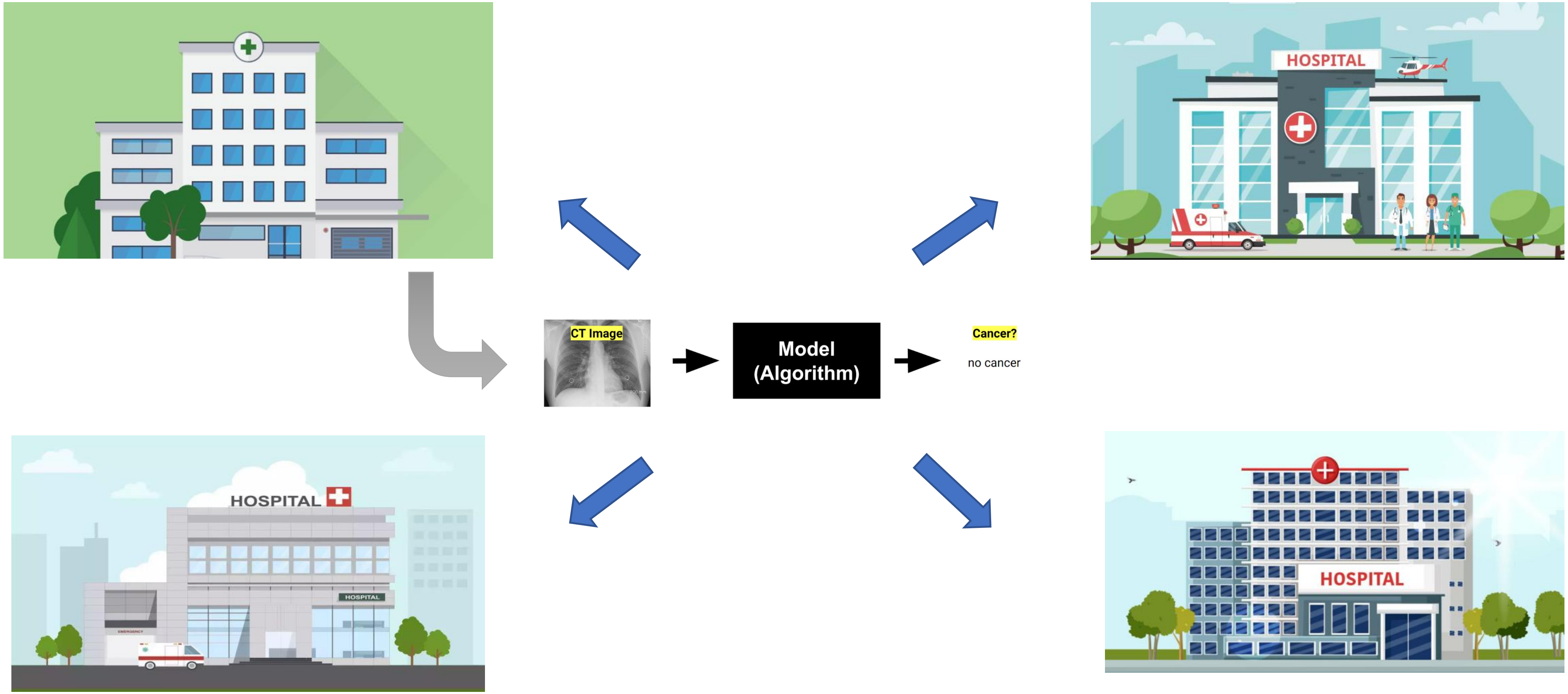
Training and test data
are *independently drawn from the same target population*



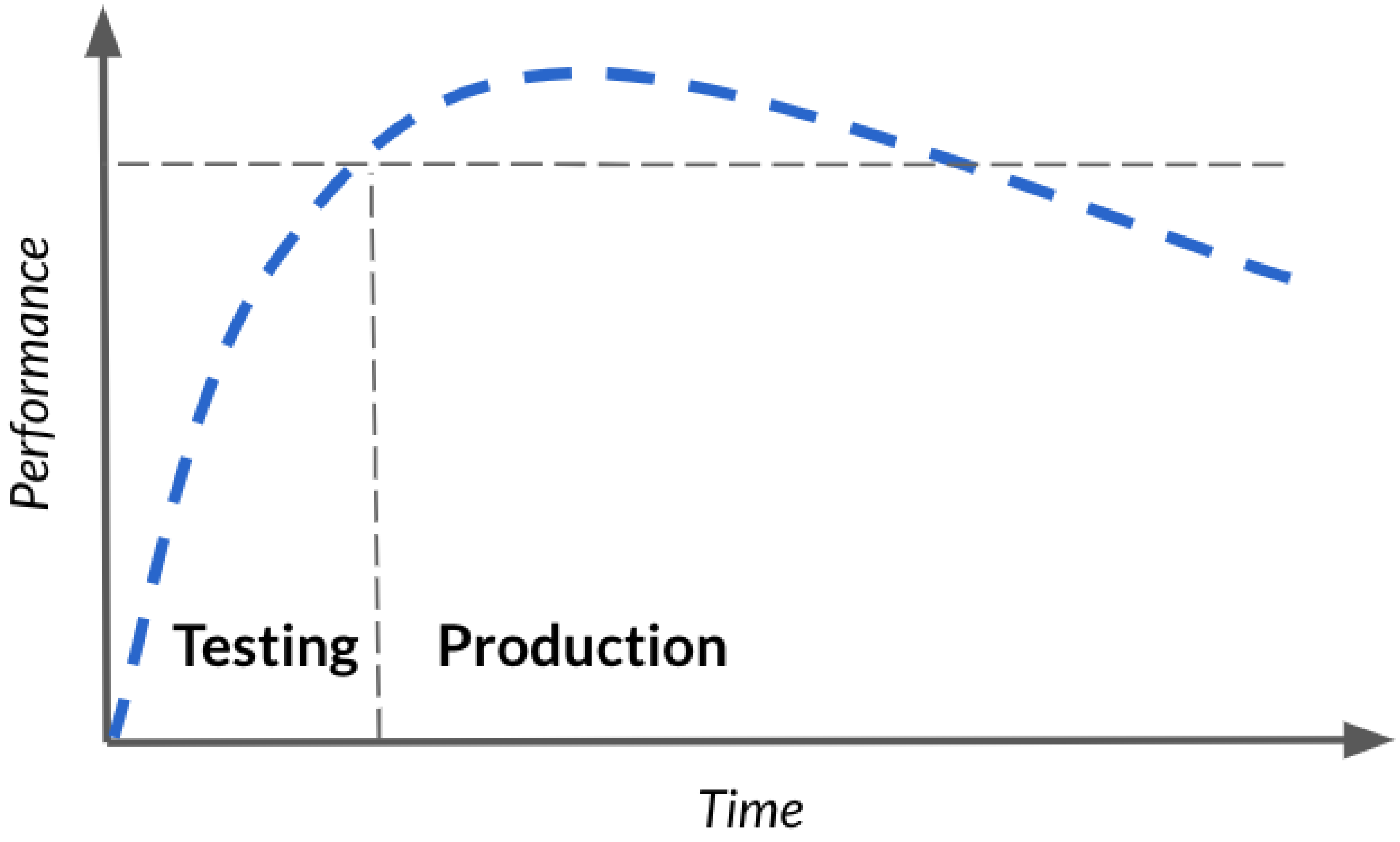
independent and identically distributed (i.i.d)

Our expectation…

Generalizing beyond
the training distribution

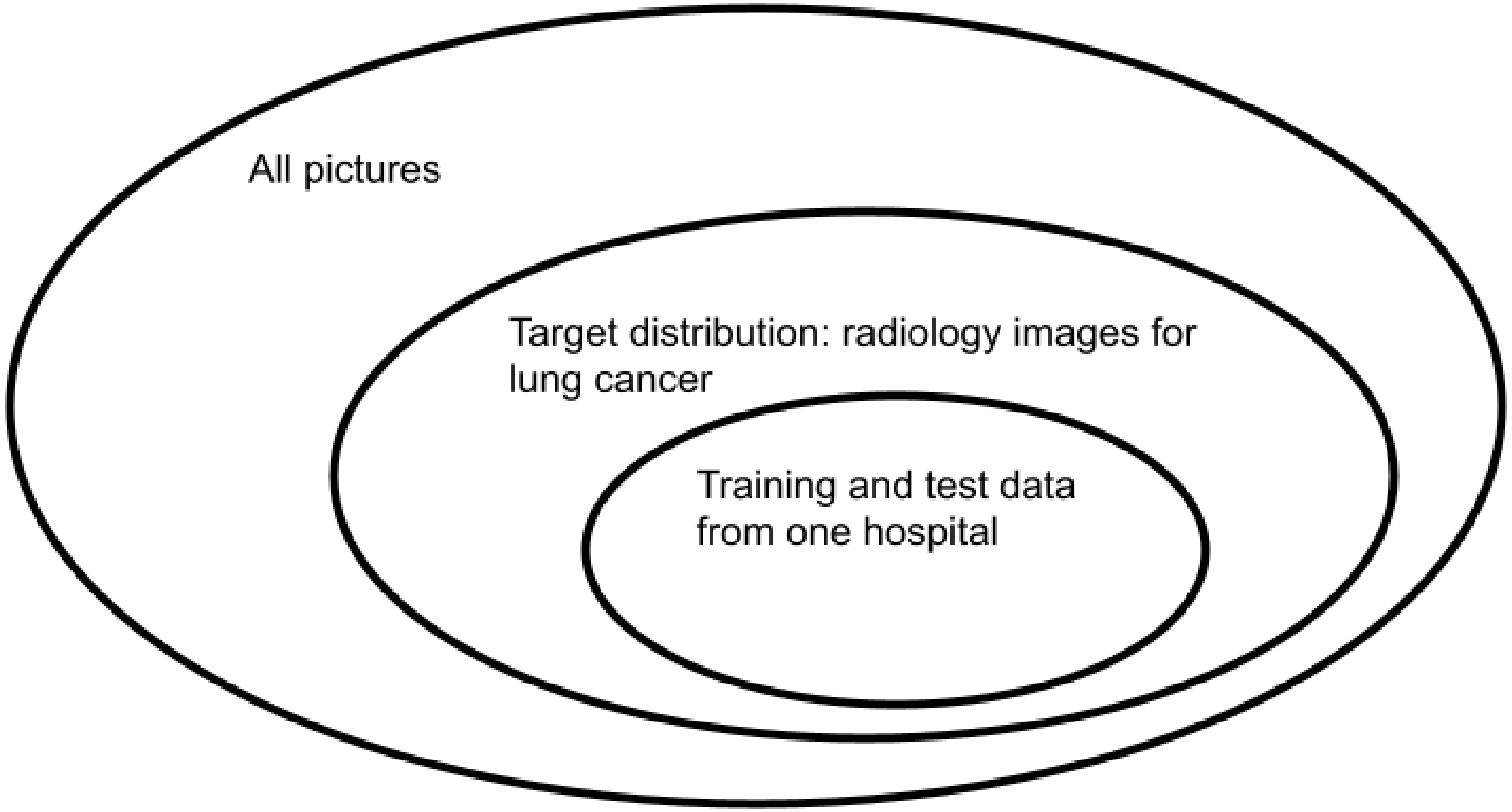

HOSPITAL
CT Image
Model
(Algorithm)
Cancer?
no cancer
HOSPITAL
HOSPITAL
HOSPITAL
HOSPITAL

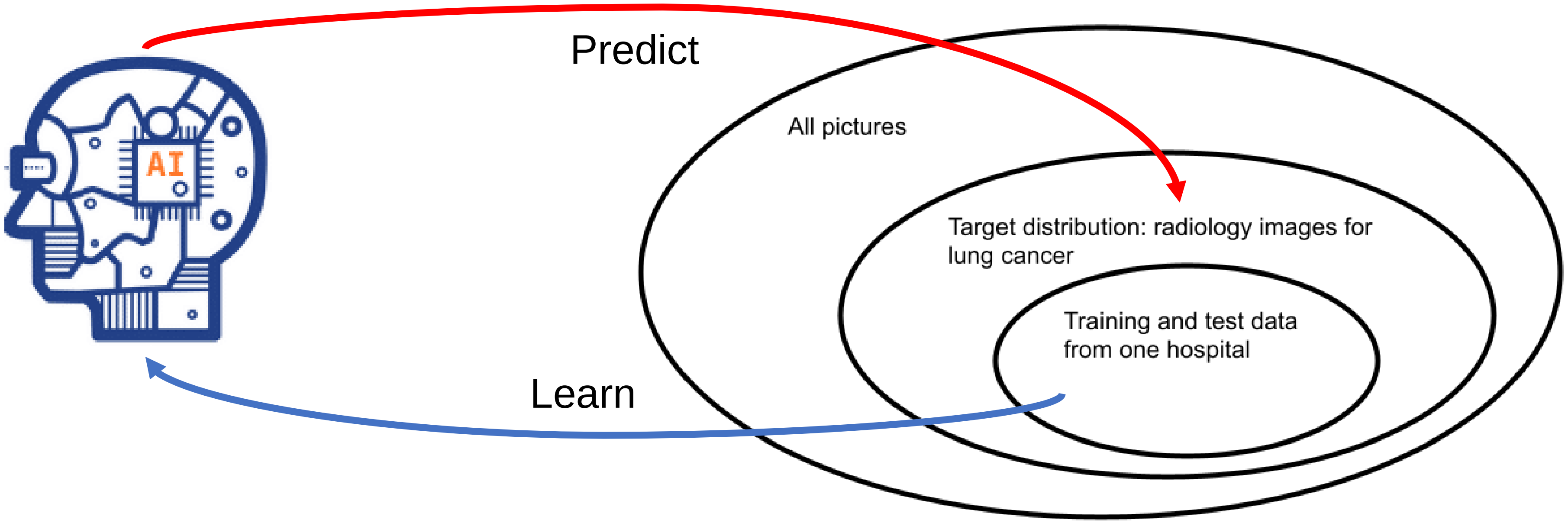
Model Performance Drifts over Time

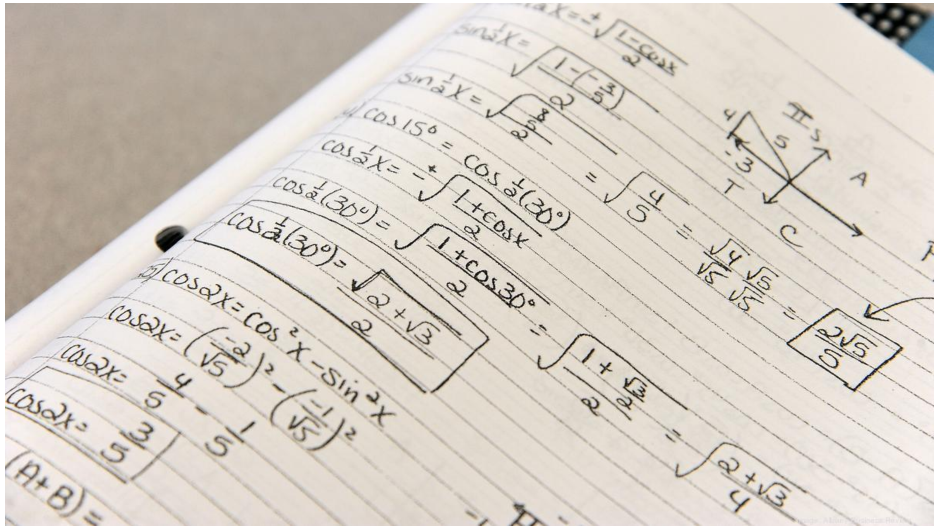


NeuralTalk2: A flock of birds flying in the air
Microsoft Azure: A group of giraffe standing next to a tree

Out-of-distribution Problem




Predict
AI
All pictures
Target distribution: radiology images for lung cancer
Training and test data from one hospital
Learn



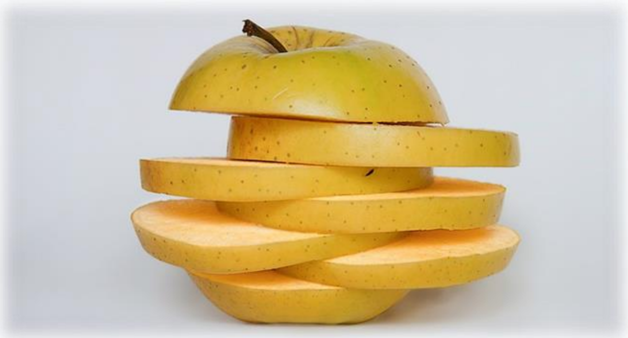
Our reactions?

Generalizing beyond the training distribution

Distribution shifts

Potential bias
in training data collection

Adversarial attacks



Slicing test data

Not All Inputs are Equal: Frequent Cases

"Call mom" "What's the weather tomorrow?" "Add asafetida to my shopping list"

Not All Inputs are Equal: Edge Cases

How do you identify Important Inputs?

Identify Important Inputs

- Curate Validation Data for Specific Problems and Subpopulations:
  - Important inputs ("call mom") -- expect very high accuracy
  - closest equivalent to unit tests
  - Different subpopulations (e.g., accents) -- expect comparable accuracy
  - Challenging cases or stretch goals -- accept lower accuracy
- Derive from requirements, experts, user feedback, expected problems etc.
- Guide testing by identifying groups and analyzing accuracy of subgroups
- Slice test data by population criteria, also evaluate interactions

IBM work: sentiment analysis on reviews from IMDB

DECADE	SUPPORT	ACC
1910s	38	78.94
1930s	338	87.87
1990s	3007	90.95
2000s	6192	91.40

MAIN_GENRE	RAT_CAT	LEN_CAT	SUPPORT	ACC
Mystery	OK	long	11	72.72
Fantasy	OK	short	36	77.77
Crime	OK	long	100	81.00
Comedy	GOOD	long	55	96.36

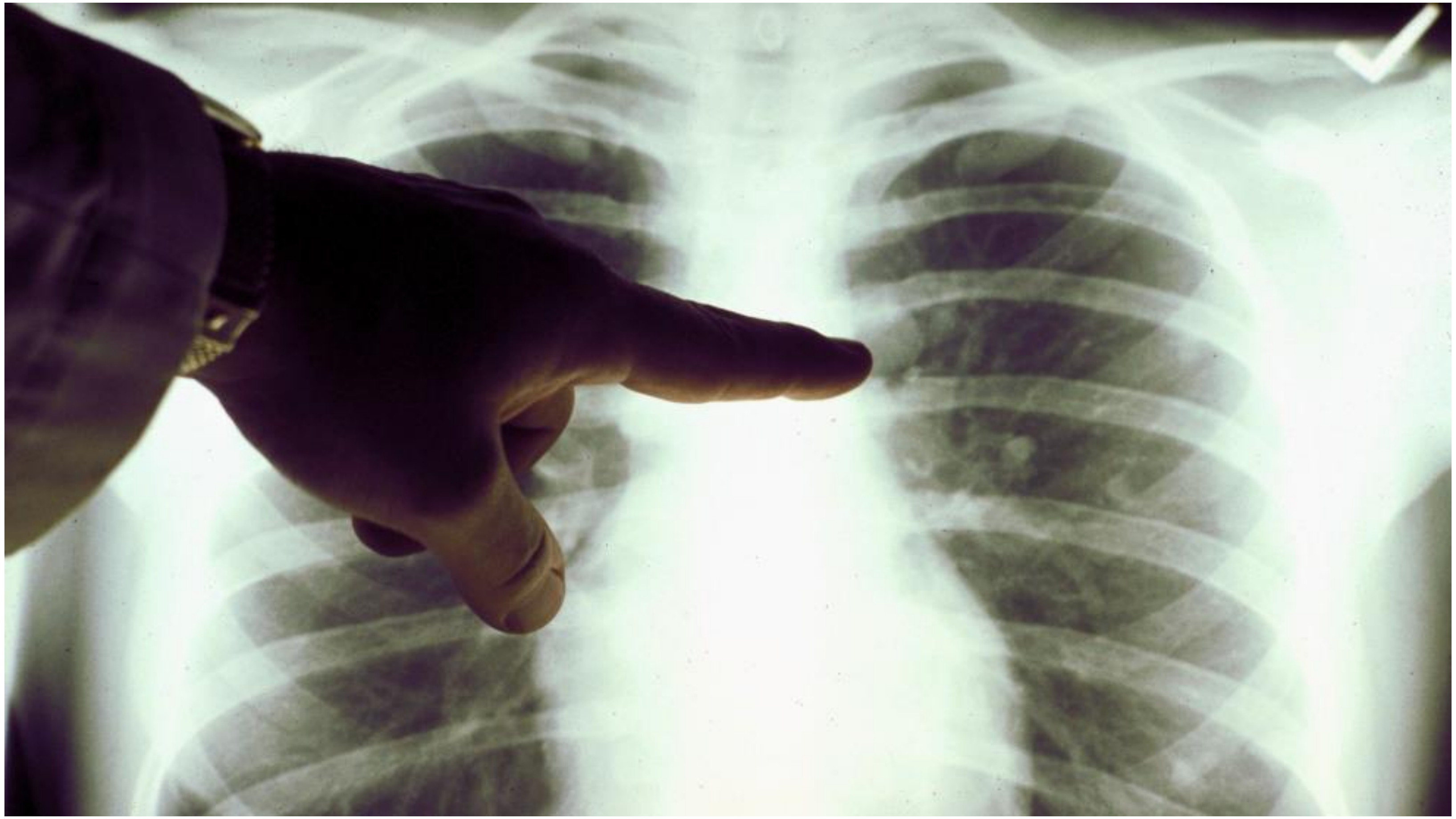
Voice Assistants?

Slicing data: Fairness in ML

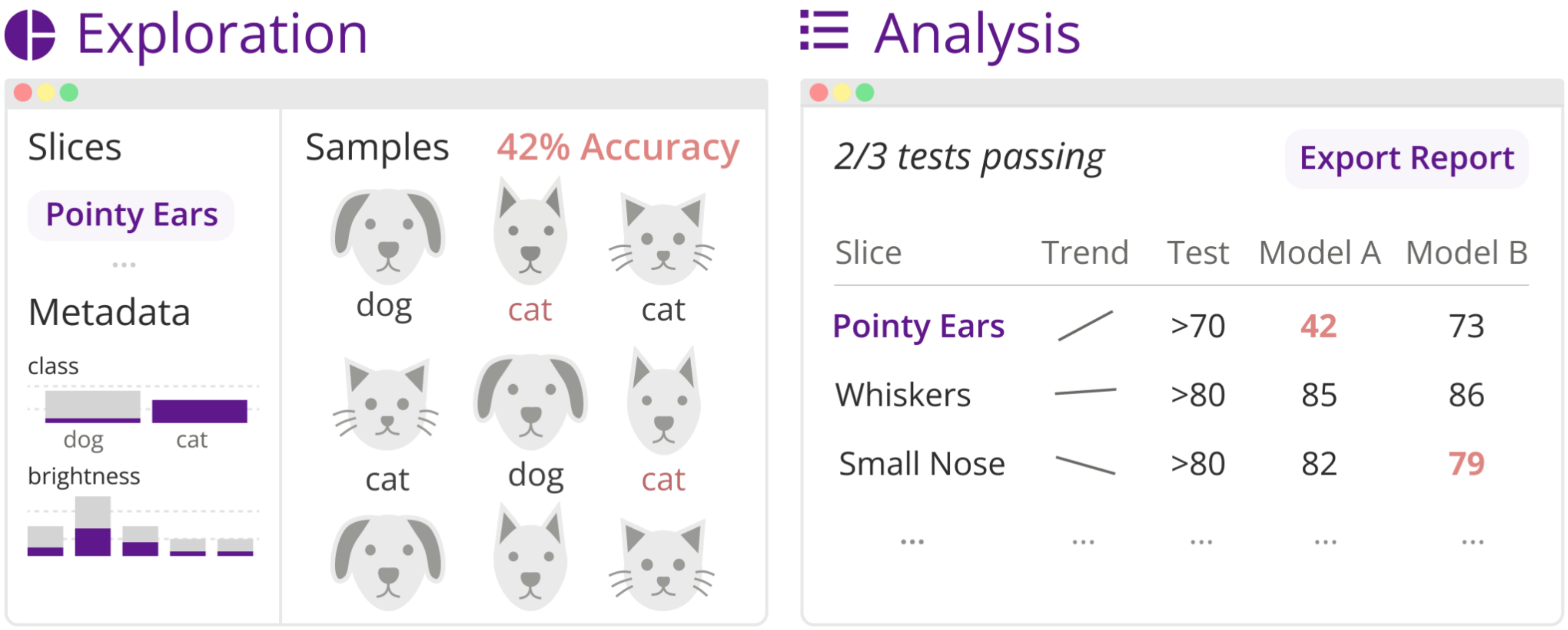
Unfair ML Model

Fair ML Model

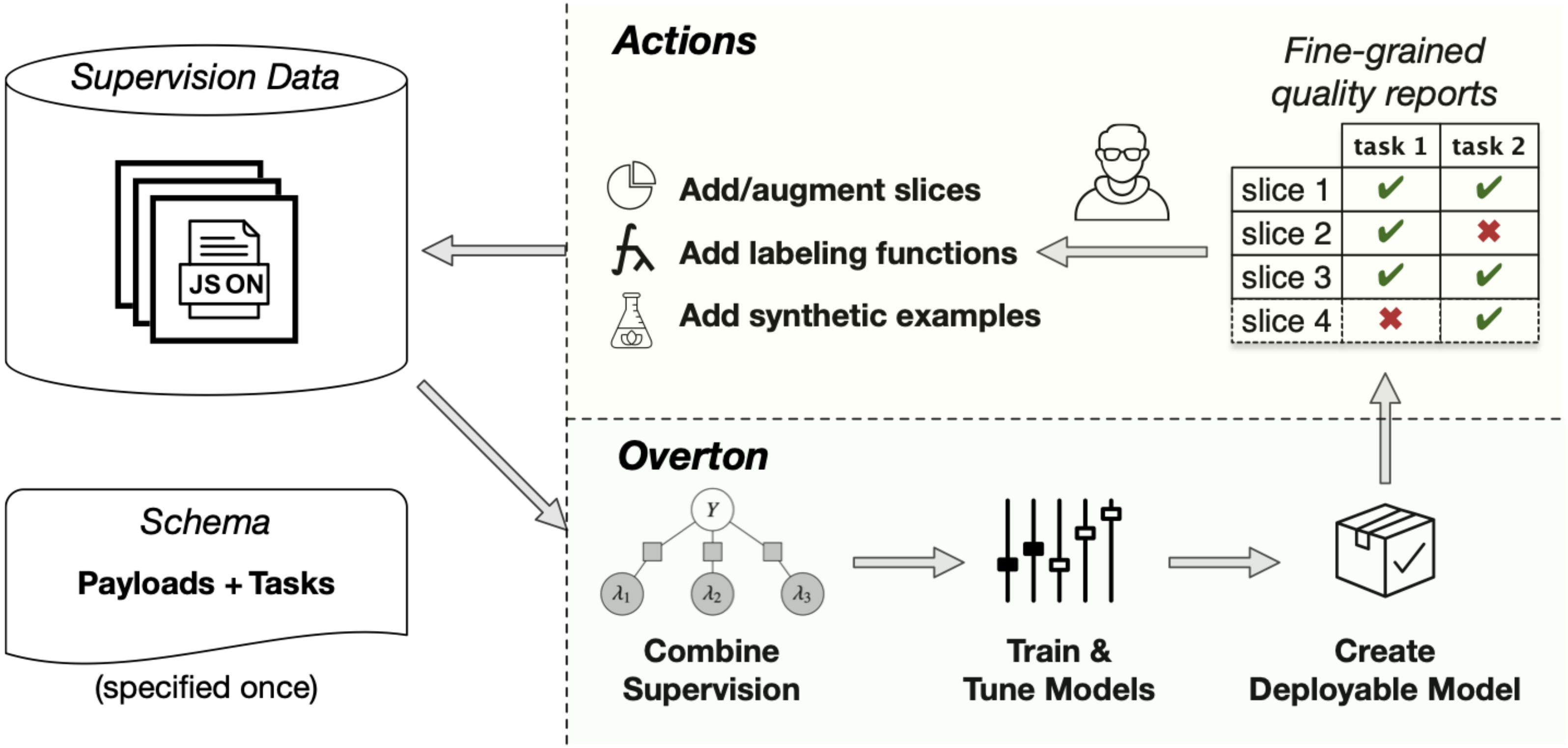
How to slice evaluation data for cancer prognosis?



Multiple slices on image recognition, and model comparison



Overton system at Apple



Slicing test data: Why?

- Identifying problems and anticipating potential problems
  - More tractable problem of improving accuracy for individual slices, rather than trying to improve for the entire input space at once.
- Encourage a team to plan for mitigations:
  - Collecting more training and test data for neglected subpopulations
  - Adjusting how the system uses predictions for these subpopulations to compensate for the reduced confidence

Capability Testing

Capability testing

- **Capability testing:** test whether model can learn key capabilities that humans found essential for solving a task
  → Better mirror human strategies for solving a problem
- Capabilities are inherently domain-specific

This course is quite fantastic 

"Oh great, the battery life on this phone is amazing! It lasts a whole 5 minutes."

Object detection

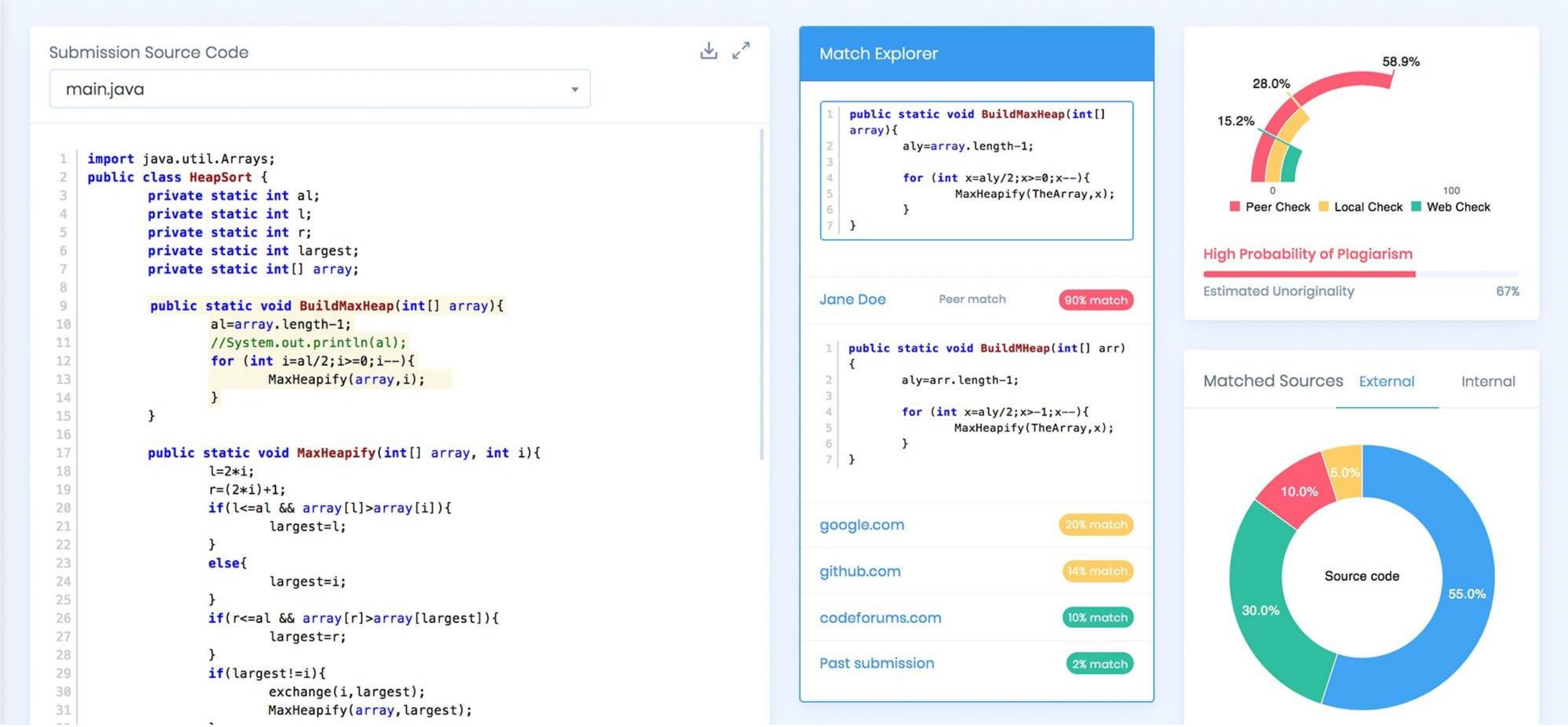
(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

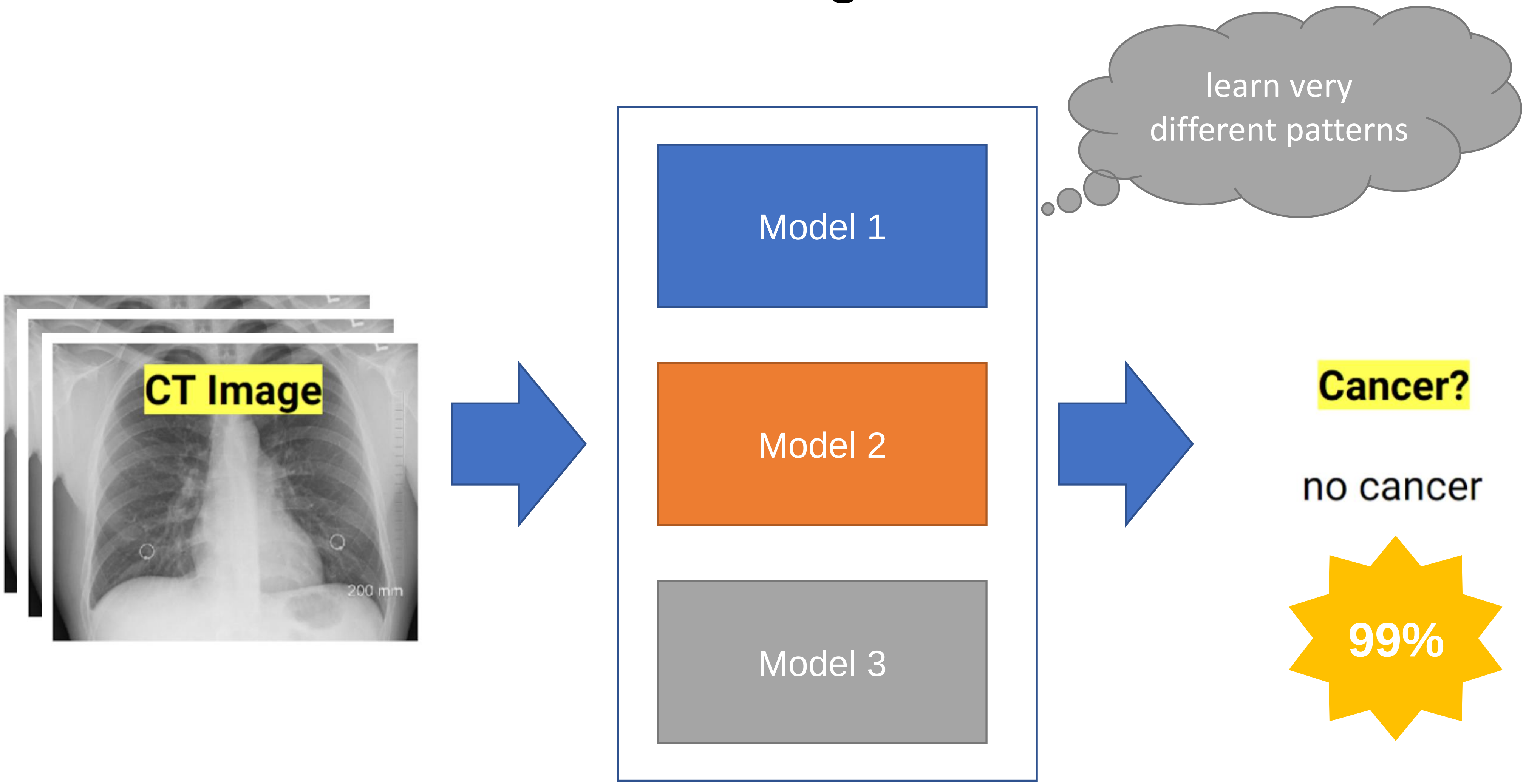
(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

Question-Answering

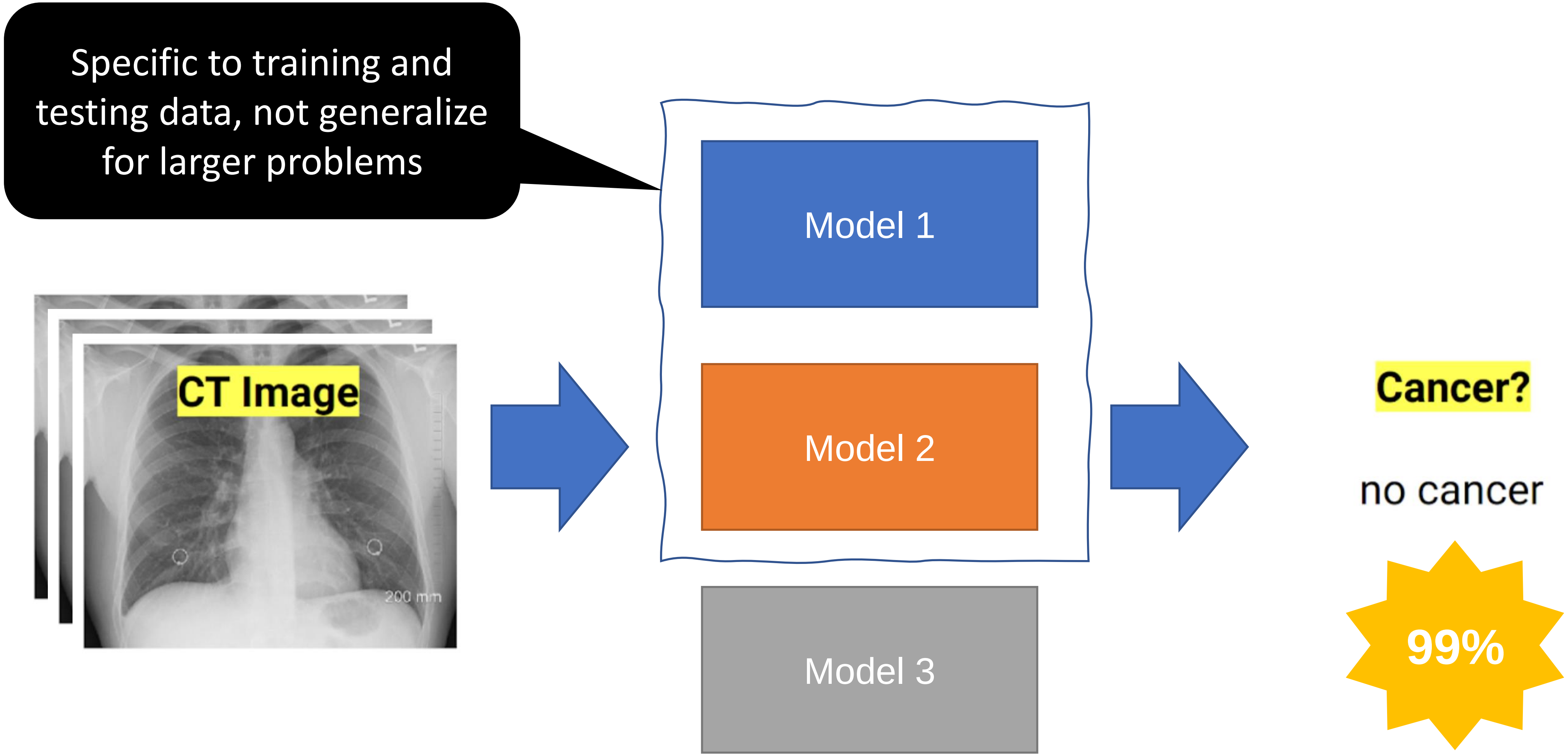
Capabilities of Code Plagiarism Checker?



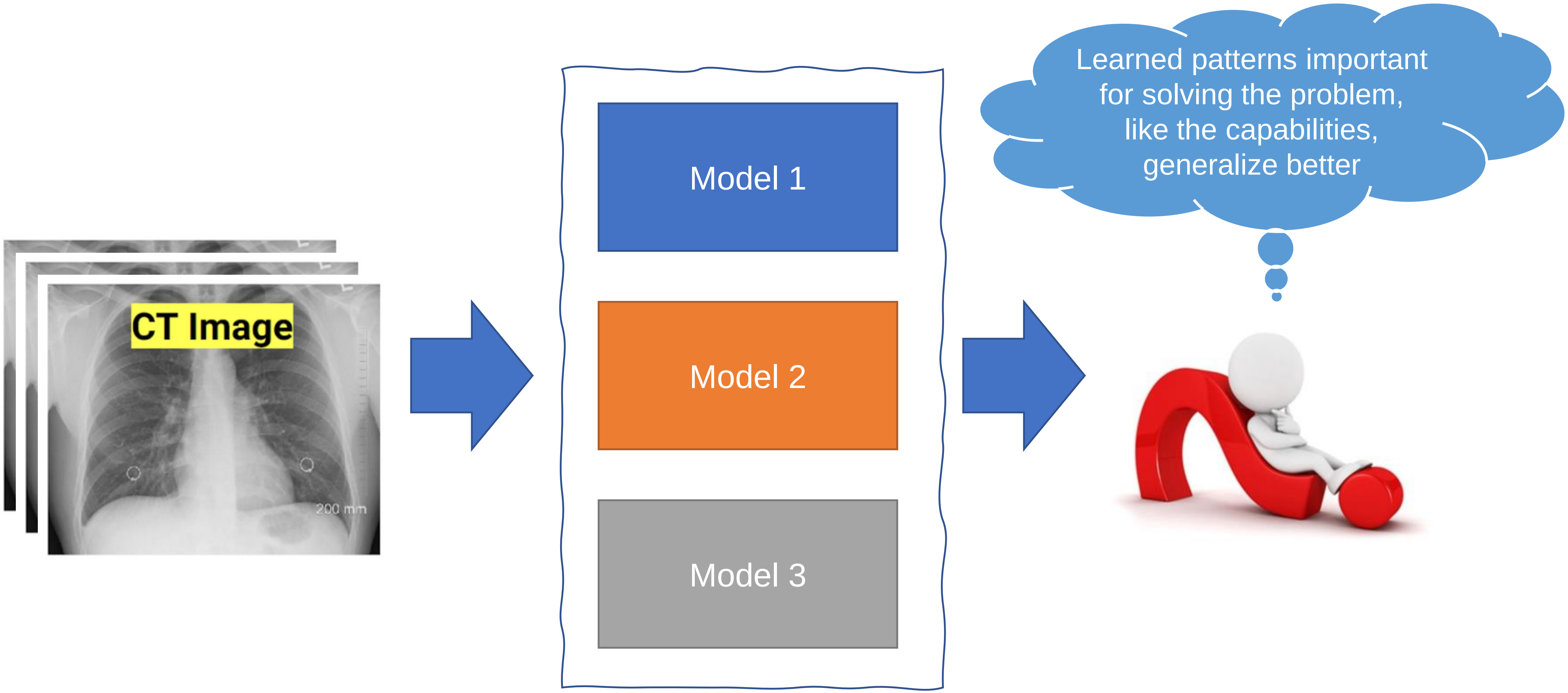
Some arguments



Some arguments

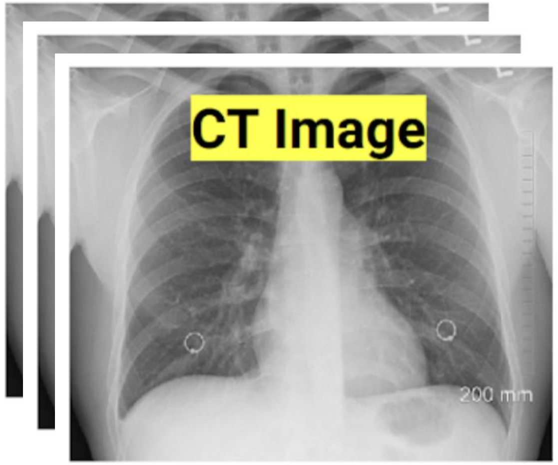


Some arguments



Some arguments

Specific to training and testing data, not generalize for larger problems



Model 1

Model 2

Model 3

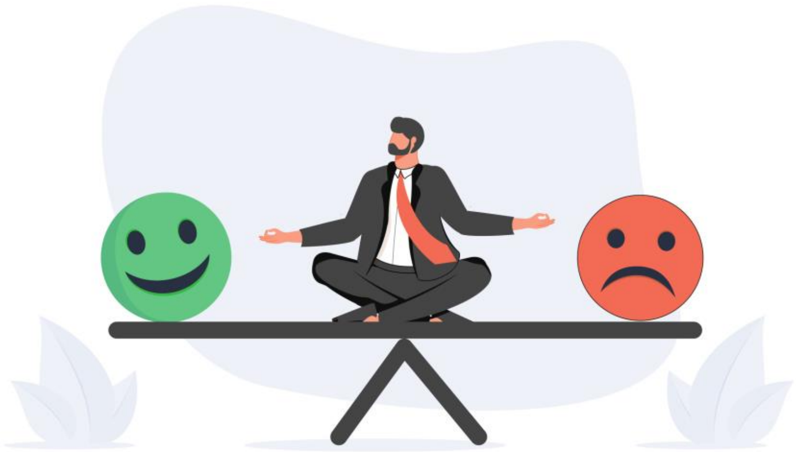
Cancer?

no cancer

99%

Capability Testing Strategies

Domain-specific example generators

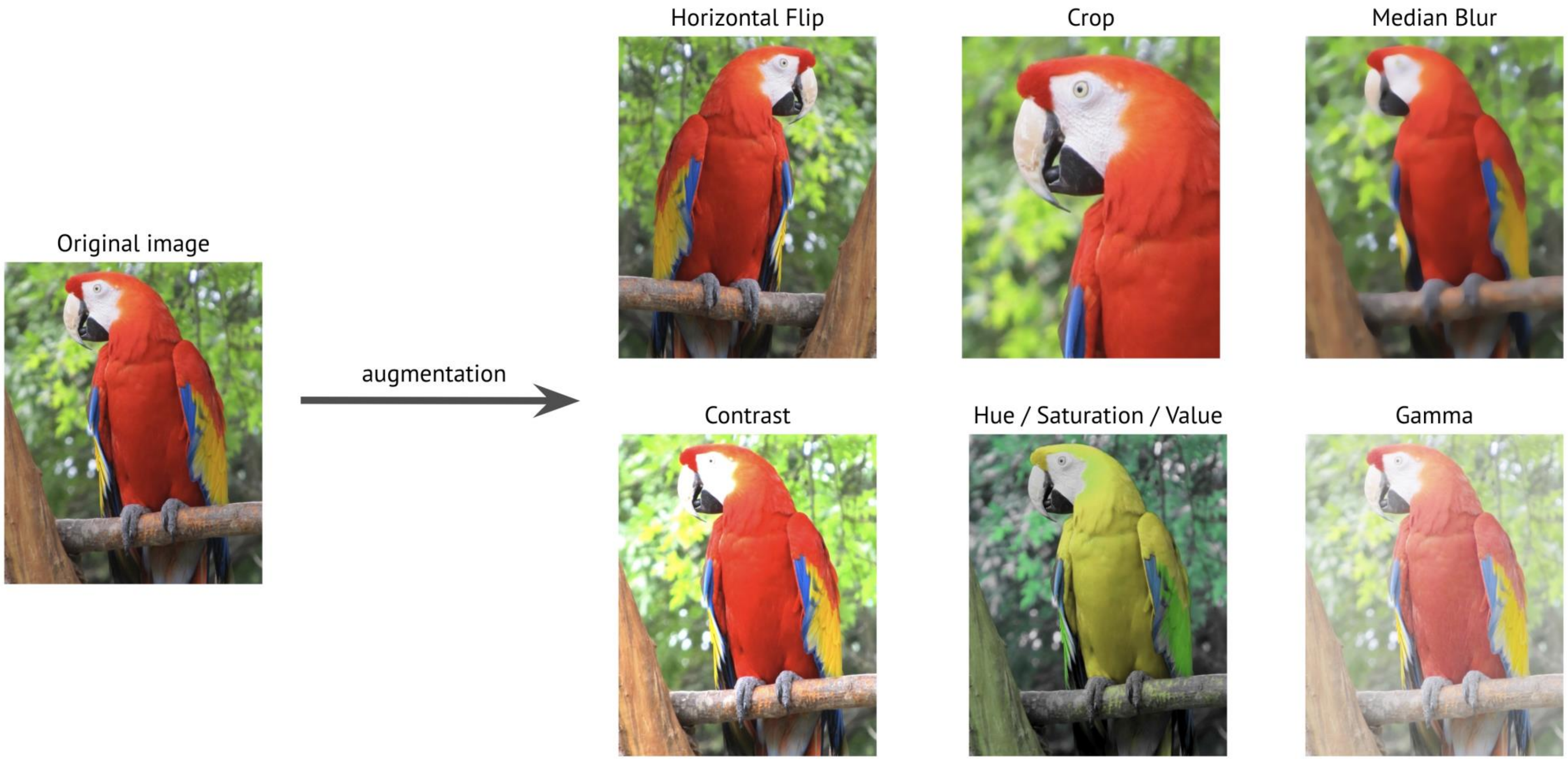


“I love the food”

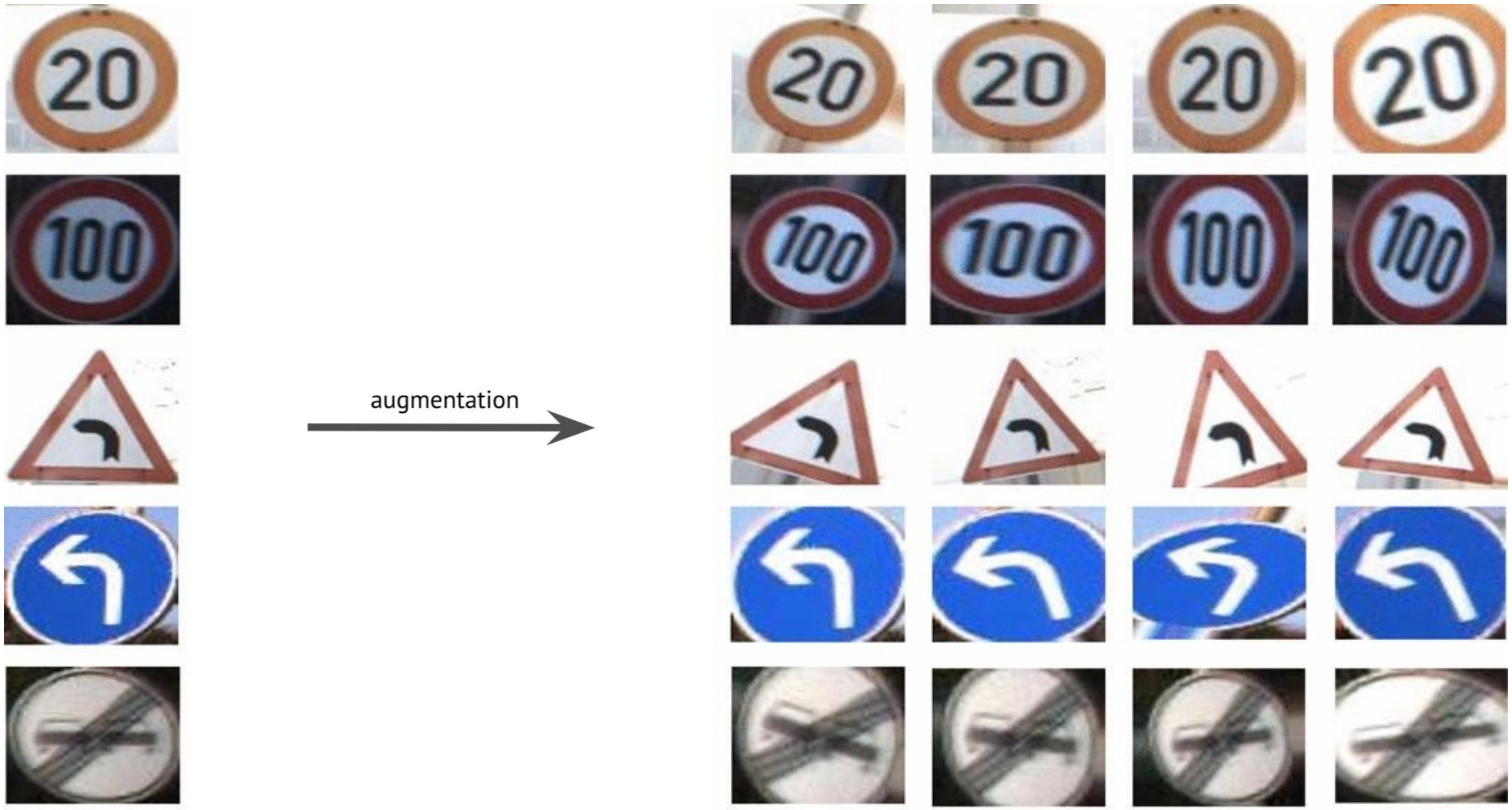
Templates:
“I {NEGATION} {POS_VERB} the {THING}.”
…

“I didn’t love the food”

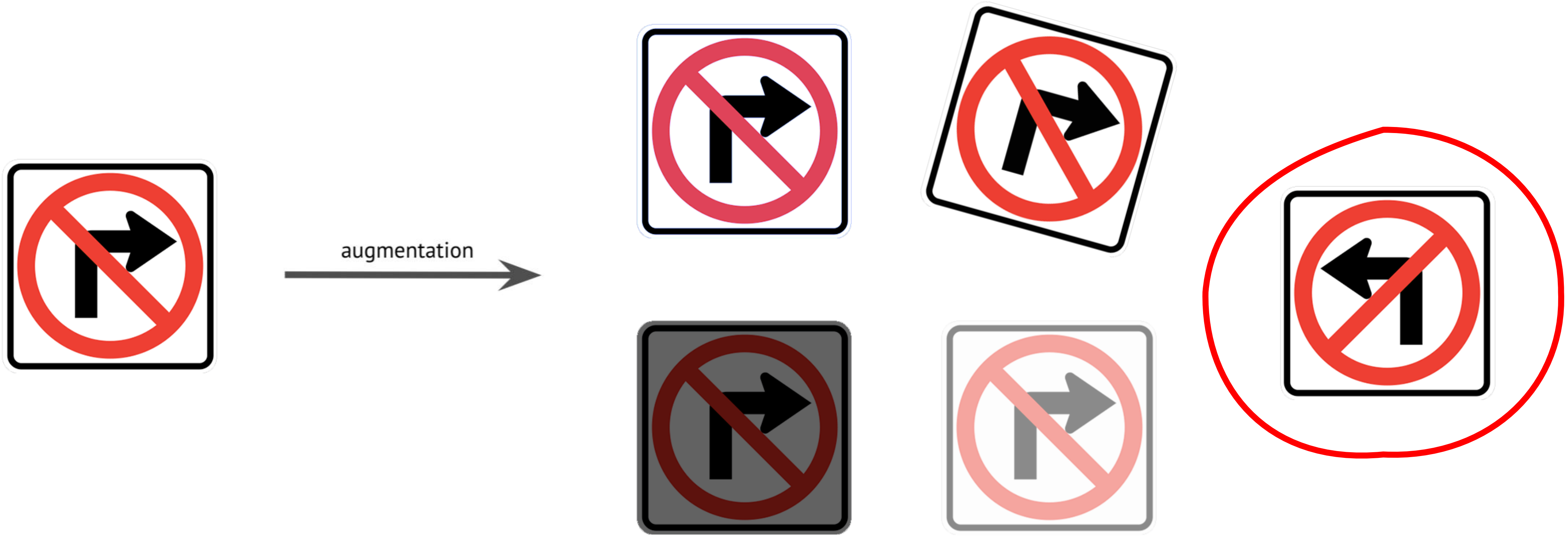
Mutating existing inputs



Mutating existing inputs



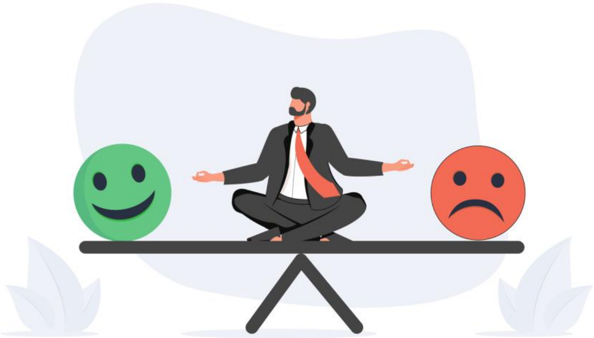
Mutating existing inputs



Original   Mutated   Mutated

Crowd-sourcing test creation

"The battery life on this phone is amazing! It easily lasts all day."

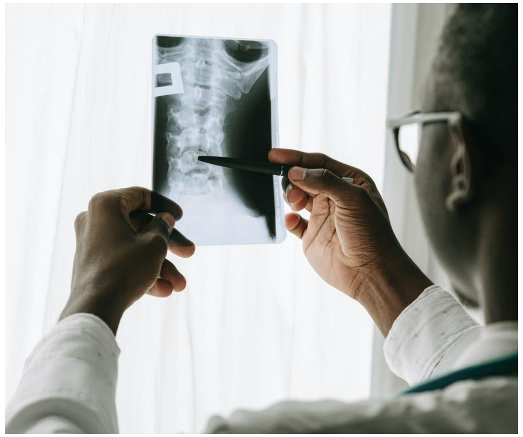


"Oh great, the battery life on this phone is amazing! It lasts a whole 5 minutes."

Identifying capabilities

- Analyzing common (known) mistakes --> making clear candidates for capabilities
  - NLP models using word overlap rather than understanding a text's content
  - Computer vision models focusing on texture over shape
- Using existing knowledge about the problem
  - In NLP, some partial understanding: synonyms, anonyms, identifying named entities, semantic role labeling, negation, and coreference.
- Observe humans
  - Where we do not have domain knowledge, we can study how humans solve a problem
- Derived from requirements (typically invariants)
  - E.g., Fairness requirements
  - Credit risk prediction shall not differ depending on gender/color
  - Changing person names should not affect the sentiment of the text

One final look…



- **Analyzing model mistakes**: The model performs poorly when brightness is not calibrated across multiple scanners
  - Capabilities: ???
- **Observing humans**: Ask radiologists why they disagree with a model
  - Capabilities experts use that the model may be missing
- **Existing knowledge:** Look into non-ML literature on cancer diagnosis
  - Capabilities that radiologists use when looking for cancer in training material for radiologists

Wrap this up…

- Identifying capabilities and then creating test data **is not that different** selecting inputs for unit tests.
- We have no a strong specification for ML problems, we have some knowledge about the problem and past mistakes.
- Identifying capabilities is **not unlike** selecting test inputs for a program without looking at code (**black-box testing**)