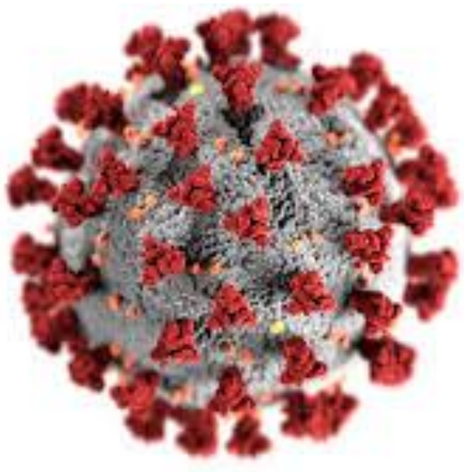
# Accuracy: Beware of Red Herrings and Black Swans

Steve Teig
Perceive
September 2020

# Setting the stage: what is accuracy?

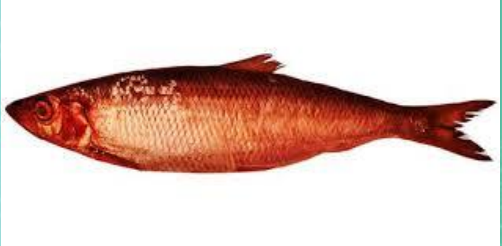
Example 1: COVID-19 testing

- About 1 person in 330 (i.e., 0.3%) worldwide has tested positive for COVID-19
- Build “AI-based test” to answer: “Do you have COVID-19?”



- Test always returns “No!” → 99.7% accurate

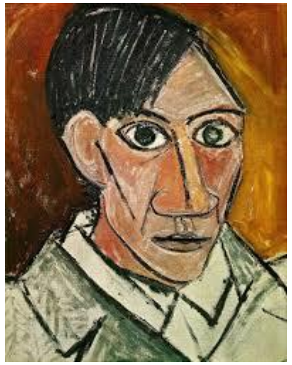
# Setting the stage: what is accuracy?



Example 2: face detection

- Every face in the training set has two eyes on opposite sides of a nose






# Setting the stage: what is accuracy?

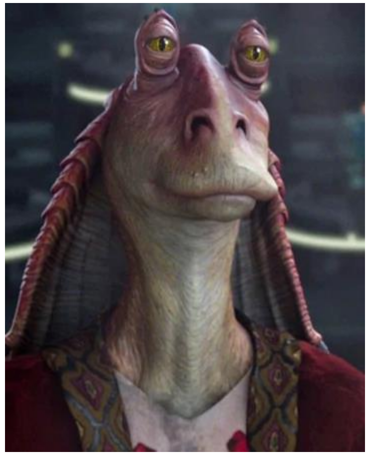
How can we build models we can trust???

What should we be optimizing for?

First tenet of optimization: never lie to the optimizer!

Make sure that if you prefer A to B, so does your loss function.

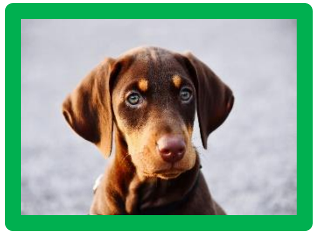
# Which model would you prefer?



95% accuracy!

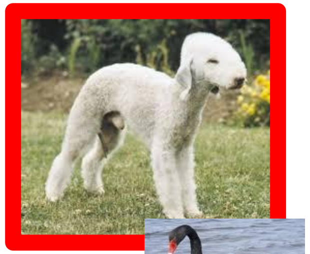
95% accuracy!

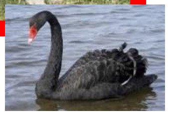
# The dogma of detecting dogs (and other objects)



Precision, recall, and F1

- Precision: true_positives / predicted_positives
  - Intuition: what fraction of the time is the model correct when it labels a picture, "dog"?
- Recall: true_positives / number_of_positives
  - Intuition: what fraction of the dog pictures did the model find?





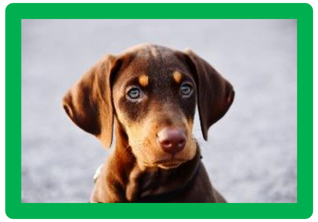
- F1: $2/(\text{precision}^{-1} + \text{recall}^{-1})$
  - Intuition: assume precision and recall matter equally, so take their (harmonic) mean
  - Matter equally??? Why?
  - Could weight them differently. By how much? Per class? Based on what *principle*?

# The dogma of detecting dogs (and other objects), cont.









Pred: “Dog” “Dog” "Dog" “Dog” “Dog”

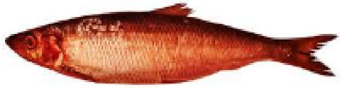
Precision, recall, and F1

- Precision: 3/5. No extra credit for identifying hard-to-detect dogs.
- Recall: 3/4. No extra credit for misidentifying cat as dog – but avoiding the helicopter.
- F1: 2/3. Is that good/bad/other? Why do you think so?

# The mythology of average accuracy

Black swan → incorrect assumption that all mistakes are equally important

- So every mistake contributes equally to the computed quality of the model
- Dog→cat ≠ dog→helicopter



Red herring → incorrect assumption that all data points are equally important

- So every data point contributes equally to the computed quality of the model
- Yet-another-frontal-face ≠ face in profile, face with mask

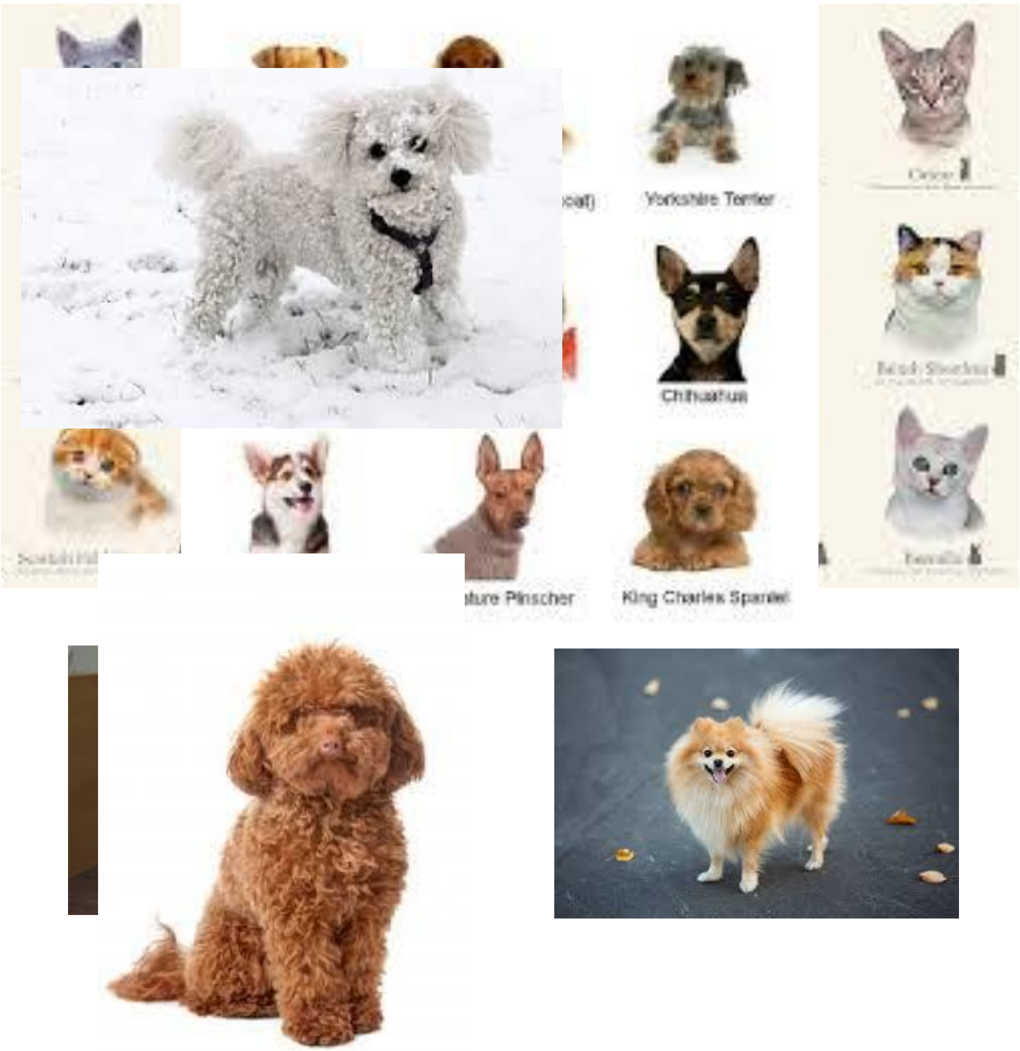
*Some errors (and some data points) matter more than others*

# The F(Law) of Averages

Average accuracy is *almost never* what customers want…

- Even though almost all developers optimize for that!

# Defeating the red herring is harder than it looks

# OK, smarty-pants. Then, what should we do instead?

Weird data points, if real, are some of the most informative

- Pay attention to them!

Weird data points: vital evidence that “surprising” data points exist

- Example: dog with no hair, dog with one ear or three legs or artificial color or…
- Discriminate among the many models that fit the unsurprising points

One can learn a lot even from <u>one</u> black swan

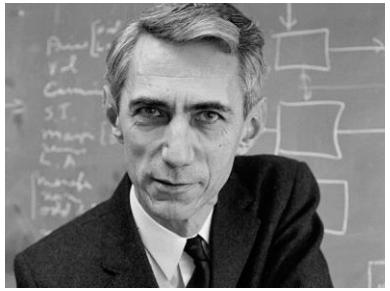
- Even one data point should be able to change your model <u>if it is surprising enough</u>

<u>Every</u> data point is an adversary…

# Surprise!

How can we quantify surprise?

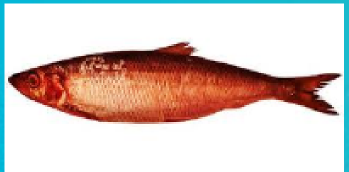
- If event A has probability 1 → surprise(A) = 0
- If prob(A) < prob(B) → surprise(A) > surprise(B)
- If A and B are independent → surprise(A and B) = surprise(A) + surprise(B)



Claude Shannon (1916-2001)

Above criteria → surprise(x) ≡ $-\log_b(\text{prob}(x))$

More bits → more "unusual" → more surprising → more *informative*

# Confronting the  : coincidences in training data

During training, repeatedly estimate informativeness of each datum → how surprising it is

Consider object classification: dog, cat, helicopter, …

Typically, model being trained returns probability distribution over classes

Select datum $d_i$ with probability proportional to its (current) surprise

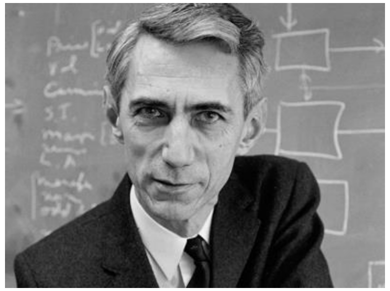
Surprise($d_i$) = -lg(prob($d_i$))

Still need informative, representative dataset… but making the best use of it!

# Entropy

Entropy: $\mathrm{H} \stackrel{\text{def}}{=} -\sum_i p_i * \lg(p_i)$

- $\mathrm{H} = \sum_i p_i * -\lg(p_i) = \frac{(\sum_i p_i * -\lg(p_i))}{\sum_i p_i} = average_i(-\lg(p_i))$



Claude Shannon (1916-2001)

Average number of bits → average surprise

Average throughput of a communications channel

Measure of uncertainty: how many yes/no questions on average would be necessary?

But we are interested in *maximum* surprise, not *average* surprise

# From entropy to "extropy"

Entropy: average number of bits = average surprise

- $\mathrm{H} \stackrel{\text{def}}{=} \sum_i p_i * -\mathrm{lg}(p_i)$
- Intuition: minimize <u>average</u> surprise

Extropy: "maximum" number of bits (or nats or…) = maximum surprise

- $\widehat{H_\alpha} \stackrel{\text{def}}{=} \frac{-1}{\alpha} ln \sum_i e^{\alpha p_i} = softmax_i(-\ln(p_i))$
- Intuition: minimize <u>maximum</u> surprise → make a big mistake as rarely as possible

# Confronting the  : making big mistakes

Goal of accuracy : maximize predictiveness of model

Minimize maximum surprise as a proxy

Minimize extropy – “maximum” of surprise – instead of entropy

Minimize cross-extropy instead of cross-entropy

Extra credit: *maximize minimum surprise* between classes in object classification

Average accuracy might go up or down – but <u>likelihood of big mistakes will go down</u>

# Summary

Red herrings: coincidences in training data lead to errors in prediction

- Importance of a training datum should be proportional to its surprise

Black swans: some errors are much more severe than others

- Importance of a training datum should be proportional to its surprise

Red herrings and black swans are different aspects of the same underlying problem

Change of assumptions: averages should have little place in machine learning

Accuracy → build predictive models → minimize maximum surprise

- Select data points for training in proportion to how surprising they are
- Select loss functions that minimize extropy, not entropy

# Resources

## More information

Precision, recall, and F1

https://en.wikipedia.org/wiki/Precision_and_recall

Extropy: approximates Rényi min-entropy

https://en.wikipedia.org/wiki/R%C3%A9nyi_entropy

Perceive

https://www.perceive.io

## 2020 Embedded Vision Summit

**"Ergo™: Perceive's ultra-low power inference chip provides data center-class inference inside edge devices"**

Thursday, September 17, 9:30 AM PDT