

# **CrossTALK: ML Crash Course**

**Q1 2025**

**Chris J. Maddison**

# Agenda

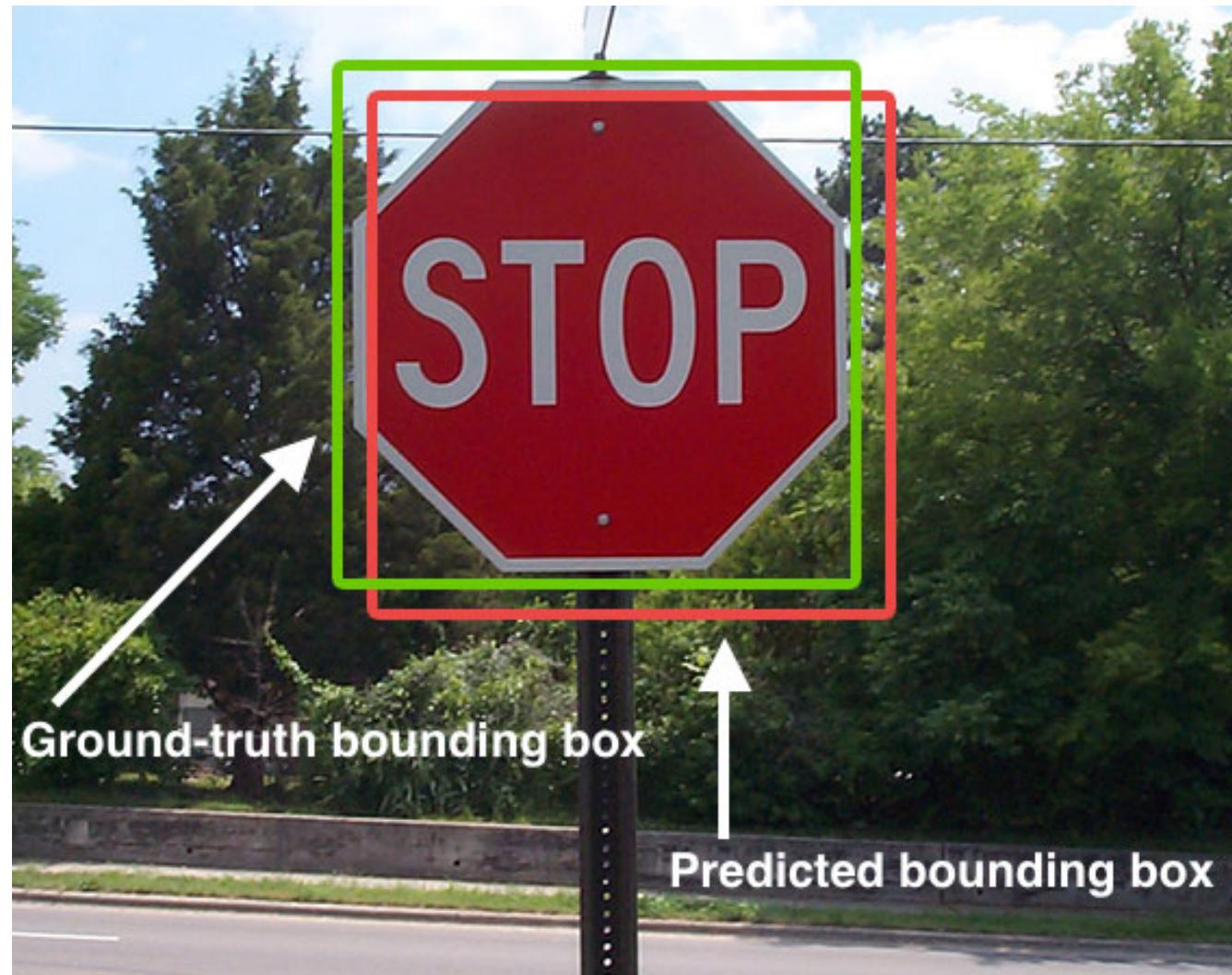
- The Story of A Single Bit
- Prediction, Learning, Conditional Prediction
- This Bootcamp

# The Story of A Single Bit

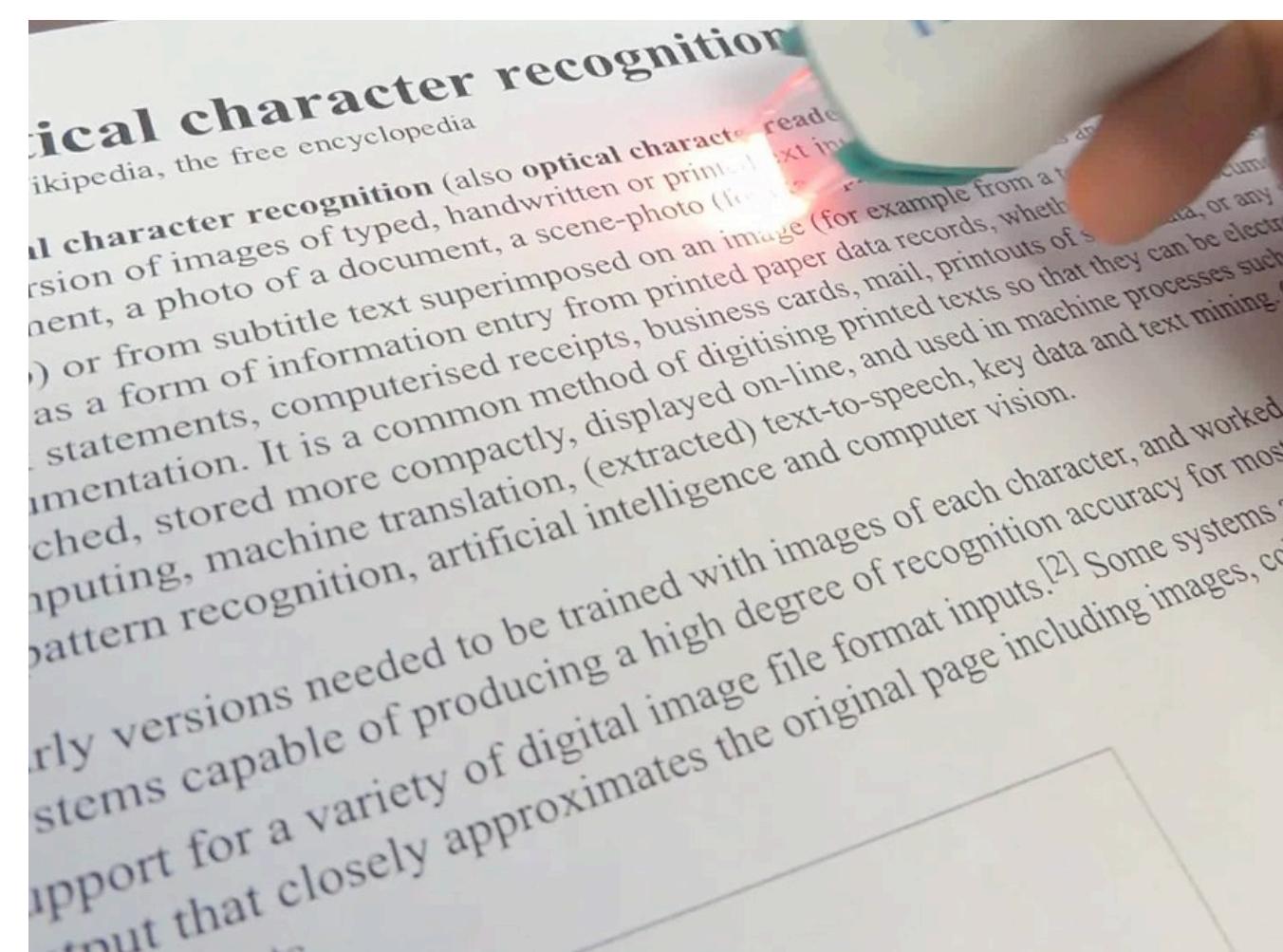
# Machine learning: algorithms for prediction

For example,

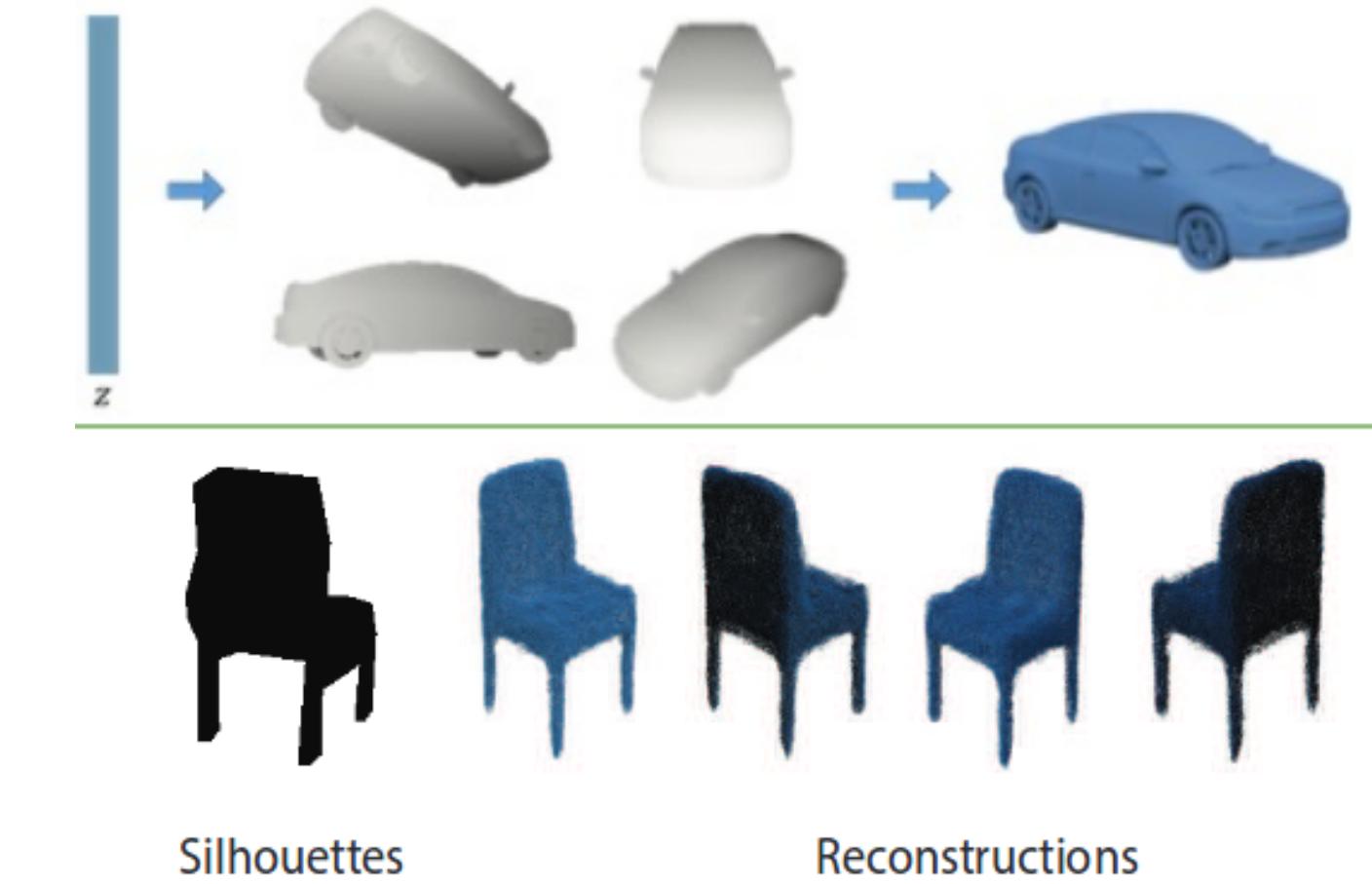
sign identification



optical character recognition

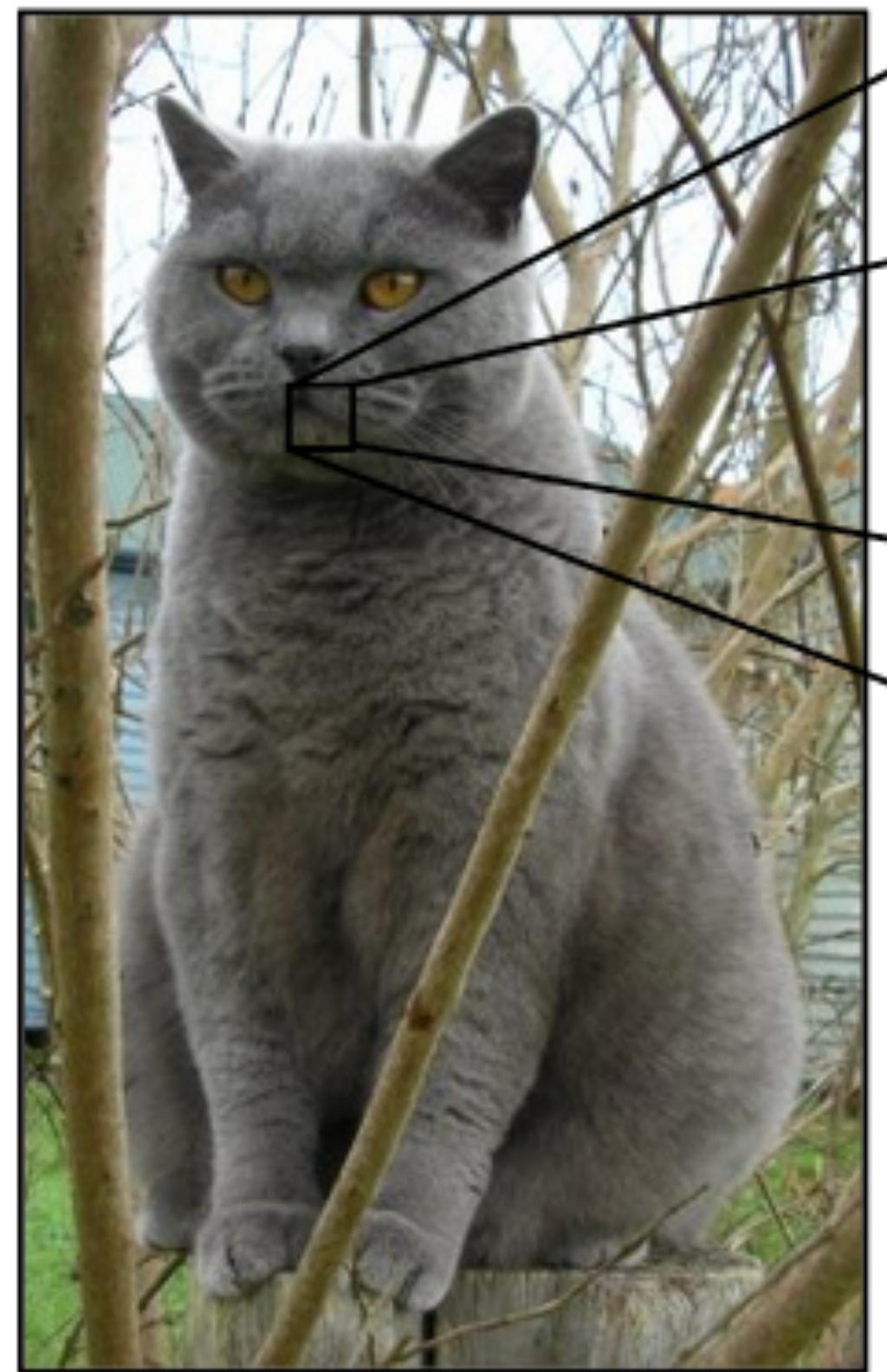


3d reconstructions



# Data begins with a measurement

- A **measurement is an action that determines a property** of a system.
  - *E.g., silver halide crystals in film reducing to metallic silver determine light intensity.*
- Stored measurements are data.
- In computers, we store data in a **digital representation**.
  - I.e., a list of numbers.



08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 68
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 40 09 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 08 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 21 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 03 59 41 92 36 54 22 40 40 28 66 33 13 80
24 47 38 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 03 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
03 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 35 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 06 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 17 47 48

What the computer sees

image classification

82% cat  
15% dog  
2% hat  
1% mug

# Bits

- More precisely, data is typically stored as sequences of 1s and 0s
$$(Y_1, Y_2, \dots, Y_n) \in \{0,1\}^n$$
- I will tell you the story of a single bit of data, which is my story.
- What I am trying to highlight:
  - **Data is not an abstract thing.**
  - The processes that produce it are complex and also very personal.

# A Single Bit

- 1987, I was born in Boston.
- 2016, MSc from UofT.
- 2019, a novel coronavirus spreads across the world.
- 2020, PhD from Oxford.
- 2020, I joined the faculty at UofT.
- 2022, I decided to stop being as cautious about COVID.

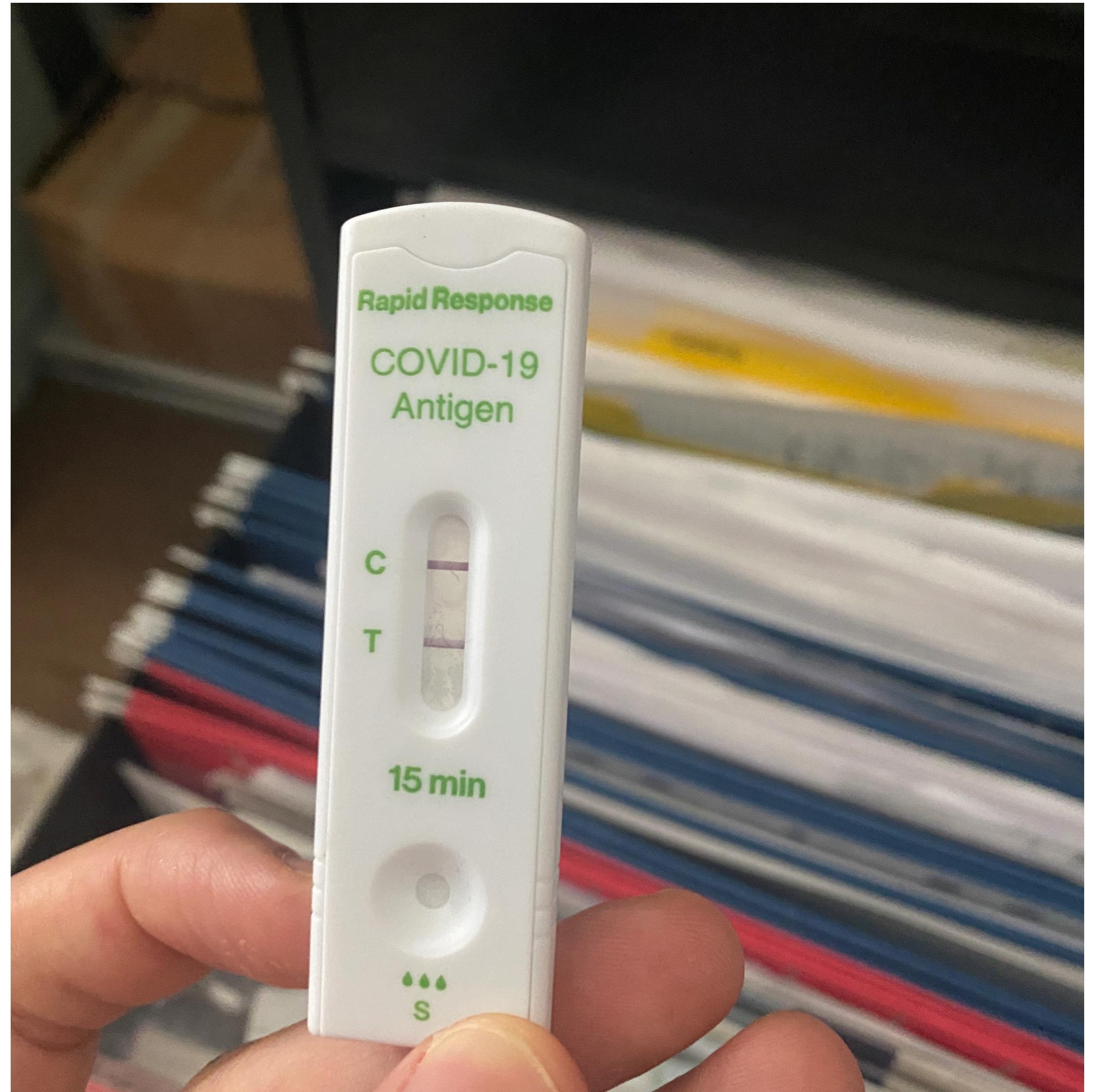


# A Single Bit

- Let  $Y$  be the outcome of a rapid antigen test, 1 if positive, 0 o.w.
- 7 February 2022, the single bit that changed my life:

$$Y = 1$$

- Something to think about: what is the provenance of a bit?



# Prediction

# Prediction

- Could I have predicted whether I would test positive before the test?
- Let's study this abstractly. At a high level, the set up is as follows.
  - We specify a **prediction before** observing the outcome. A prediction is a statement about a future event.
  - The **loss quantifies our error on average on random, unseen data.**

# Bernoulli outcome with cross-entropy

## A special case

- Represent the test outcome as a binary random variable  $Y \in \{0,1\}$ .
- We specify a prediction using a real number  $q \in [0,1]$  to model the probability that  $Y = 1$ , i.e.,  $P(Y = 1)$ .
- How do we score our prediction?

\*The base of the logarithm does not change our discussion.

# Bernoulli outcome with log-loss

## A special case

- We can score our prediction for a specific outcome using the log-loss\*,

$$\ell(Y, q) = -Y \log(q) - (1 - Y)\log(1 - q).$$

- To capture how well we expect to do *in expectation* on an *unseen outcome*, we would like to use the **cross-entropy loss**,

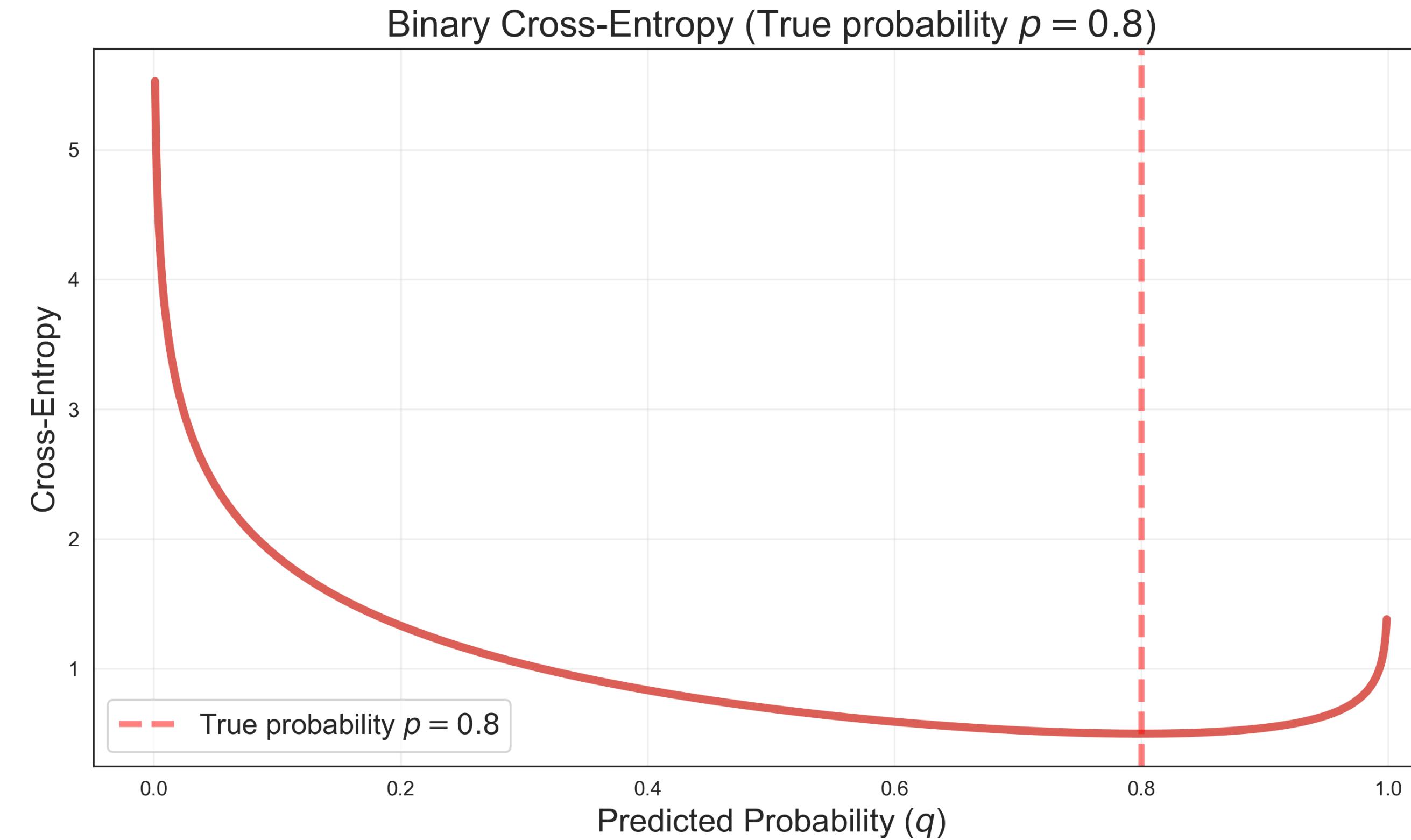
$$\ell(p, q) = -p \log(q) - (1 - p)\log(1 - q).$$

where  $p = P(Y = 1)$ .

\*The base of the logarithm does not change our discussion.

# Why is this a good choice for the loss?

$p$  uniquely minimizes the cross-entropy



The prediction with the lowest risk is the true probability  $p$ .

# Uncertainty

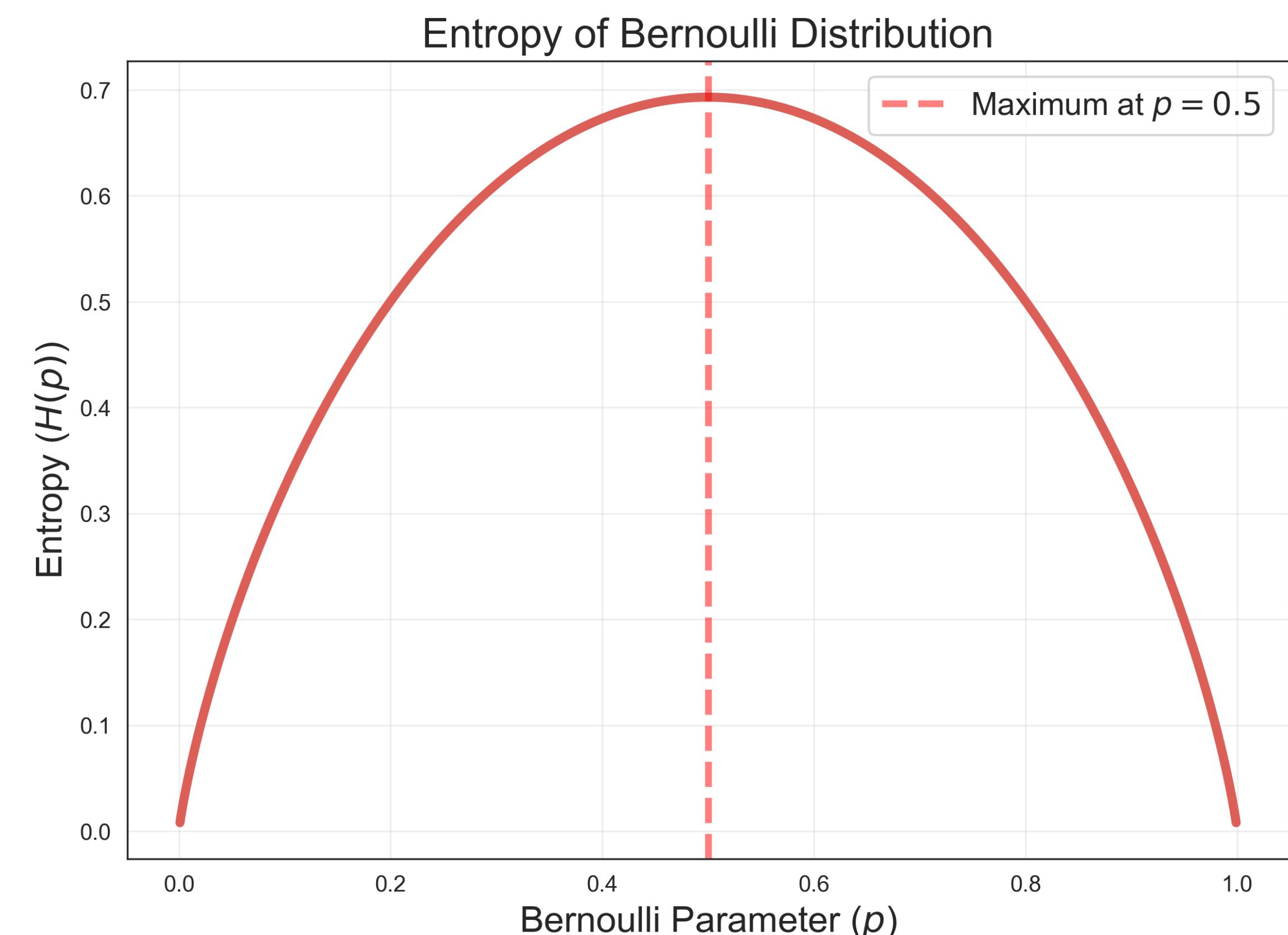
## Motivating the cross-entropy

- You can think of the cross-entropy as quantifying the “**amount of uncertainty that is resolved when we observe an outcome  $Y$  given that I expected  $P(Y = 1) = q$** ”, which measures how surprised we are to observe its value.
- This reasoning can be formalized and you can show that the cross-entropy uniquely captures a certain **statistical notion of uncertainty**.

# Entropy

## the least surprised we could be

- The cross-entropy is minimized at  $q = p$  and equals
$$-p \log(p) - (1 - p)\log(1 - p)$$
- Called the **entropy, which is the least surprised we could be.**
- Most surprised at outcomes if  $p = 0.5$  and least surprised if  $p \in \{0,1\}$ .



# Recap

## Prediction

- A prediction is a statement about a future event.
- We can predict random bits by specifying the probability of them being 1.
- The cross-entropy scores our surprise at observing the outcome.
- The optimal prediction under the cross-entropy is the true probability.

# Learning

# Learning

- To predict well, we want  $P(Y = 1)$ . **But how can we get this in practice?**
- **Learning is the study of procedures that estimate predictors from data.**
- When we do a good job of learning, i.e., we found a good predictor from a set of observations, we say that we have achieved good **generalization**.
- Returning to our Bernoulli example, we will study a simple learning algorithm.

# Learning

## A special case

- We observe a data set  $\{Y_i\}_{i=1}^n$  for outcomes  $Y_i$  that are i.i.d.  $\text{Bern}(p)$ .
  - *E.g.*, a set of COVID test outcomes from Toronto during the pandemic.
- Ideally we would minimize the cross-entropy to get  $p$ ,

$$\min_{q \in [0,1]} \ell(p, q)$$

# But there's a problem!

## A special case

- So far we've been referring to the cross-entropy  $\ell(p, q)$ , but we don't actually have  $p$ !
- To get around this, we can approximate the cross-entropy with **the empirical cross-entropy**:

$$\frac{1}{n} \sum_{i=1}^n \ell(Y_i, q).$$

- Notice that  $\frac{1}{n} \sum_i \ell(Y_i, q) \rightarrow \ell(p, q)$  by properties of random variables

# Learning

## A special case

- This motivates **empirical risk minimization**, which is the estimation procedure that finds

$$\min_{q \in [0,1]} \frac{1}{n} \sum_{i=1}^n \ell(Y_i, q)$$

- Given  $\{Y_i\}_{i=1}^n$  i.i.d.  $\text{Bern}(p)$ , the ERM w.r.t. the empirical cross-entropy is minimized by

$$\hat{p}_n = \frac{1}{n} \sum_{i=1}^n Y_i$$

# Learning with cross-entropy

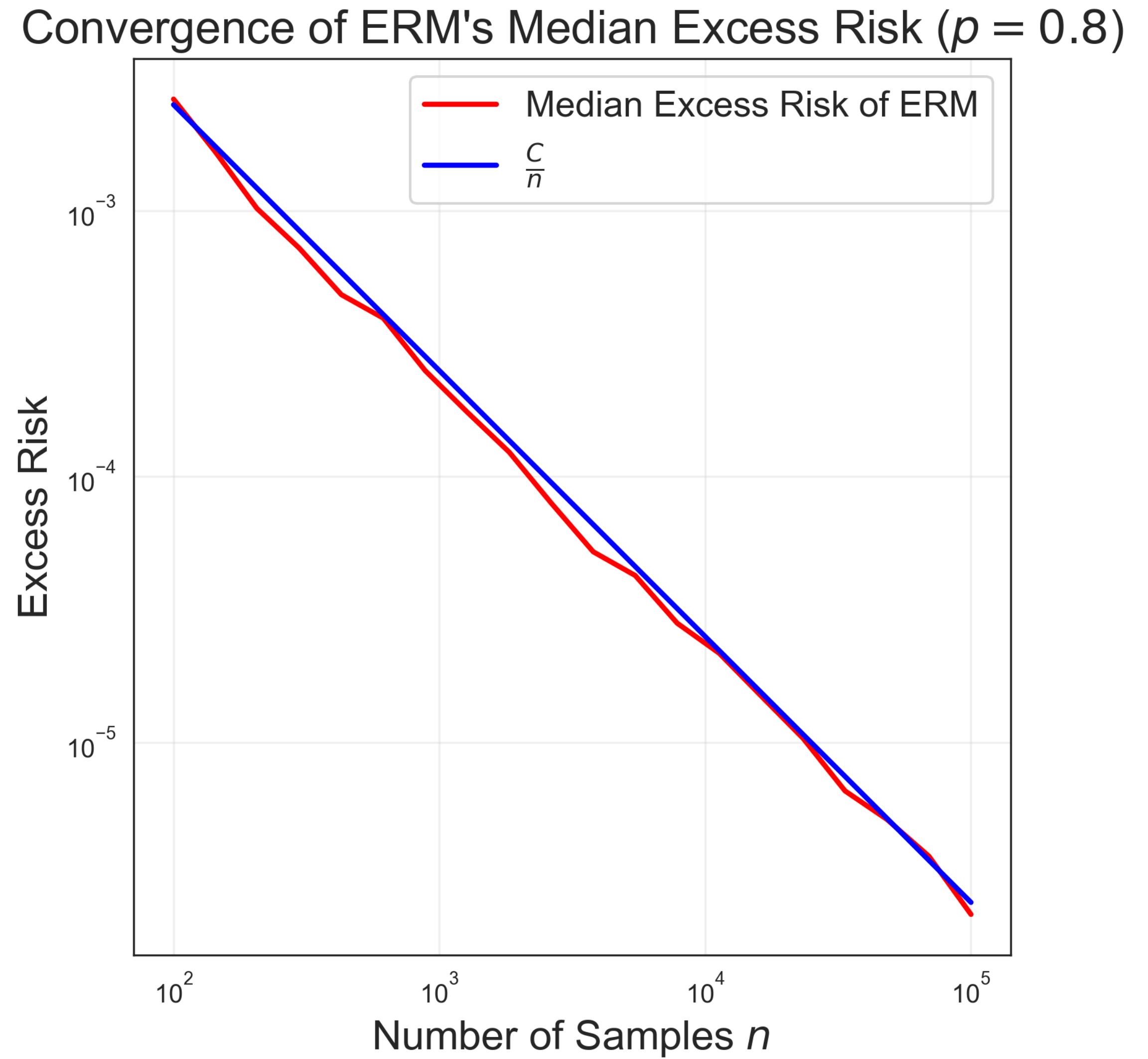
- Does  $\hat{p}_n$  approach  $p$  as  $n \rightarrow \infty$  in some sense?
- We can study this by studying the behaviour of **the excess cross-entropy**

$$\mathcal{E}(q) = \ell(p, q) - \ell(p, p)$$

- Note two things:
  - $\mathcal{E}(q) \geq \mathcal{E}(p) = 0$  for all  $q \in [0,1]$ .
  - $\mathcal{E}(\hat{p}_n)$  is a non-negative real-valued *random* variable.

# Learning convergence

- Let's study convergence in simulation (derivation in the Bernoulli case is a bit tricky).
- **Median converges like,**  
 $\text{median} [\mathcal{E}(\hat{p}_n)] \rightarrow C/n$ 
  - Typical rate for learning.
  - **Key take-home: the more data we have, the better our predictions.**



# Recap

## Learning

- Learning is the study of procedures that estimate predictors from data.
- Empirical risk minimization tries to solve this by picking the predictor that minimizes the average loss on a data set.
- Predictors obtained from data are random because the data is random.
- We can study predictors by studying the excess loss, *i.e.*, the deviation of the expected loss from the best possible expected loss.
- When the excess loss is small, we have generalized.
- **The more data we have, the better we generalize.**

# Conditional Prediction

# Conditional prediction

- Could I have **predicted my test more accurately, if I had other measurements** about me?
  - Did anything in the last 37 years make  $Y = 1$  more likely?
- Answer is **typically yes!**
- Let's study a special case of conditional prediction: **logistic regression**.



# Logistic regression

## A special case

- Let  $X \in \mathbb{R}^d$  be a random vector of other measurements called “features”.
  - E.g., my age as a number, my location as coordinates, etc.
- Seeing  $X = x$  may inform us about  $Y$  and make  $Y = 1$  more predictable.
  - I.e.,  $P(Y = 1 | X = x)$  may have less entropy than  $P(Y = 1)$ .
- To take advantage of this, we can build conditional predictions of  $Y$  given  $X$ .
  - Logistic regression is a special case!

# Logistic regression

## A special case

- **Logistic regression:** predict  $P(Y = 1 | X)$  with a sigmoid function that depends linearly on  $X$ :

$$q(X) = \sigma\left(\sum_{i=1}^d w_i X_i\right)$$

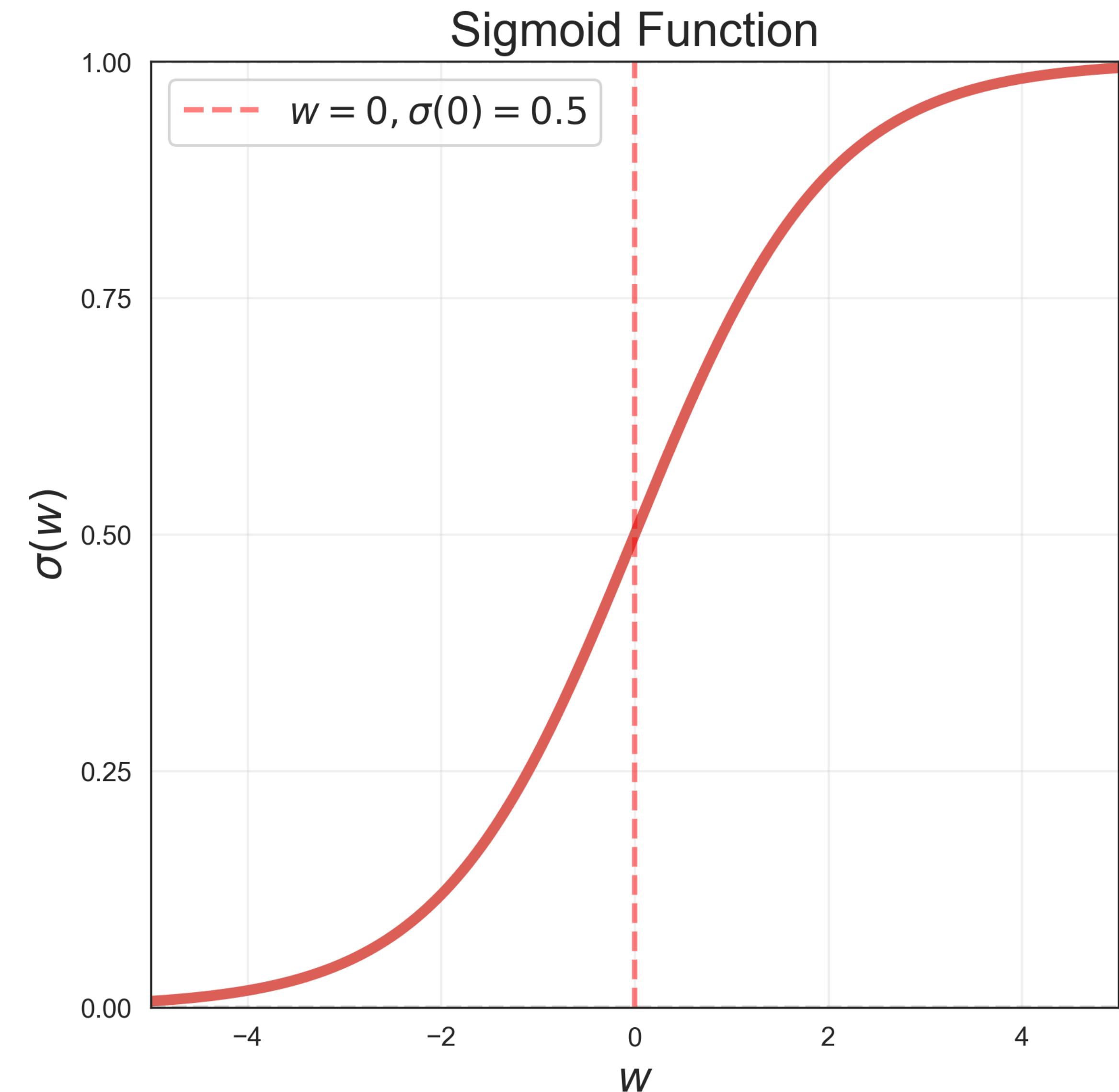
where  $\sigma(t) = \frac{1}{1 + \exp(-t)}$  is the sigmoid and  $w \in \mathbb{R}^d$  are called parameters.

- The params. plus the rule for computing the prediction is called **the model**.

# Logistic regression

## Intuition

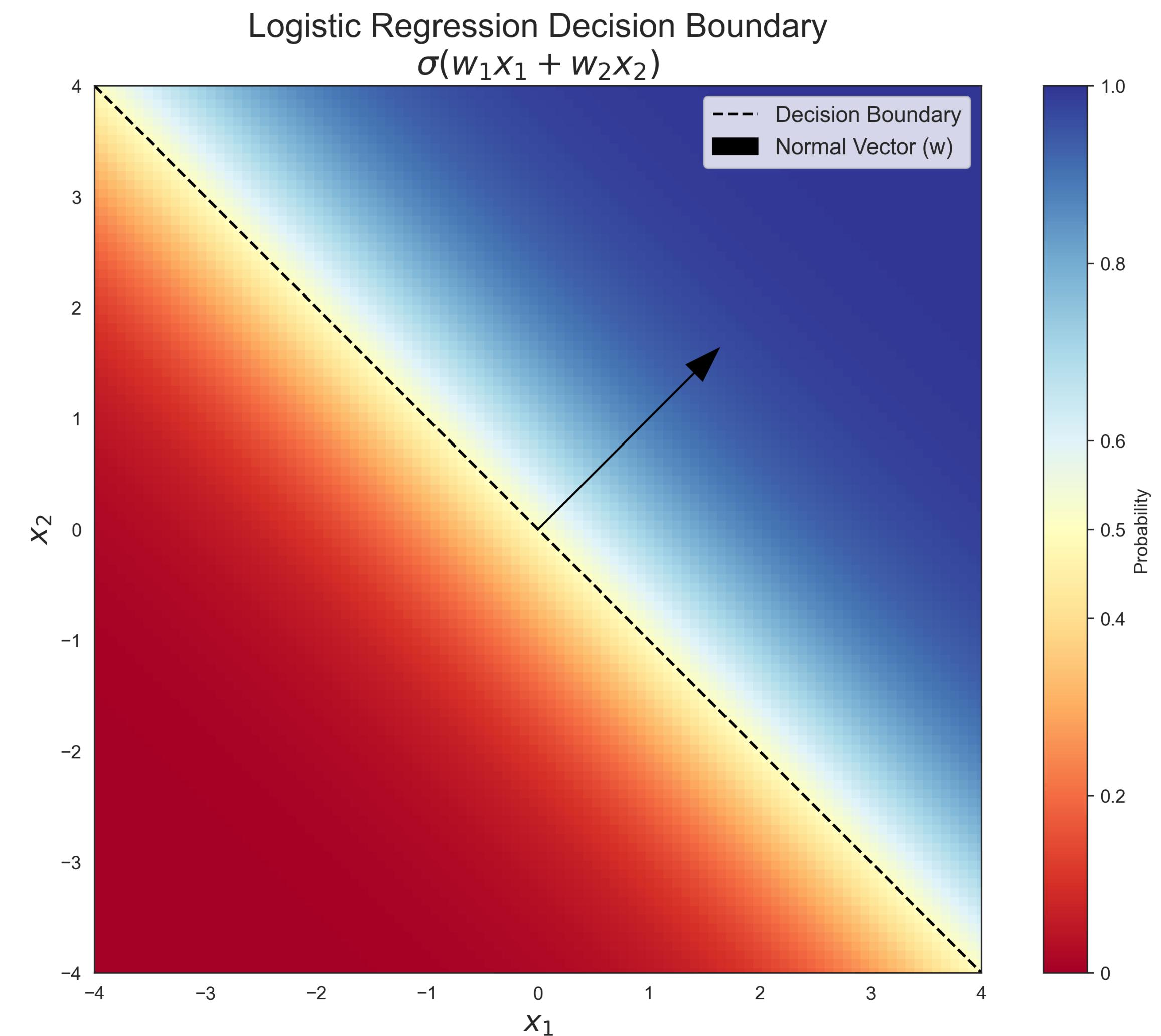
- This is a generalization of the Bernoulli prediction case we considered.
  - Take  $X \equiv 1$  to be constant.
  - Then  $q(1) = \sigma(w)$  where  $w \in \mathbb{R}$ .
  - Can represent any  $q \in (0, 1)$  this way.



# Logistic regression

## Intuition

- The set of  $x$  where  $\sigma(w^\top x) = 0.5$  is the hyperplane
$$\{x \in \mathbb{R}^d : w^\top x = 0\}$$
- **$w$  separates our predictions.**
  - As  $x$  travels along  $w$ , our prediction that  $Y = 1$  increases towards 1.
  - As  $x$  travels along  $-w$ , our prediction that  $Y = 1$  decreases towards 0.



# Logistic regression

Which  $w$  should we pick?

- How do we get  $w^*$ ? As before, it is common to **optimize the conditional cross-entropy**:

$$\hat{R}_n(w) = \frac{1}{n} \sum_{i=1}^n \ell(Y_i, \sigma(w^\top X_i)) \text{ where } (X_i, Y_i) \text{ are i.i.d. as } (X, Y)$$

# Logistic regression

## Which $w$ should we pick?

- Optimizing is harder than the Bernoulli case: (i) sometimes there's no minimizer, (ii) when there is a minimizer, it's not always unique, and (iii) even if it's unique, there's often no closed form!
- Out of scope to study, let's assume there exists a unique minimizer of the empirical risk,  $\hat{w}_n^*$ .
- How can we find the  $\hat{w}_n^*$ ? **Gradient descent is one choice!**

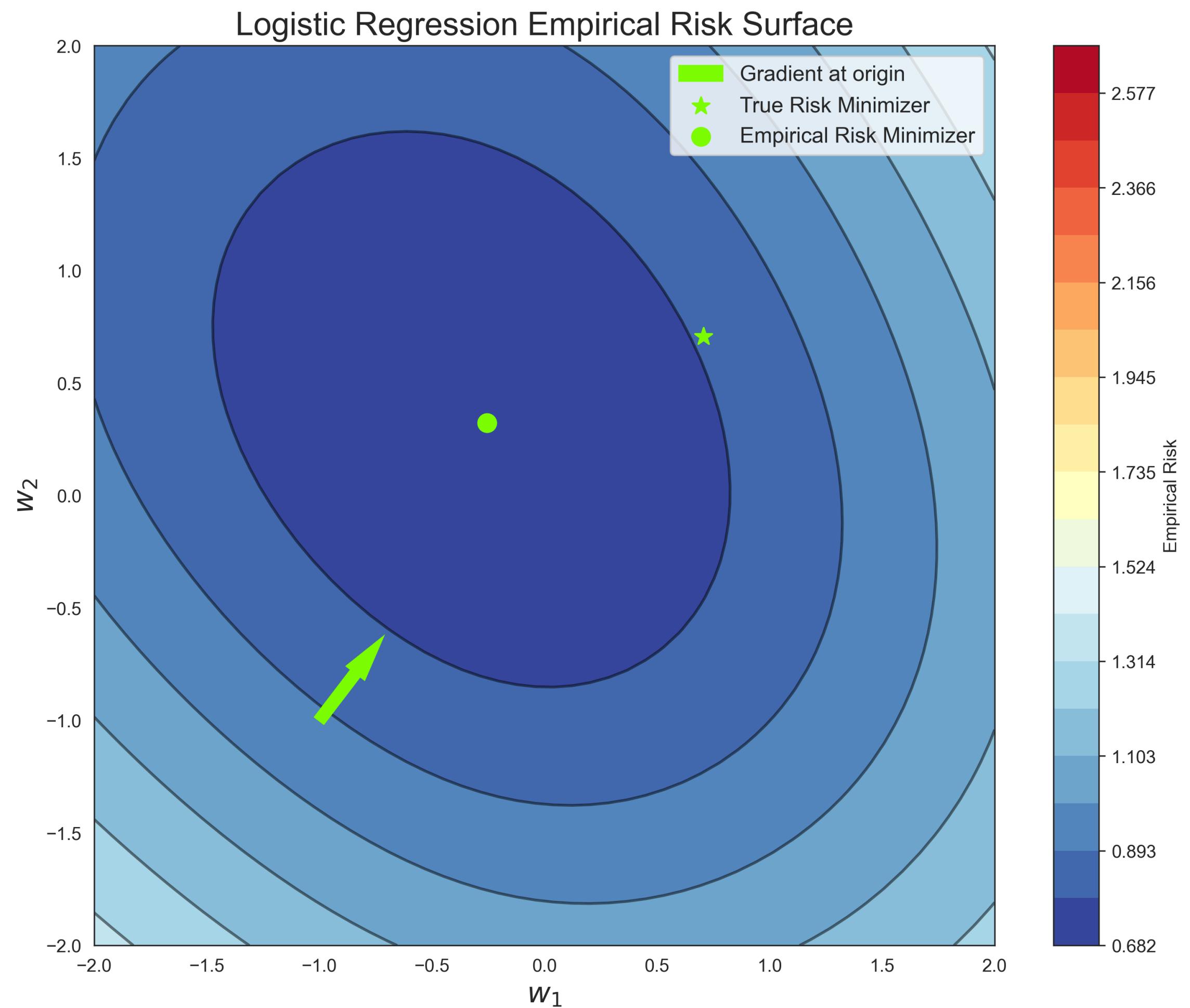
# Gradient descent

## Intuition

- The **gradient of the empirical cross-entropy** is the vector of partial derivatives,

$$\nabla \hat{R}_n(w) = \left( \frac{\partial \hat{R}_n}{\partial w_j} \right)_{j=1}^d$$

- The **negative gradient is the direction of greatest instantaneous descent on the surface of  $\hat{R}_n(w)$ .**



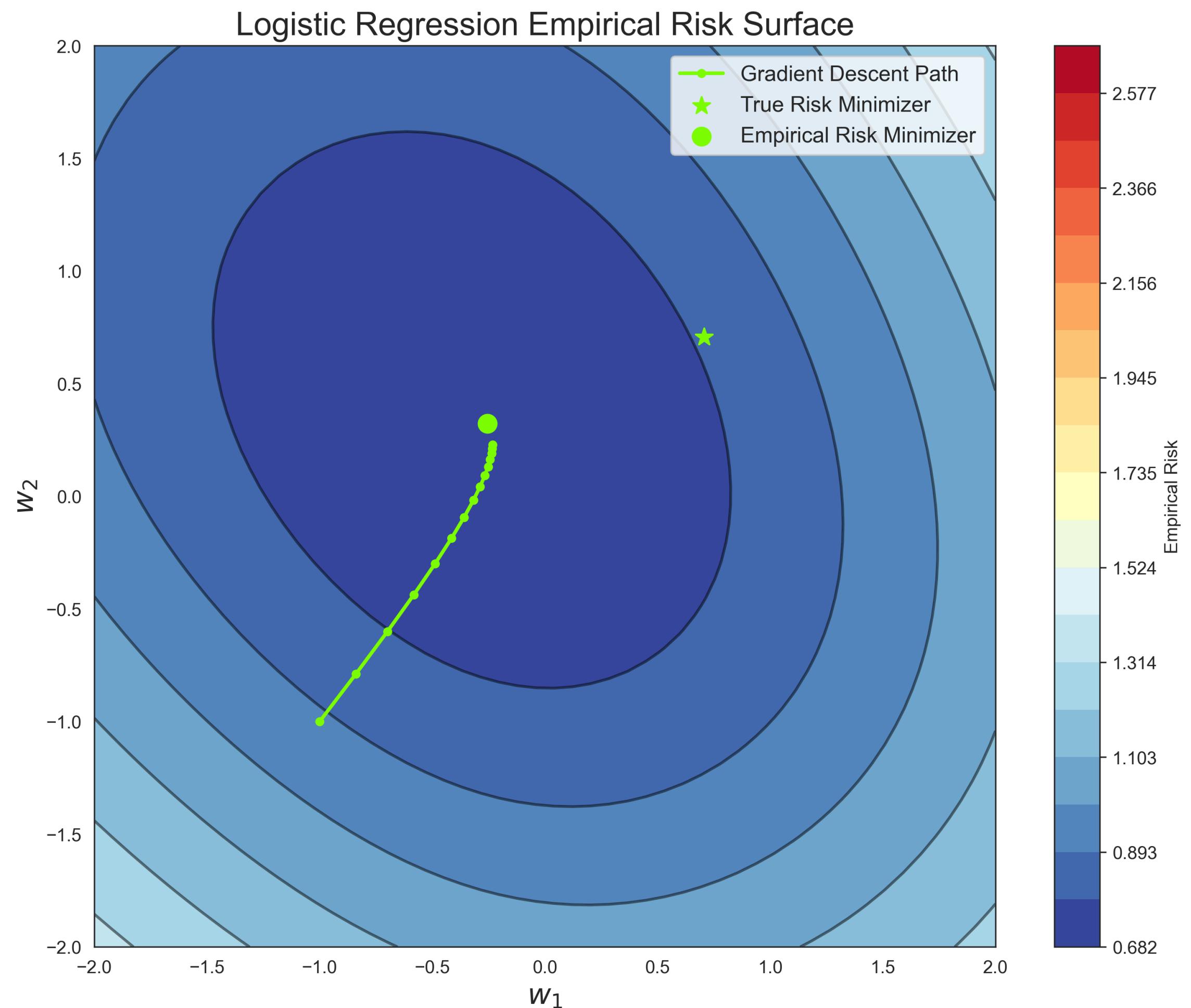
# Gradient descent

## Intuition

- The gradient descent algorithm iteratively **follows the gradient**:

$$w^{(t+1)} = w^{(t)} - \eta \nabla \hat{R}_n(w^{(t)})$$

for some step-size  $\eta > 0$ .



# Gradient descent

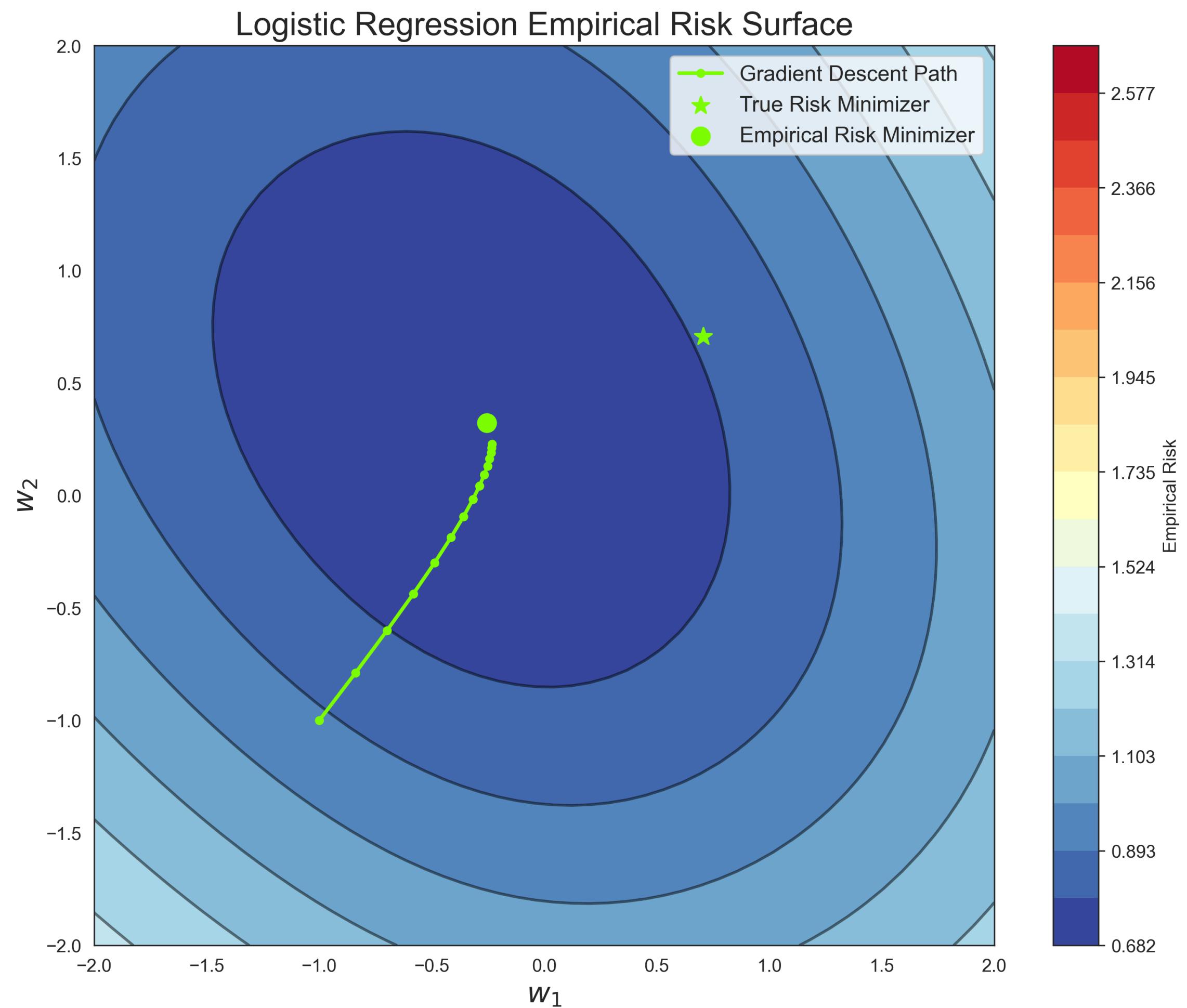
## Intuition

- In our setting, we have

$$w^{(t)} \rightarrow \hat{w}_n^*$$

for small  $\eta$  because the empirical risk is **smooth** and **convex**.

- Technical terms that are out of scope for us.
- Optimization is another very large, very deep field (also out of scope).



# Logistic regression

How well does ERM perform in this case?

- Does  $\hat{w}_n^*$  approach the best possible  $w^*$  in expectation as  $n \rightarrow \infty$  in some sense?
- Classical result (see Ostrovskii and Bach, 2020): under mild smoothness conditions, the *expected loss on a held out data point behaves like*

$$\frac{d}{2n} + o(n^{-1}) \text{ as } n \rightarrow \infty$$

- **Key take-home: the more data we have, the better our predictions, BUT the more parameters, the worse our predictions.**

# Recap

## Conditional Prediction

- Observing more measurements can sometimes improve predictions.
- We can compute conditional predictions with parametric models like logistic regression.
- In general, finding the ERM of parametric models is challenging and we often resort to iterative optimization algorithms like gradient descent.
- **Rule of thumb: learning in parametric models improves with data and deteriorates with parameter count.**

# This Bootcamp

# Quick admin

- You should have your teams already.
- Schedule:
  - Today: wrap up the lectures (hopefully)
  - March 3: hackathon, first model submission
  - March 6: wrap-up hackathon

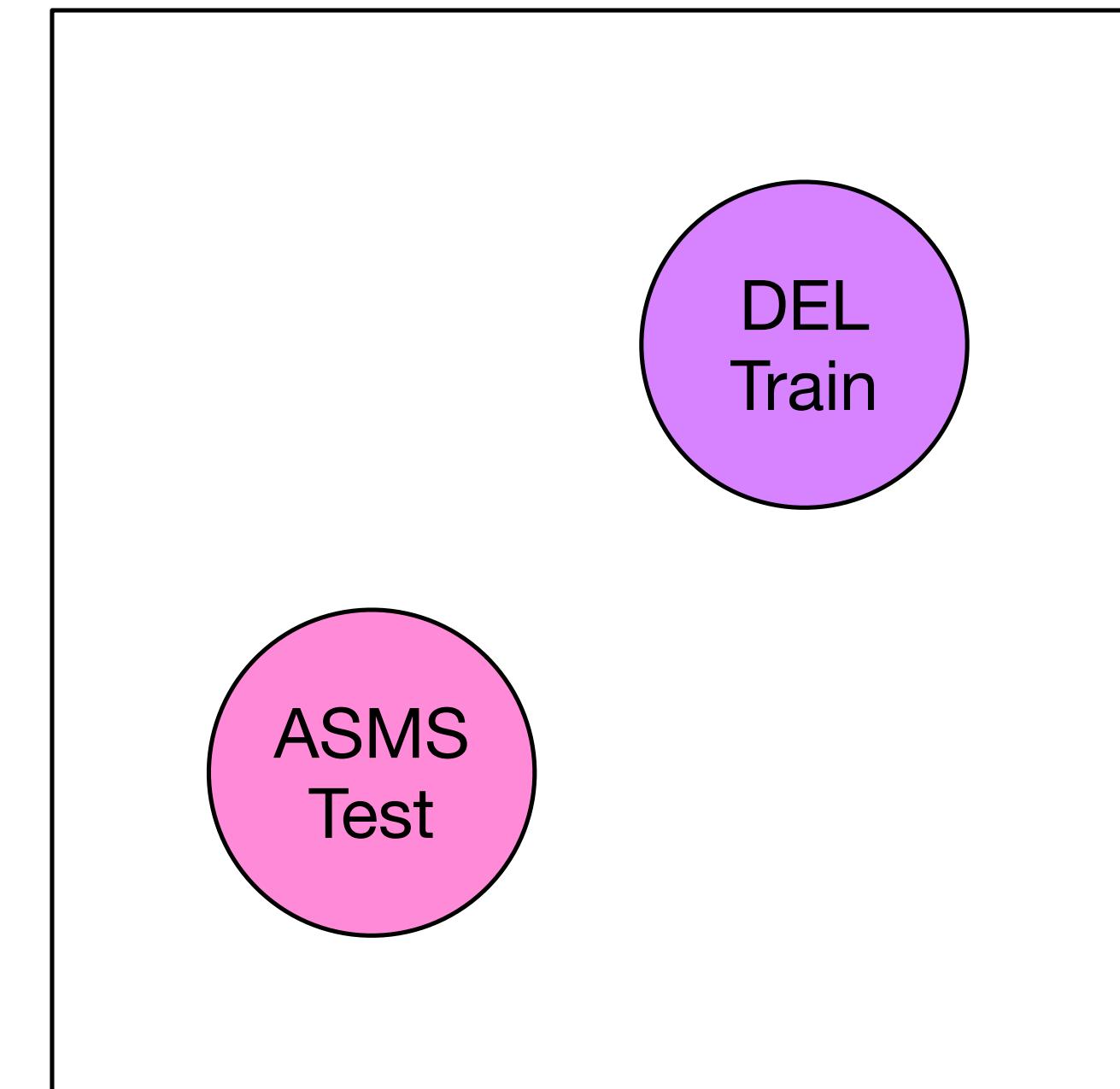
# Quick admin

- You have a **training set** of pairs  $(X, Y_{DEL})$  where X is some chemical representation and Y is a binary outcome of hit / no hit on a DEL screen for WDR91.
  - Ben covered chemical representations.
- Your models will be **tested on a separate library** of compounds by according to how many SPR hits against WDR91 they can find.
  - We release the test inputs.
  - You submit predictions.
  - We report your performance on the test labels without releasing the labels.

# The challenge

- You are facing two challenges:
  1. The test compounds look very different from training compounds.
  2. The test labels are from a different assay.
- 1. is probably a harder challenge.

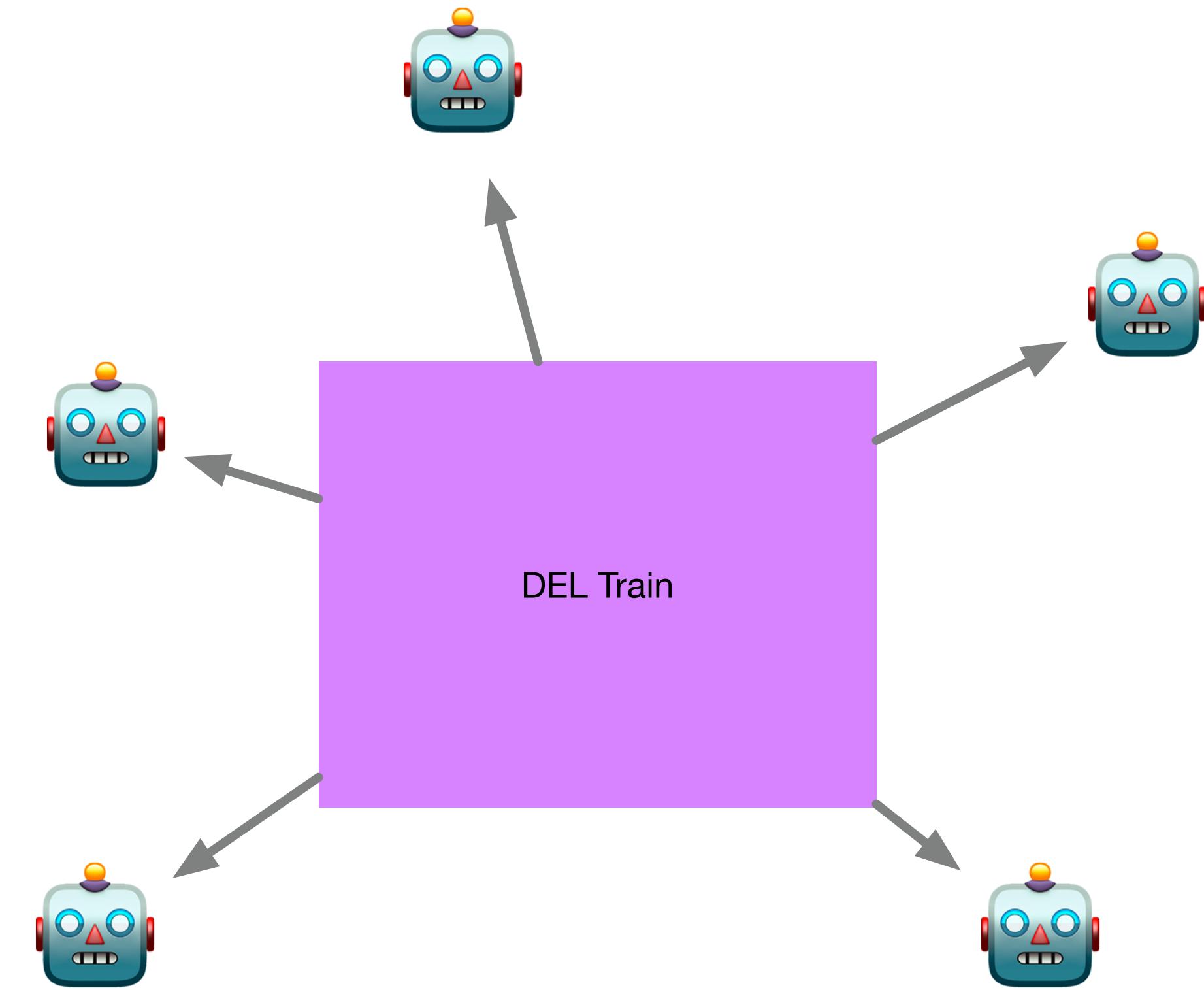
Chemical Space



# The challenge

## More generally

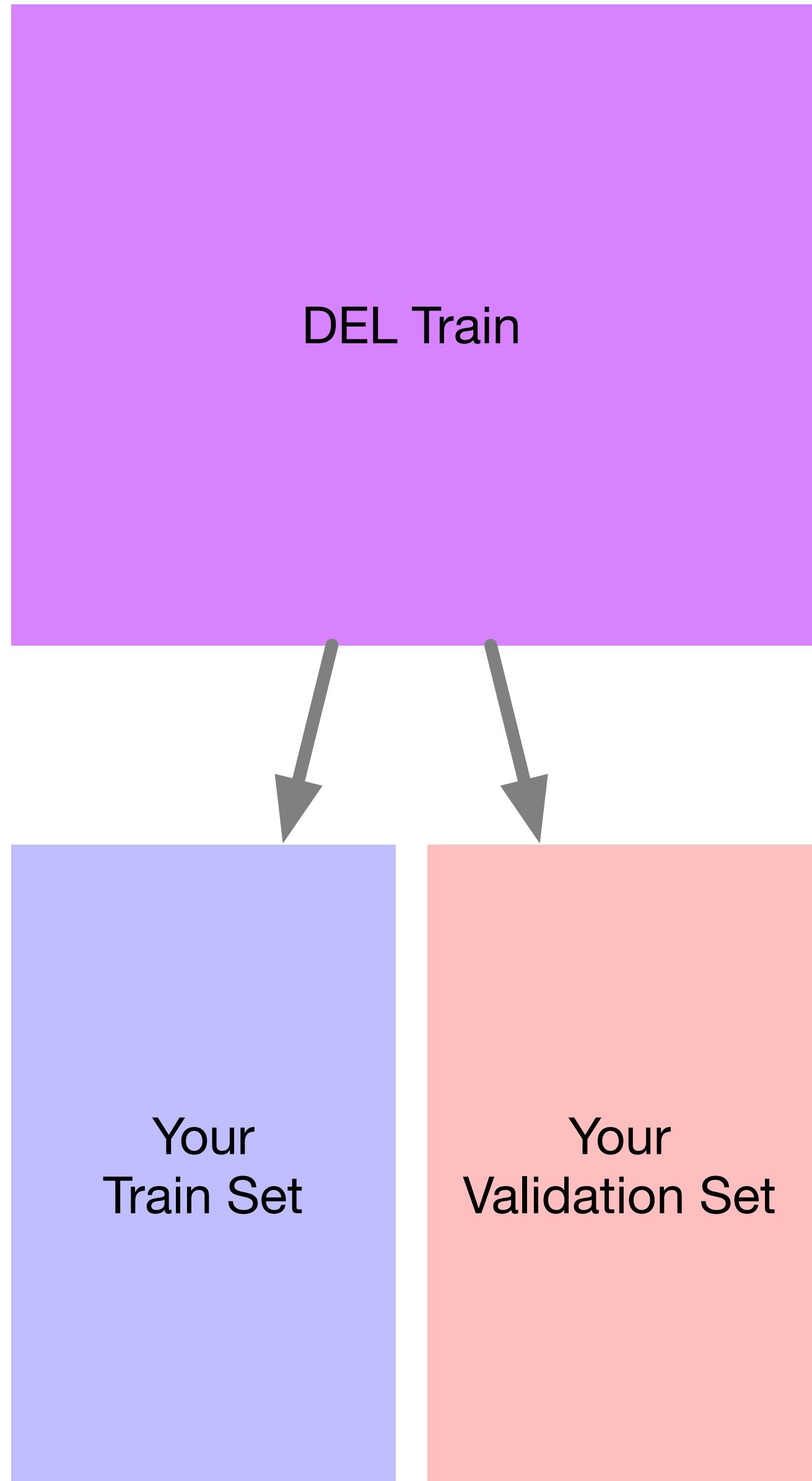
- You will be faced with a bunch of choices when you develop your model.
- Then your model will be assessed by metrics on the test set that we report back to you.
- How do you make those choices?



Which model do I pick???

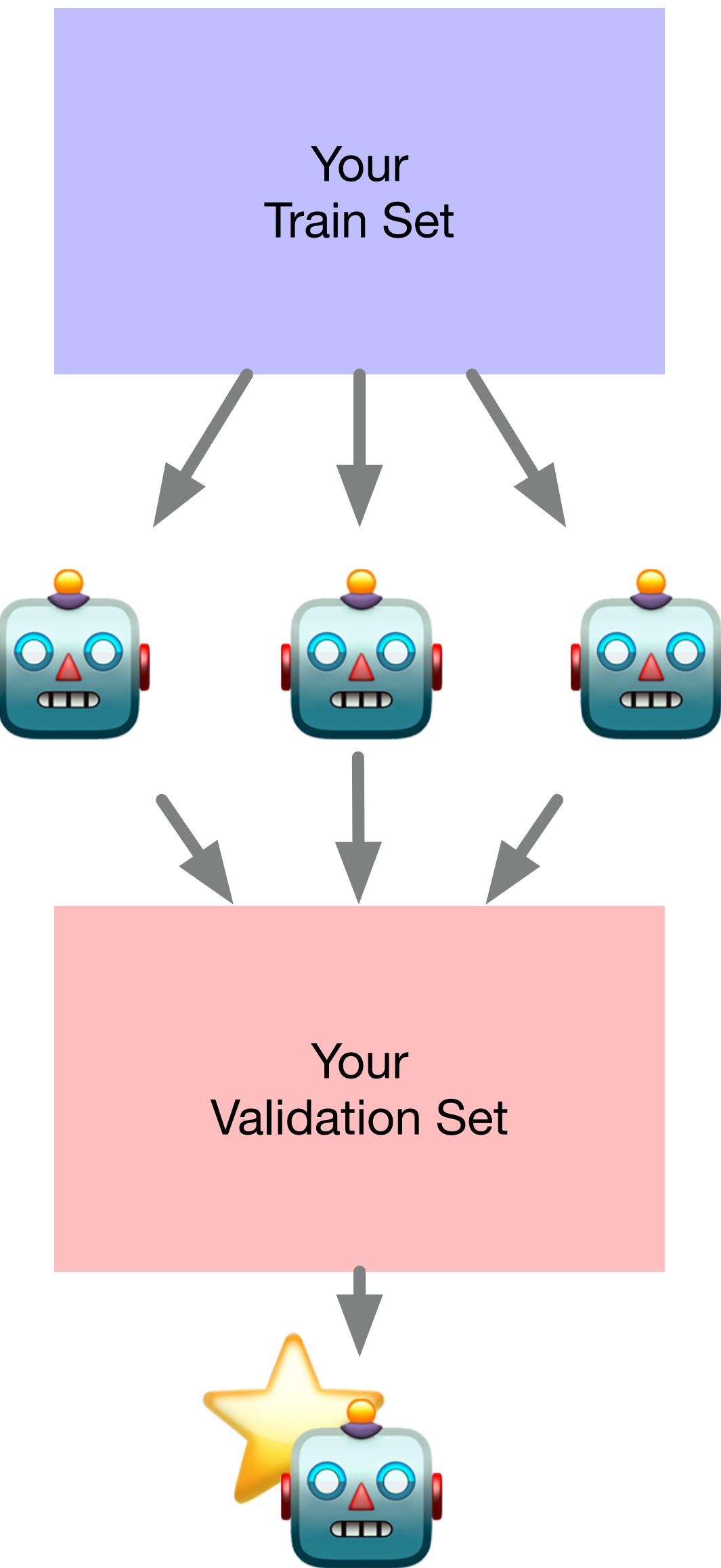
# Validation sets

- You have full access to the DEL training set.
- You can mimic the testing set up by splitting the train data into a training set and **a validation set**.



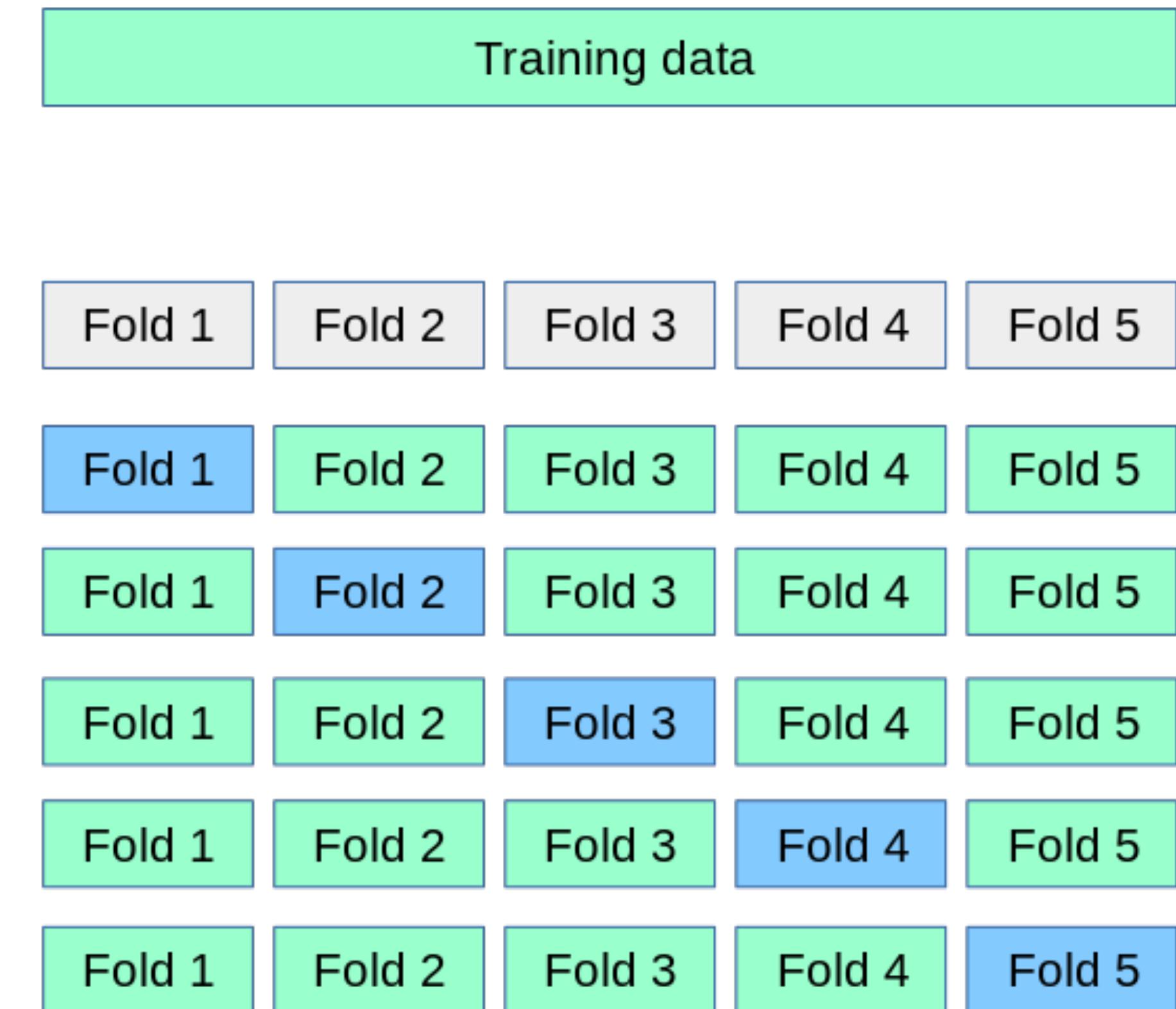
# Validation sets

- The basic idea:
  - Train models on the train set
  - Pick the model that performs best on the validation set (according to your favourite metric).
- **Question: what is the best validation set if I want to pick a model that performs well on the CrossTALK test?**



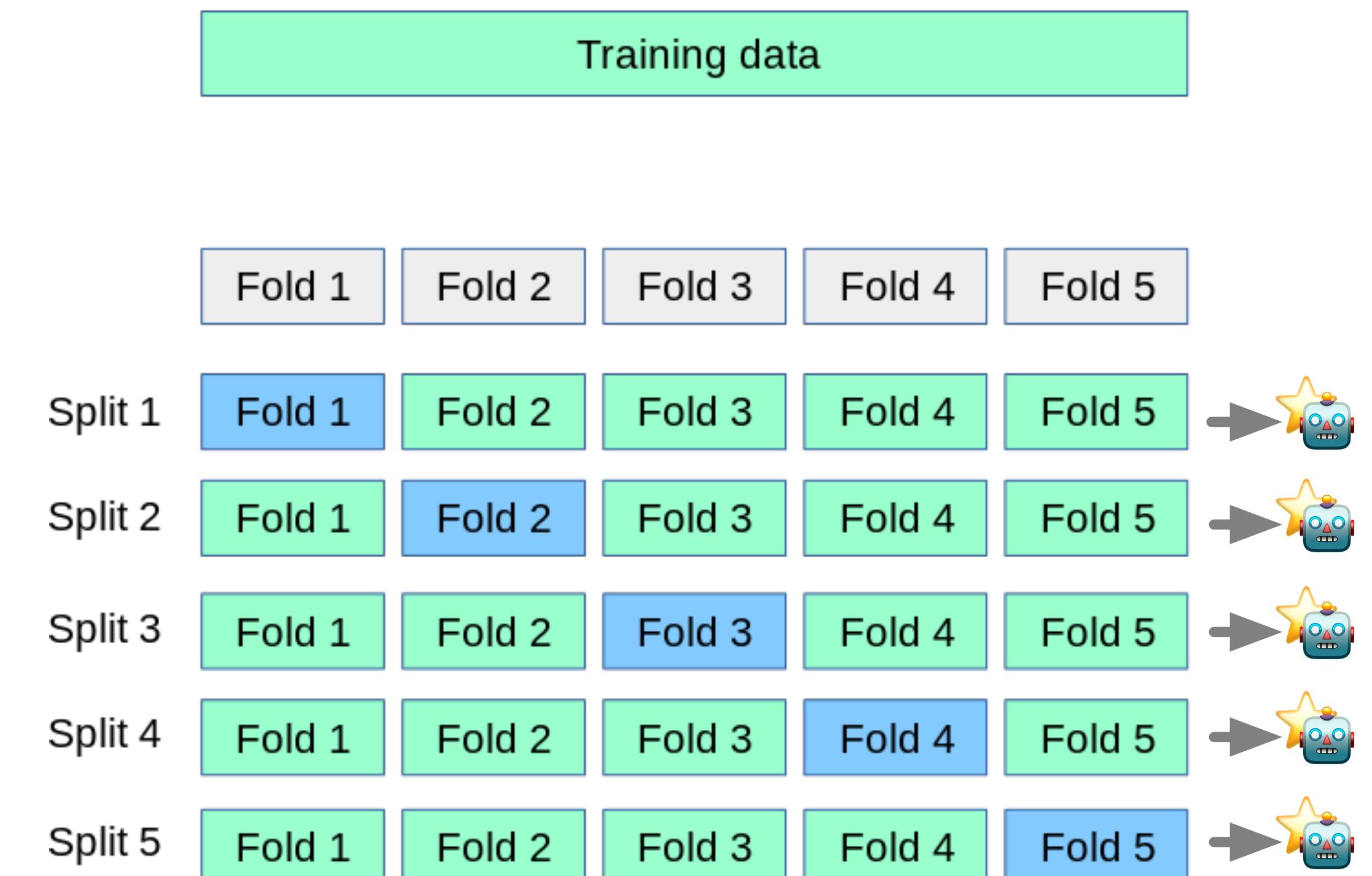
# Cross-validation

- Sometimes we don't have enough data to split into test and train.
- Cross-validation is one way to make better use of your data to help you make model choices.
- Split training data into K folds and each fold takes a turn as the validation set.



# Cross-validation

- Many ways to use cross-validation.
- For example, if you're trying to check whether model A is better than model B.
- Train model A and B on all folds and average their performance on the validation sets. The one that does better on average is probably better.



# Parting thoughts

- Machine learning is a vast and deep field
- There are a lot of approaches you can take to this problem
- Key questions:
  - How similar is your training loss to the testing loss?
  - How similar is your validation loss to the testing loss?
  - How similar is your validation set to the test set?
  - How challenging is your validation split as compared to the test split?

# Parting thoughts

- Some approaches you might consider
  - Training losses that are less similar to cross-entropy and more similar to the ranking metrics that you'll be tested on
    - E.g., <https://github.com/google/rax>
  - Designing validation sets that mimic the kind of shift that you will be tested on
    - E.g., <https://proceedings.mlr.press/v202/klarner23a.html>

**Thanks!**