CLASSIFICATION

Inzamam Rahaman

WHAT IS CLASSIFICATION?

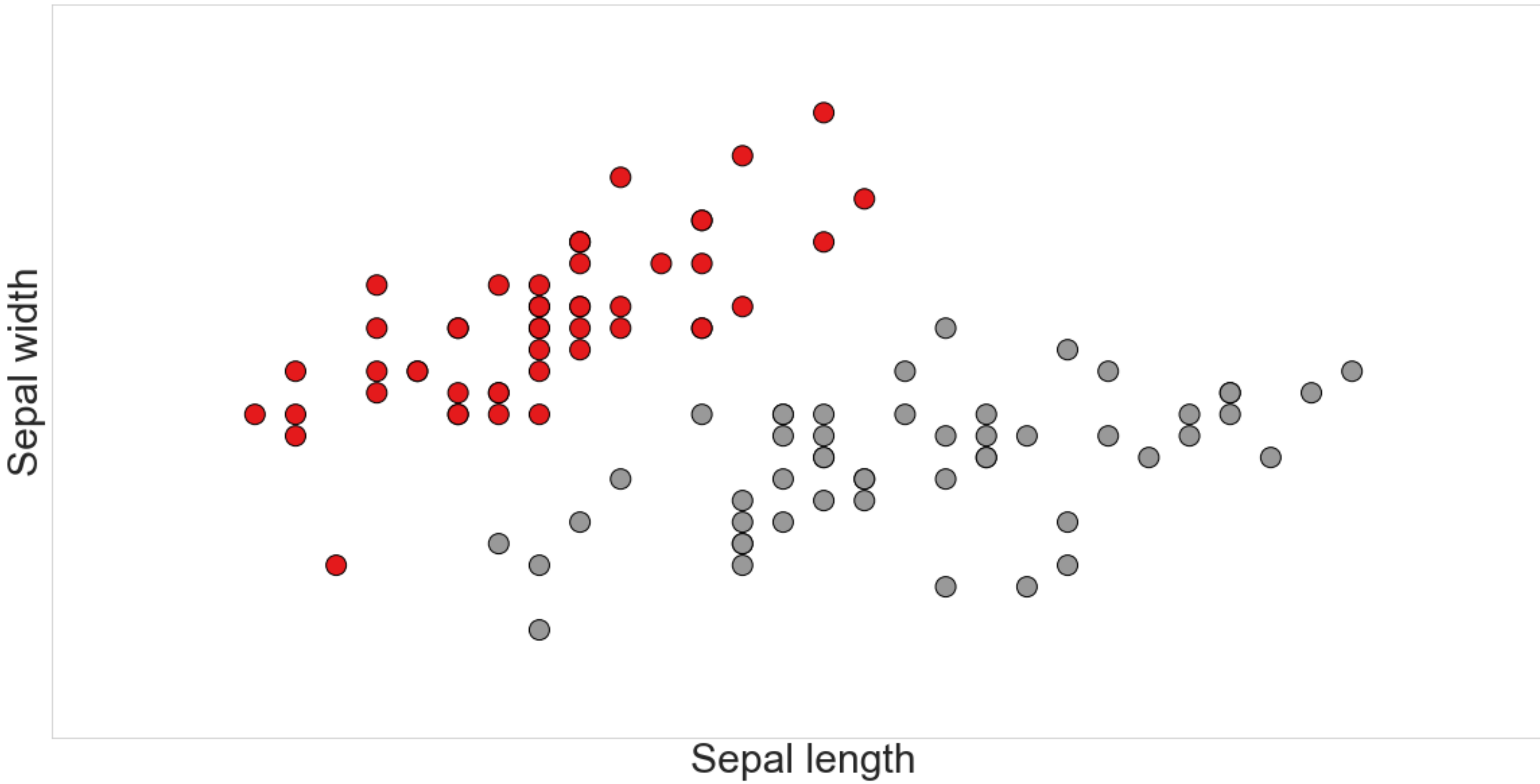
- Supervised Learning
  - Given input-output pairs train model to predict output value(s) for unseen input
  - Two main sub-tasks:
    - Regression (last week)
    - Classification (today)
      - Ordinal Regression (something later?)

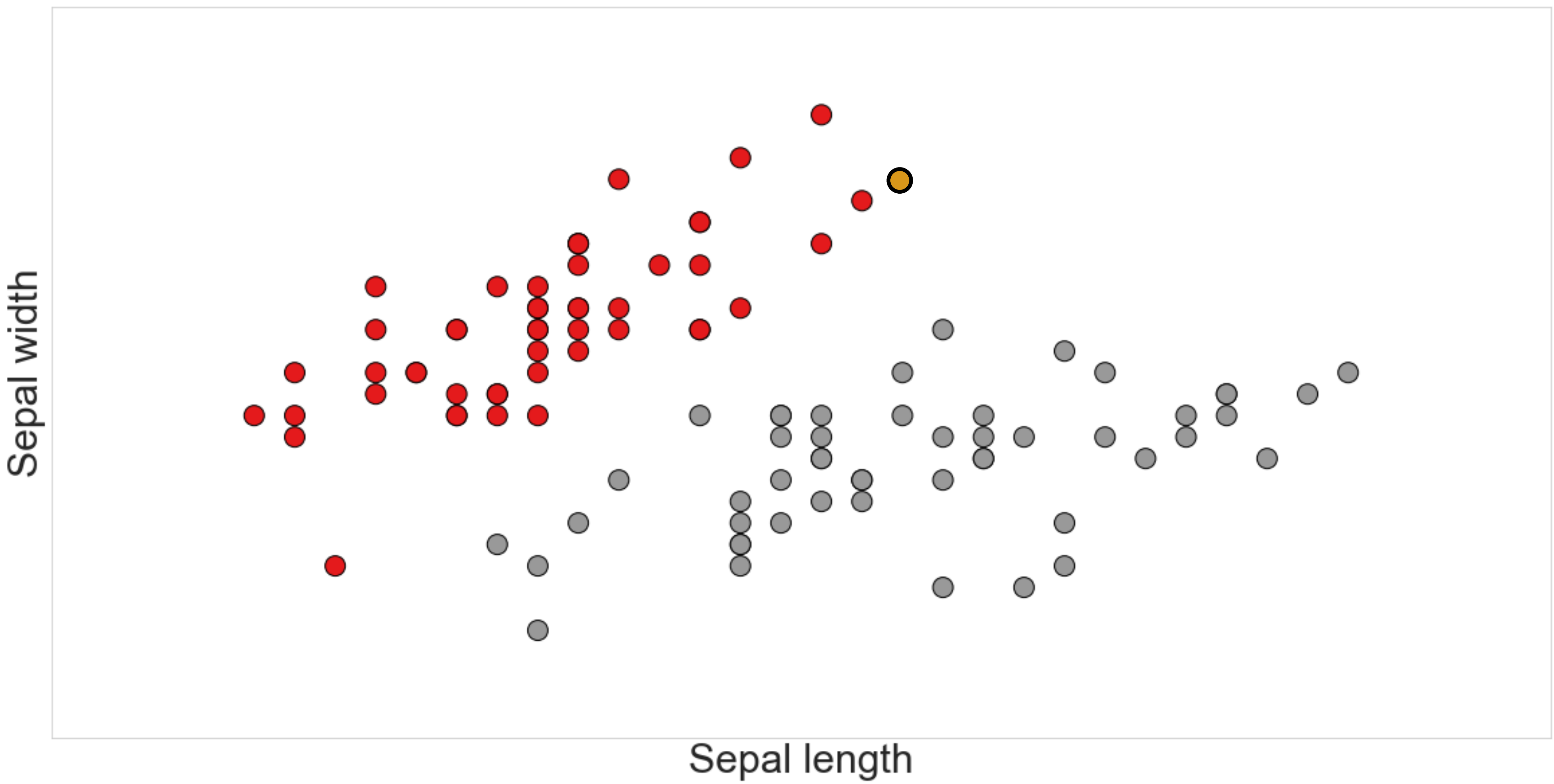
CLASSIFICATION VS REGRESSION

- $\mathscr{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)\}$
- Regression learns model (function) that takes input in $\mathbb{R}^d$ and forecasts output in $\mathbb{R}$ or infinite sized subset of $\mathbb{R}$
- What if our output domain is finite?
  - Regression models assume assume infinite domain.

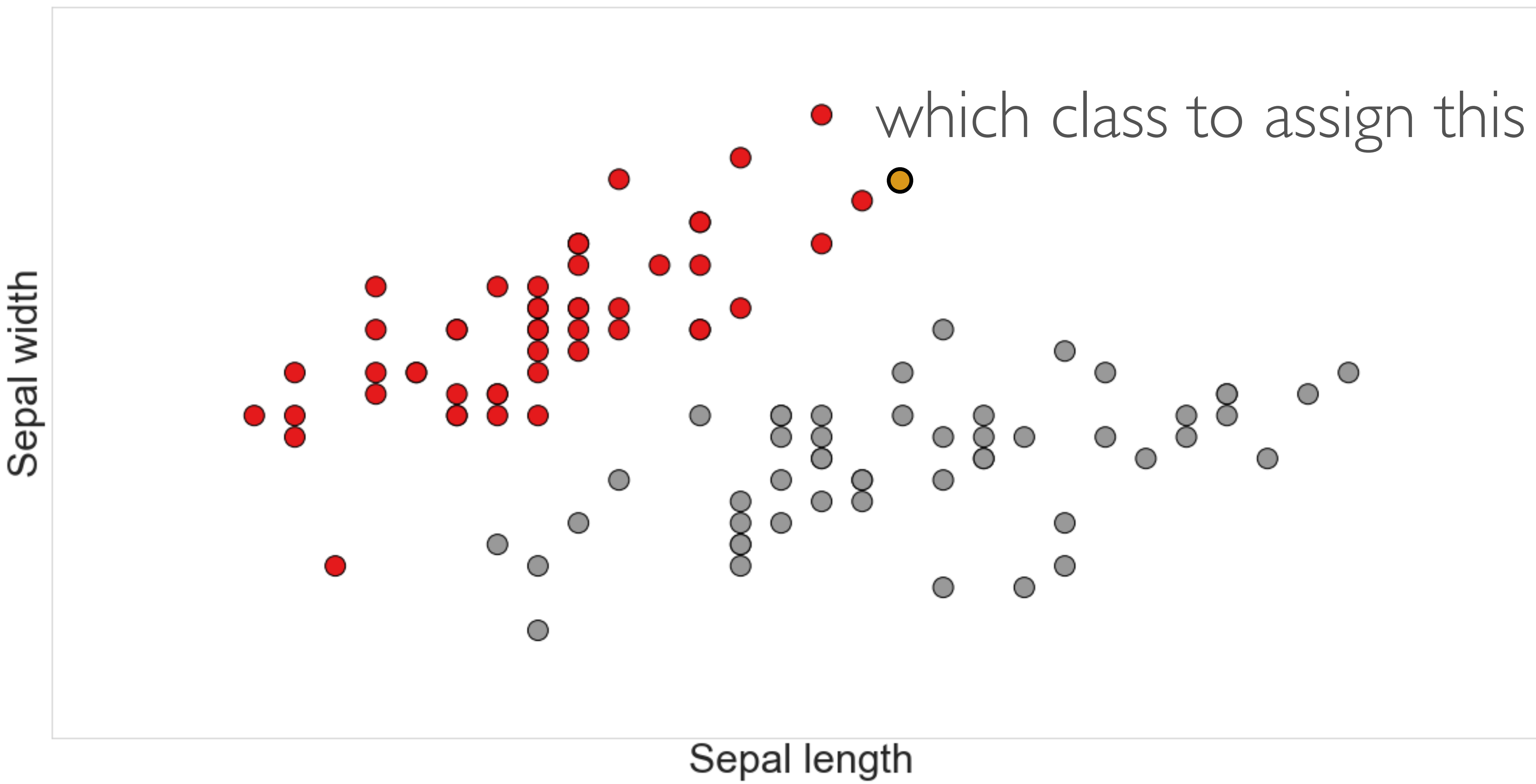
CLASSIFICATION

- Classification assumes that our output domain is discrete and finite
  - E.g. $\{$yes, no$\}$
  - We call elements of our output domain classes
  - When labels are not numerical, we assign an integer label as a proxy for the non numerical class
    - E.g. yes $= 1$ and no $= 0$


Sepal width
Sepal length


Sepal width
Sepal length

- which class to assign this point



CLASSIFICATION VS REGRESSION

- Mindset shift from forecasting values to assigning a data point to the best set
  - E.g. difference predicting the number of months of viable use for a machine part vs indicating whether the part is going to fail in 6 months
    - Notice that there is a relationship here!
    - In principle we can assign threshold to regression output and call that a classification model
    - In fact that is the basis of …

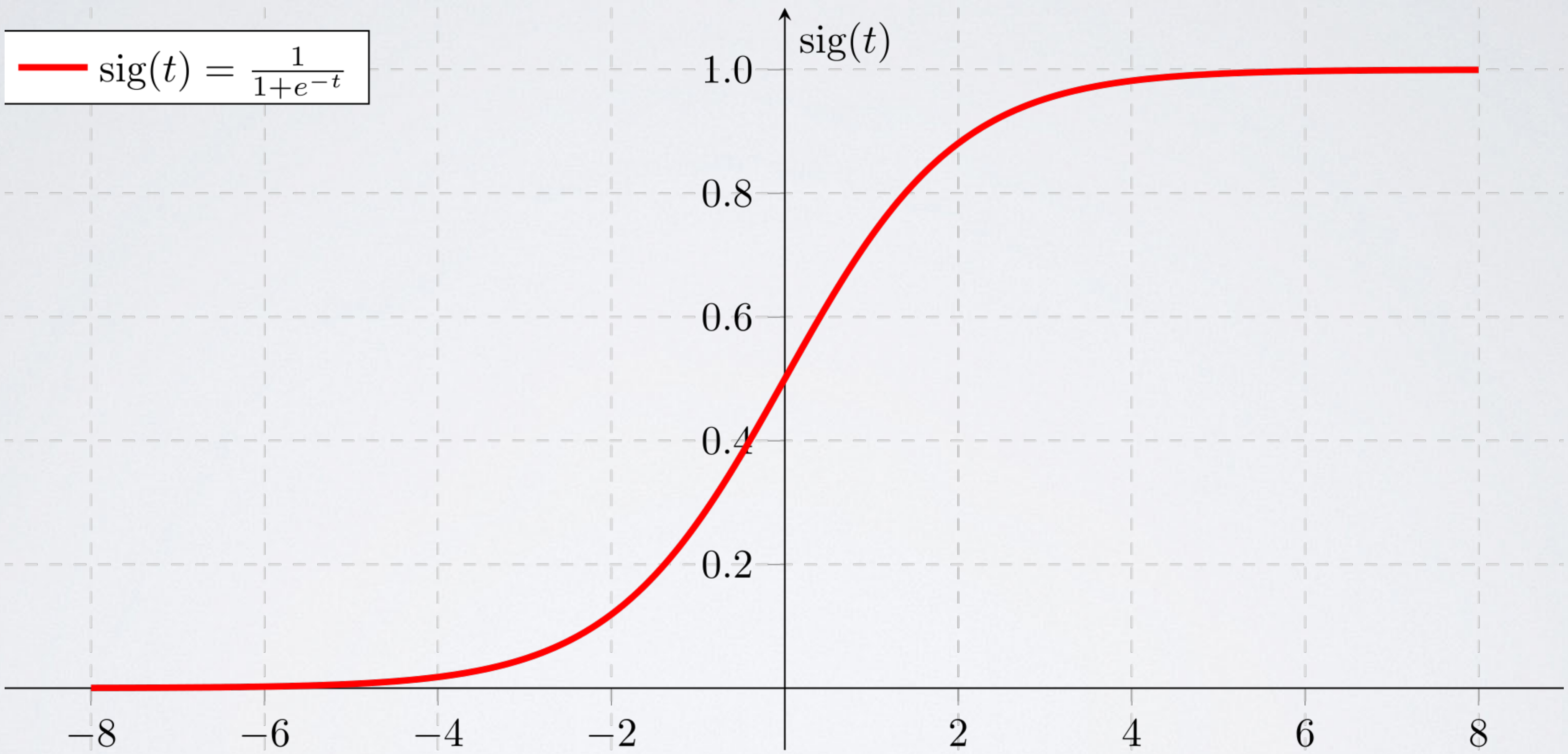
LOGISTIC REGRESSION

- Idea: frame problem of classification as a regression problem
- Learn a function that computes the probability of belonging the classes in our output domain
  - Choose class with highest probability as output
  - Going forward will assume binary case, but results generalise (to an extent)
  - In binary case, $y_i \in \{0,1\}$

LOGISTIC REGRESSION

- Can we use linear regression?
  - No!
  - Probabilities are bounded between 0 and 1, linear regression does not bound output
  - Can we bound output of linear model
    - Yes
    - Use Squashing function

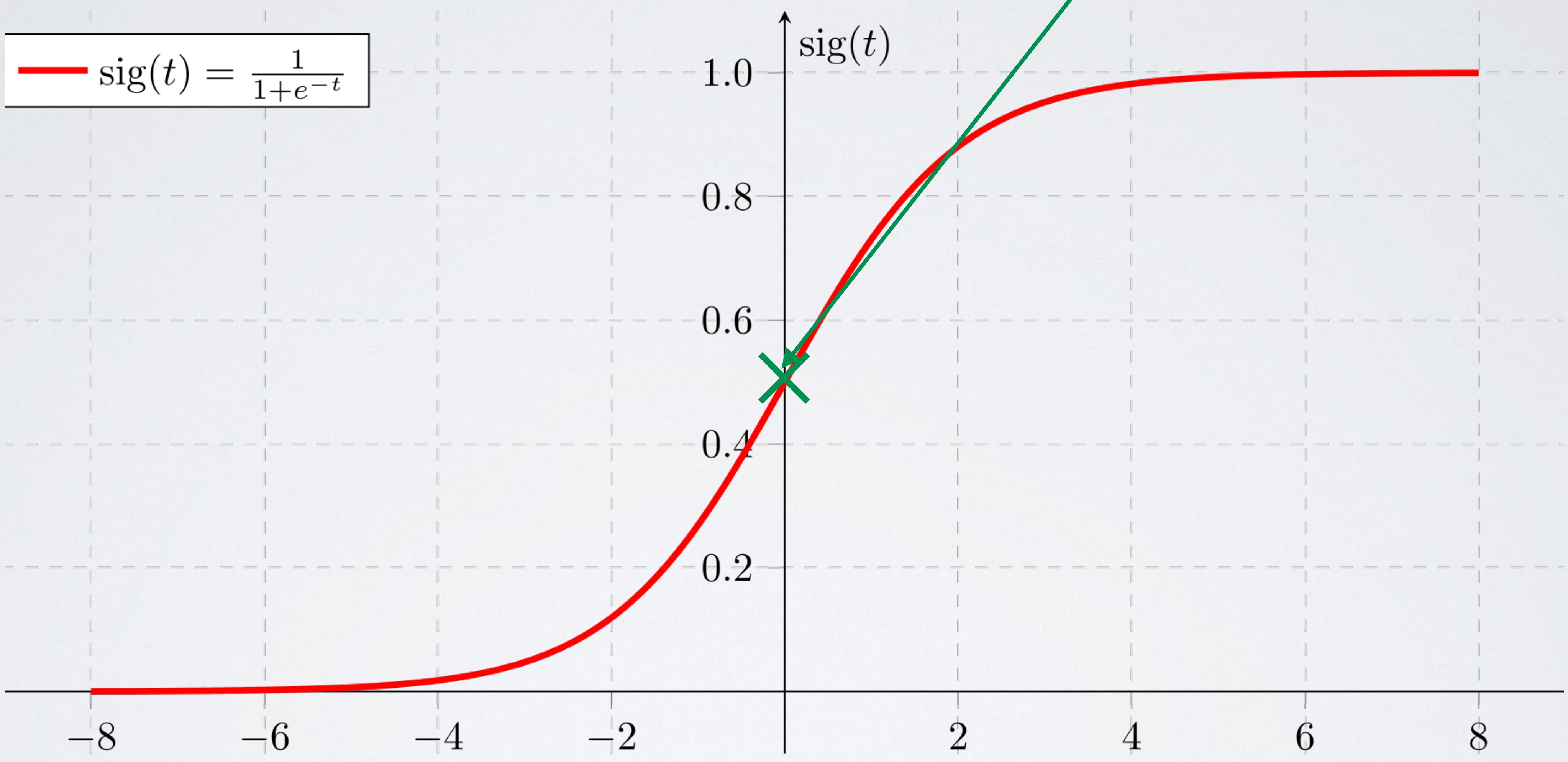
SIGMOID



SIGMOID FUNCTION

- Recall that in regression, we used $f_w(x_i) = w^T x_i = \hat{y}_i$ in regression.
  - Model parameterised by weights, w. Try to find "best" w, i.e. the best fit line
  - We need to squash output using sigmoid
  - So we pass result of dot product through sigmoid
  - $f_w(x_i) = \sigma(w^T x_i) = \hat{y}_i$
  - If $\hat{y}_i \geq \mathbf{0.5}$, return positive class, else return negative class

SIGMOID



0.5 at $t = 0$
$\mathrm{sig}(t) = \frac{1}{1+e^{-t}}$
$\mathrm{sig}(t)$
1.0
0.8
0.6
0.4
0.2
−8
−6
−4
−2
2
4
6
8

DECISION BOUNDARY

- This means that we return the positive class for data point i when $w^T x_i \geq 0$
- Hence, our weights, w, helps establish a decision boundary between positive and negative instances

SIGMOID FUNCTION AND DECISION BOUNDARY

- The function inside of the

LOGISTIC REGRESSION MODEL

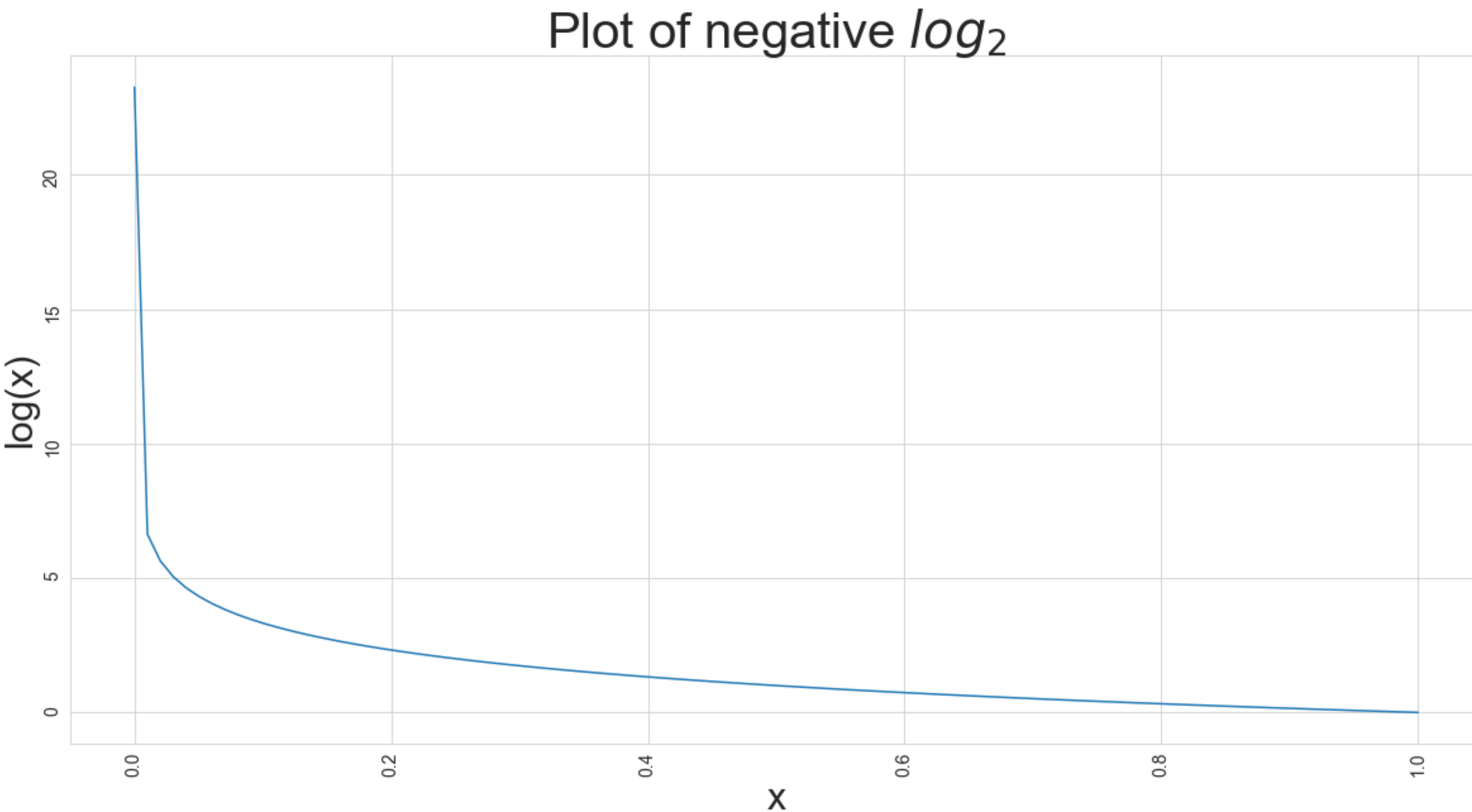
- So we have the form of our model :-)
  - $f_w(x_i) = \sigma(w^T x_i) = \hat{y}_i$
- But now we need to be able to define what makes a model "good".
- Can we use MSE loss?

CROSS ENTROPY

- Strictly speaking we should not
- We are dealing with probabilities. Output class label in binary case can be considered a probability of "yes"
- Measures* of differences between probabilities is well understood problem with good well understood solutions
  - Cross entropy is an example of such a function

CROSS ENTROPY LOSS

- In a binary case, the probability of negative is $1 - \hat{y}_i$
- We want a loss function that rewards correctness and punishes deviation
  - Rewards for higher $\hat{y}_i$ when $y_i = 1$
  - Rewards for higher $1 - \hat{y}_i$ when $y_i = 0$



Plot of negative log_2
log(x)
x
0
5
10
15
20
0.0
0.2
0.4
0.6
0.8
1.0

CROSS ENTROPY LOSS

- Cross entropy loss of example i
  - $-(y_i log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i))$
  - $-(y_i log(f_w(x_i)) + (1 - y_i)log(1 - f_w(x_i)))$
  - $-(y_i log(\sigma(w^T x_i) + (1 - y_i)log(1 - \sigma(w^T x_i)))$

CROSS ENTROPY LOSS

$$-\frac{1}{n}\sum_{i=1}^{n}(y_i log(\sigma(w^T x_i)) + (1 - y_i)log(1 - \sigma(w^T x_i))$$