ML Classification

Agenda

1. Classification in ML
2. Maximum Likelihood
3. Cross Entropy

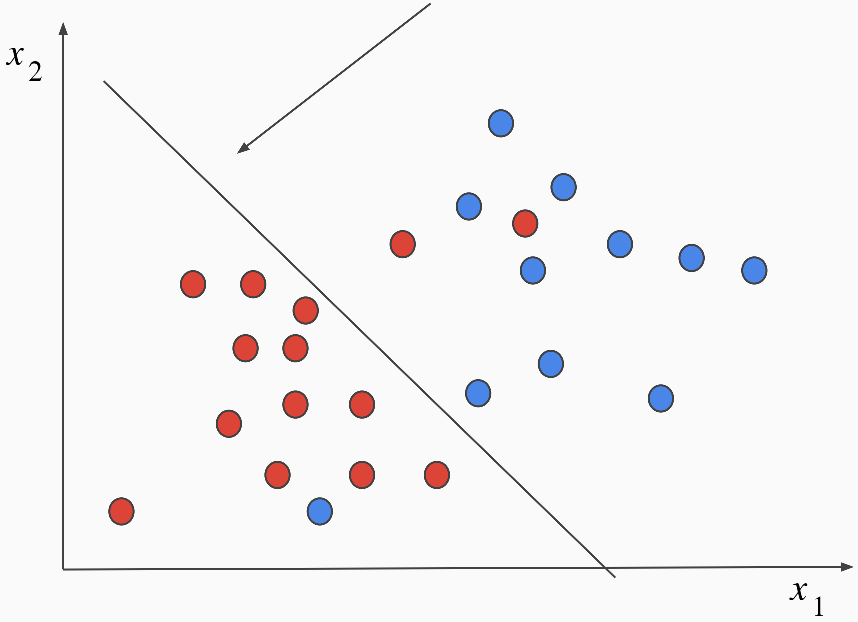
Classification Problems

We need to find a **boundary Line**

$$w_1 x_1 + w_2 x_2 + b = 0$$

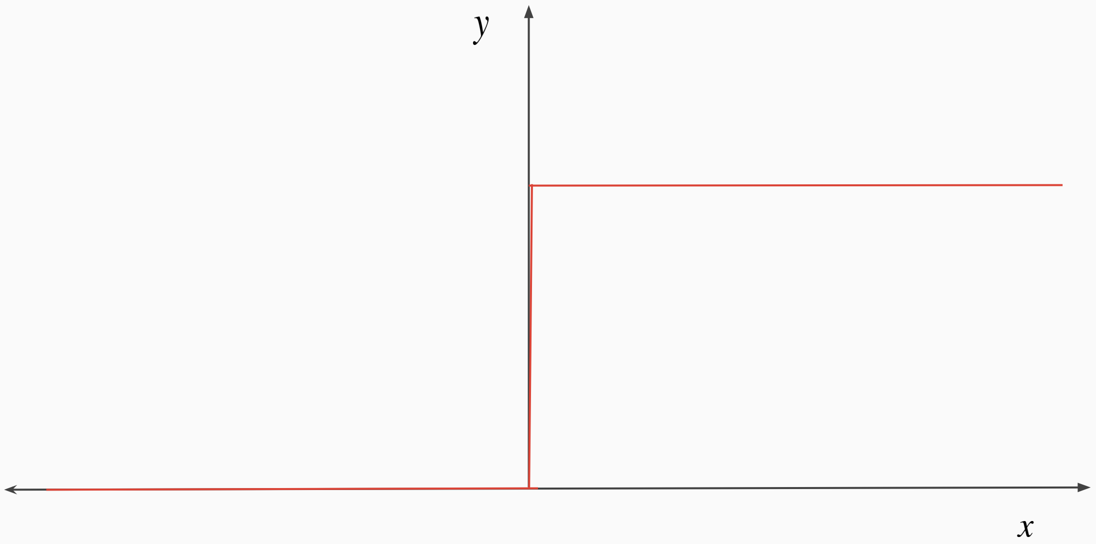
So that we can **predict** the class of data points

$$\hat{y} \begin{cases} 1 \ if \ \ Wx + b \geq 0 \\ 0 \ if \ \ Wx + b < 0 \end{cases}$$

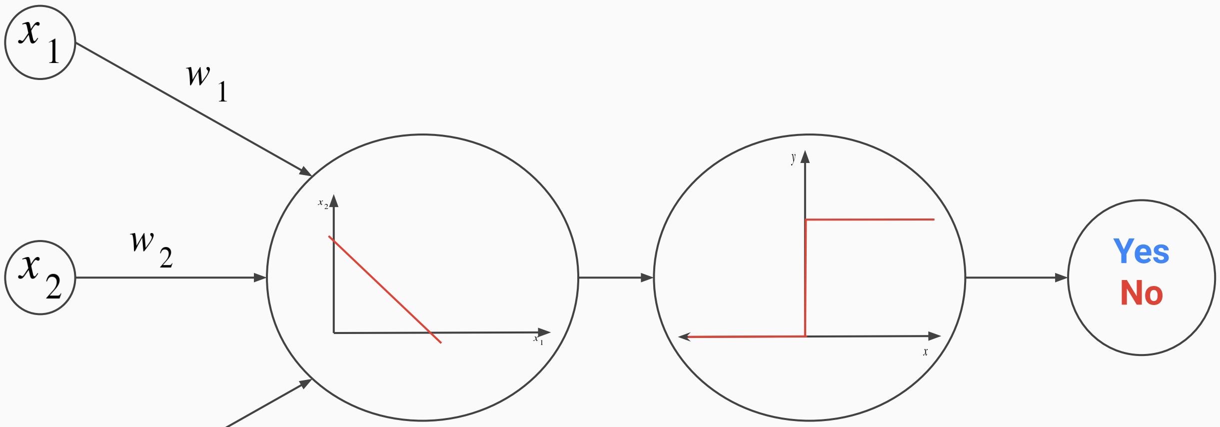


Step Function

$$y \begin{cases} 1 \ \mathit{if} \ \ x \geq \ 0 \\ 0 \ \mathit{if} \ \ x < 0 \end{cases}$$



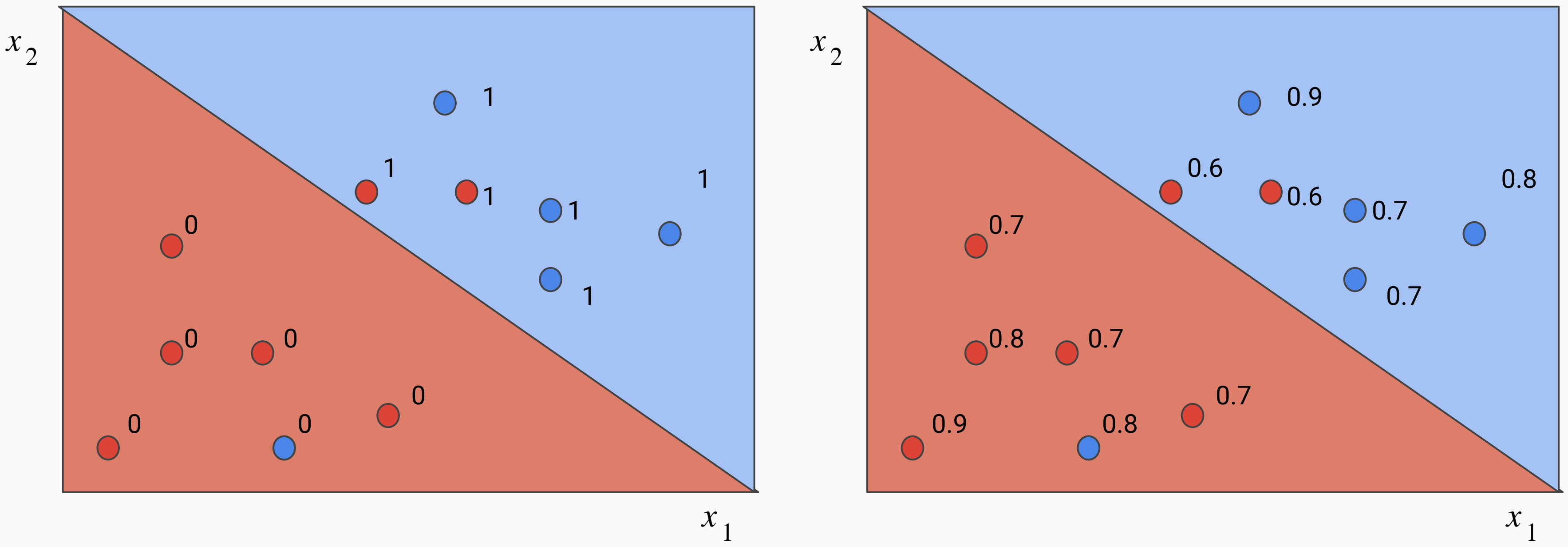
Perceptron



$$w_1x_1 + w_2x_2 + b$$

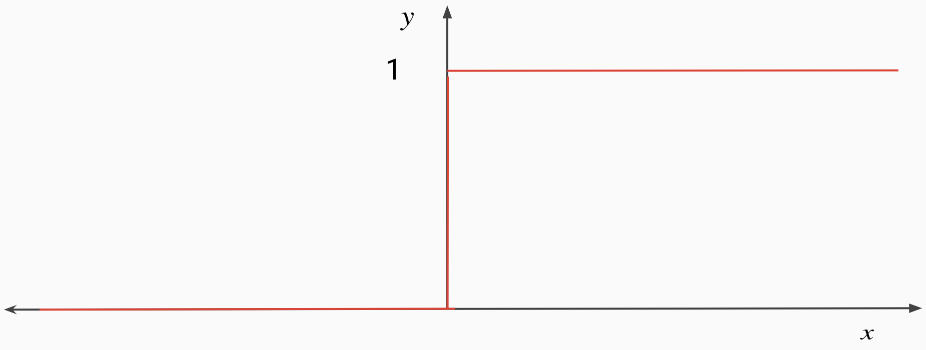
$$\hat{y}\begin{cases} 1 \; if \;\; Wx + b \geq 0 \\ 0 \; if \;\; Wx + b < 0 \end{cases}$$

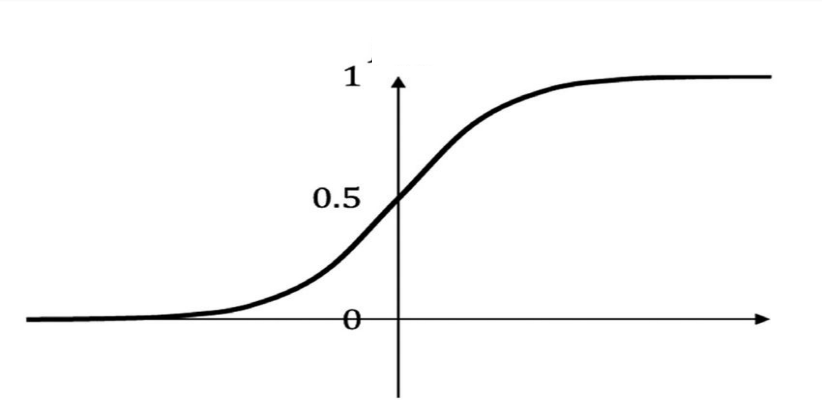
Discrete vs. Continuous



Discrete vs. Continuous

Snippet





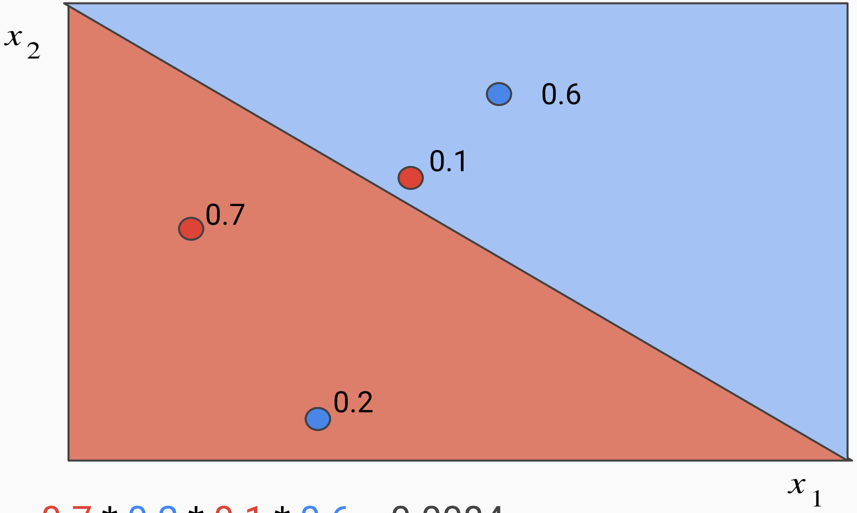
Step Function

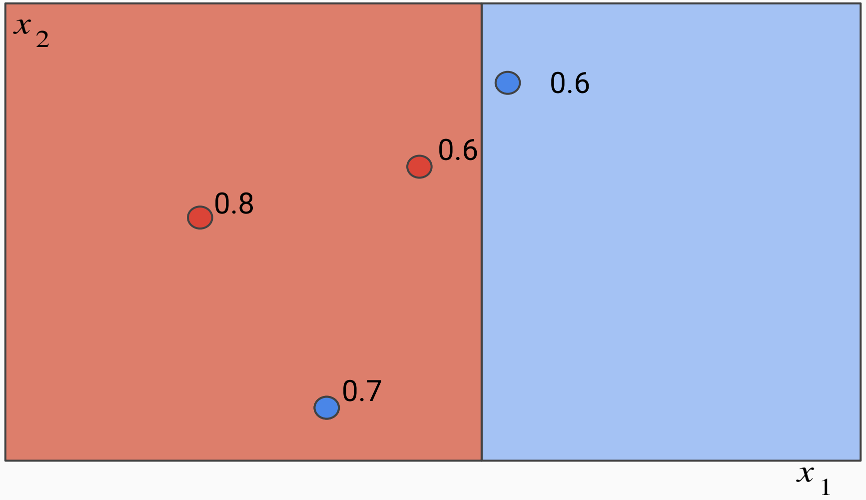
$$y \begin{cases} 1 \; if \;\; x \geq \; 0 \\ 0 \; if \;\; x < \; 0 \end{cases}$$

Sigmoid Function

$$\sigma = \frac{1}{1 + e^{-x}}$$

Maximum Likelihood





0.7 * 0.2 * 0.1 * 0.6 = 0.0084

0.8 * 0.7 * 0.6 * 0.6 = **0.2016**

Products and Sums

The product of thousands of probabilities will create a tiny number.

Changing one of the probabilities will have a big effect on the product.

We want to work with sums instead.

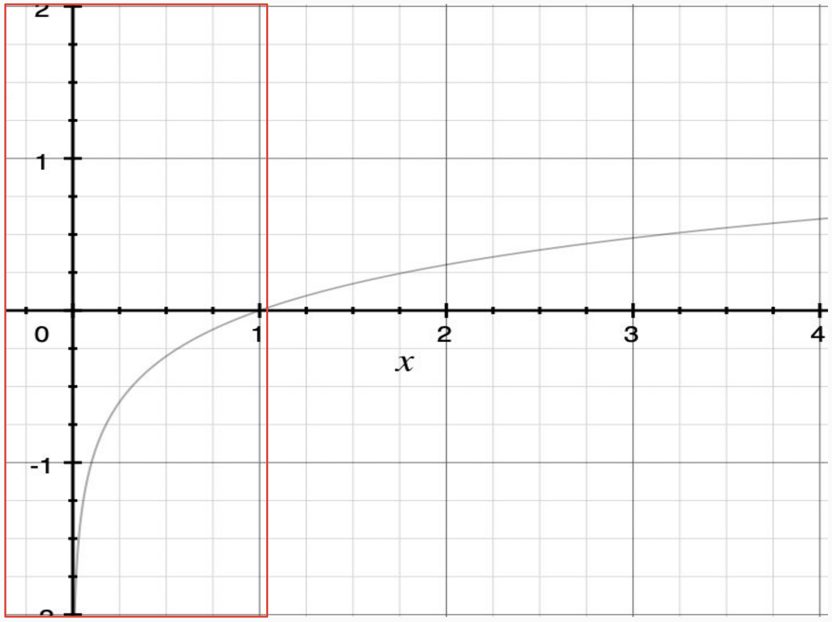
Which function can we use to convert products into sums ?

Logarithm Function

log(ab) = log(a) + log(b)

0.9 * 0.7 * 0.8

log(0.9) + log(0.7) + log(0.8)



Negative Logarithm

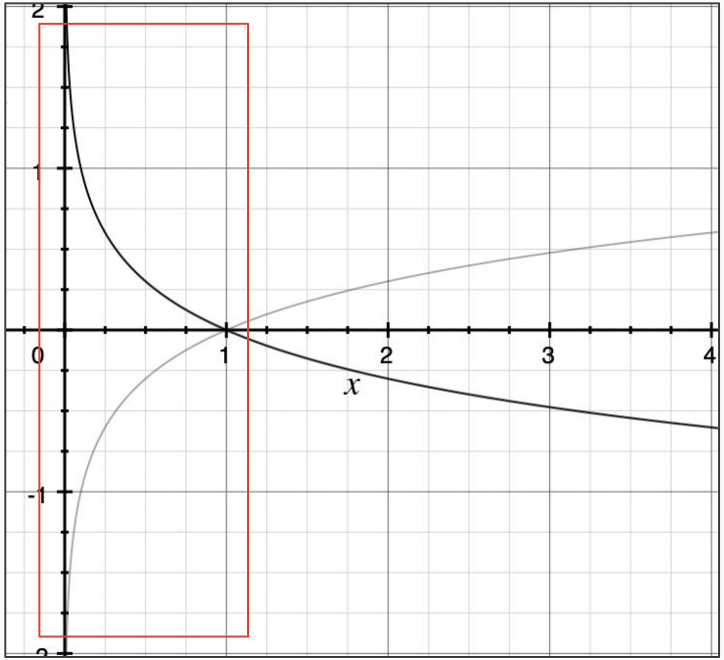
log(ab) = log(a) + log(b)

0.9 * 0.7 * 0.8

log(0.9) + log(0.7) + log(0.8) -> negative number

Optimizers usually work with minimizing a function.

-log(0.9) + -log(0.7) + -log(0.8) -> positive number



Negative Logarithm

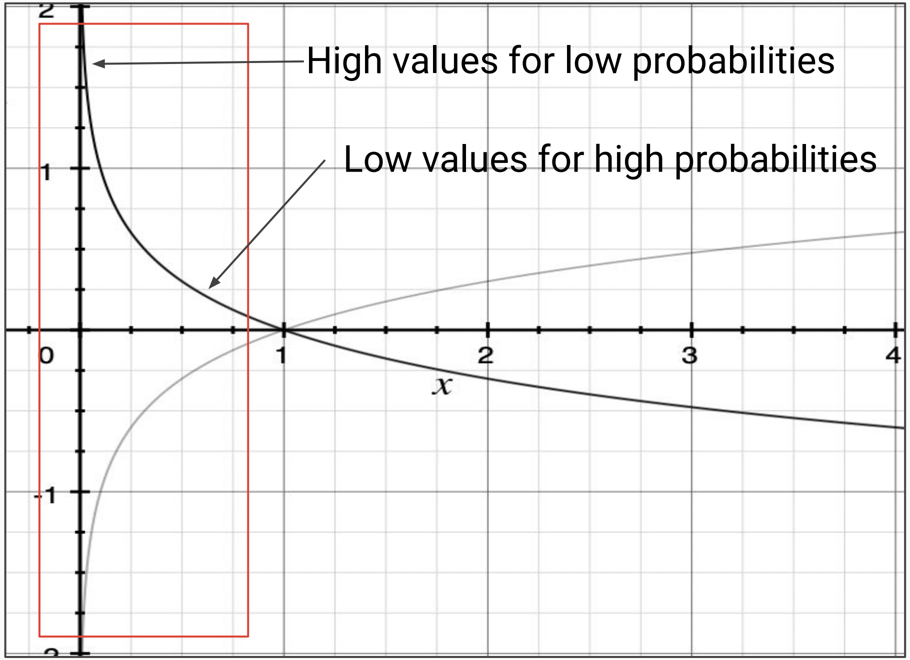
log(ab) = log(a) + log(b)

0.9 * 0.7 * 0.8

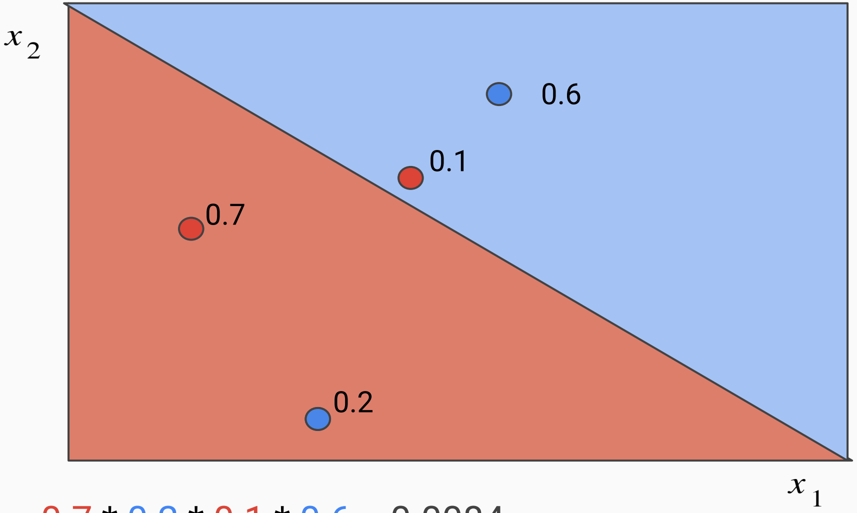
log(0.9) + log(0.7) + log(0.8) -> negative number

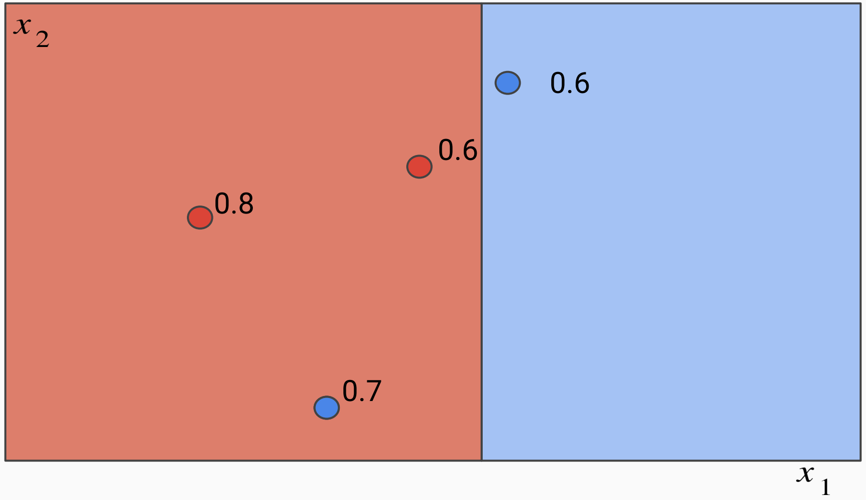
Optimizers usually work with minimizing a function.

-log(0.9) + -log(0.7) + -log(0.8) -> positive number



Cross Entropy





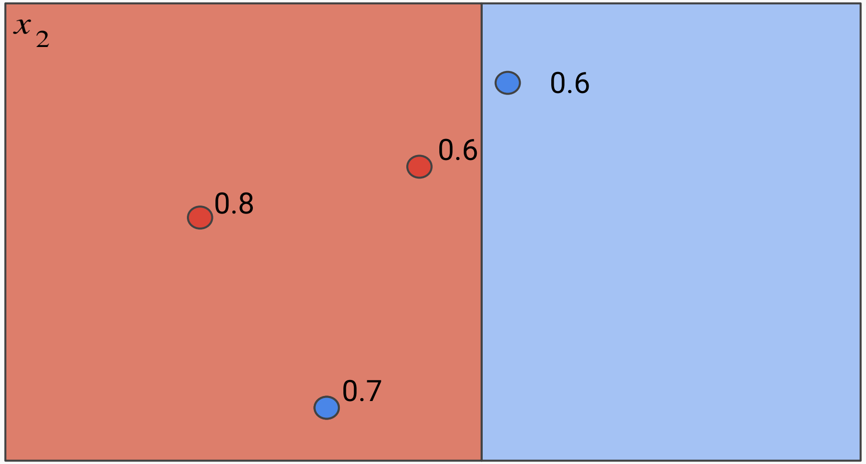
0.7 * 0.2 * 0.1 * 0.6 = 0.0084
- log(0.7) - log(0.2) - log(0.1) - log(0.6) = 4.8

0.8 * 0.7 * 0.6 * 0.6 = **0.2016**
- log(0.8) - log(0.7) - log(0.6) - log(0.6) = **1.601**

Cross Entropy

If y=1, P(blue) = $\hat{y}$

Error = -log($\hat{y}$)



- log(0.8) - log(0.7) - log(0.6) - log(0.6) = **1.601**

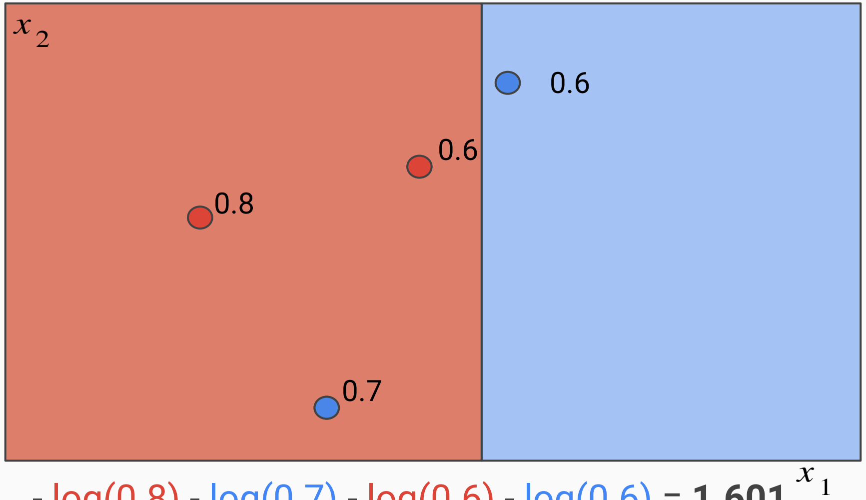
Cross Entropy

If $y=1$, P(blue) = $\hat{y}$

Error = $-\log(\hat{y})$

If $y = 0$, P(red) = 1 - P(blue) = $1 - \hat{y}$

Error = $-\log(1 - \hat{y})$



$- \log(0.8) - \log(0.7) - \log(0.6) - \log(0.6) = \mathbf{1.601}$

Cross Entropy

Error = - (1 - y)log(1 - $\hat{y}$) - y log ($\hat{y}$)

Error = $-\frac{1}{m}\sum_{i=1}^{m}$ - (1 - y)log(1 - $\hat{y}$) - y log ($\hat{y}$)

Error = $-\frac{1}{m}\sum_{i=1}^{m}$ - (1 - y)log(1 - $\sigma(Wx^i + b)$) - y log ($\sigma(Wx^i + b)$)