Day 3: Logistic Regression and Evaluation

Review: Linear Regression

Goal: Find line of best fit 

How? Learn θ which minimize sq. error

How? Find θ which minimize MSE loss

How? Find $\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

$$\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

$$= \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \text{distance}(y_i, \hat{y}_i)$$

How? will cover later

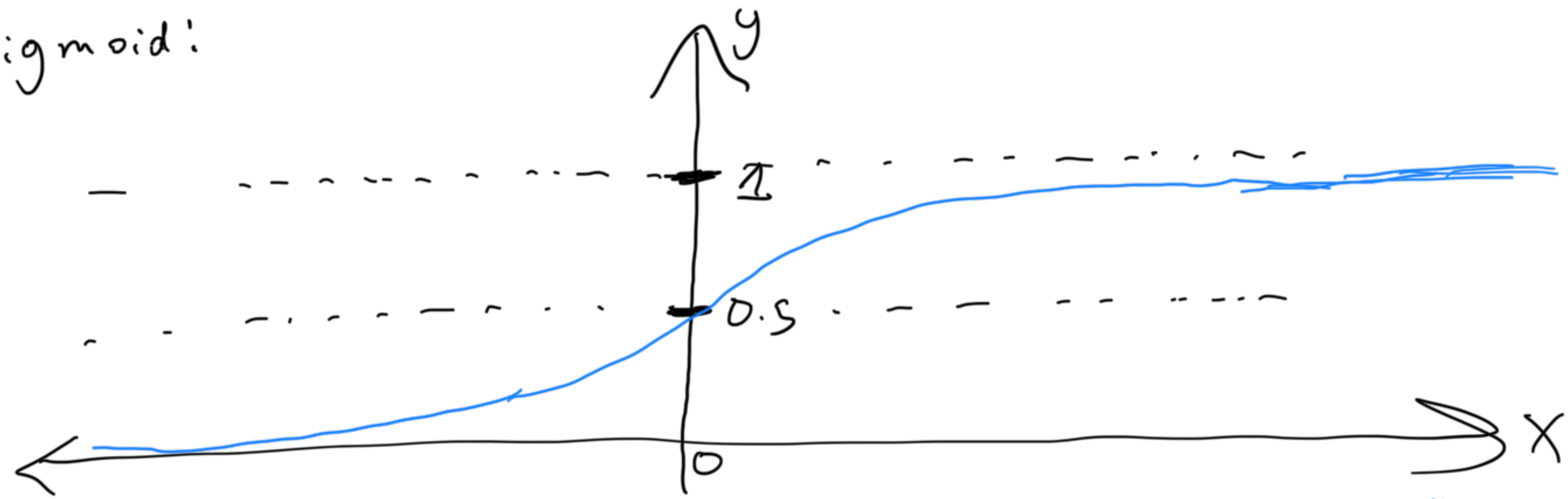
First Classification model

Logistic Regression (LR)

For binary classification, output the probability that class = 1

<u>Class</u>: different things you can predict in classification problems

Sigmoid:



$$\text{Sigmoid}(x) = s(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

Useful Properties of sigmoid function:

* output is between 0 and 1

(i.e., a probability)

* confidence (probability) increases more quickly near the middle (i.e., when X is near 0 and more slowly away from the middle

Logistic Regression:

Learn optimal θ (i.e., m's and b) for the function:

$$y = S(mx + b) = \frac{1}{1 + e^{-(mx+b)}}$$

Or more generally:

$$y = S(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)}}$$

Optimization Goal:

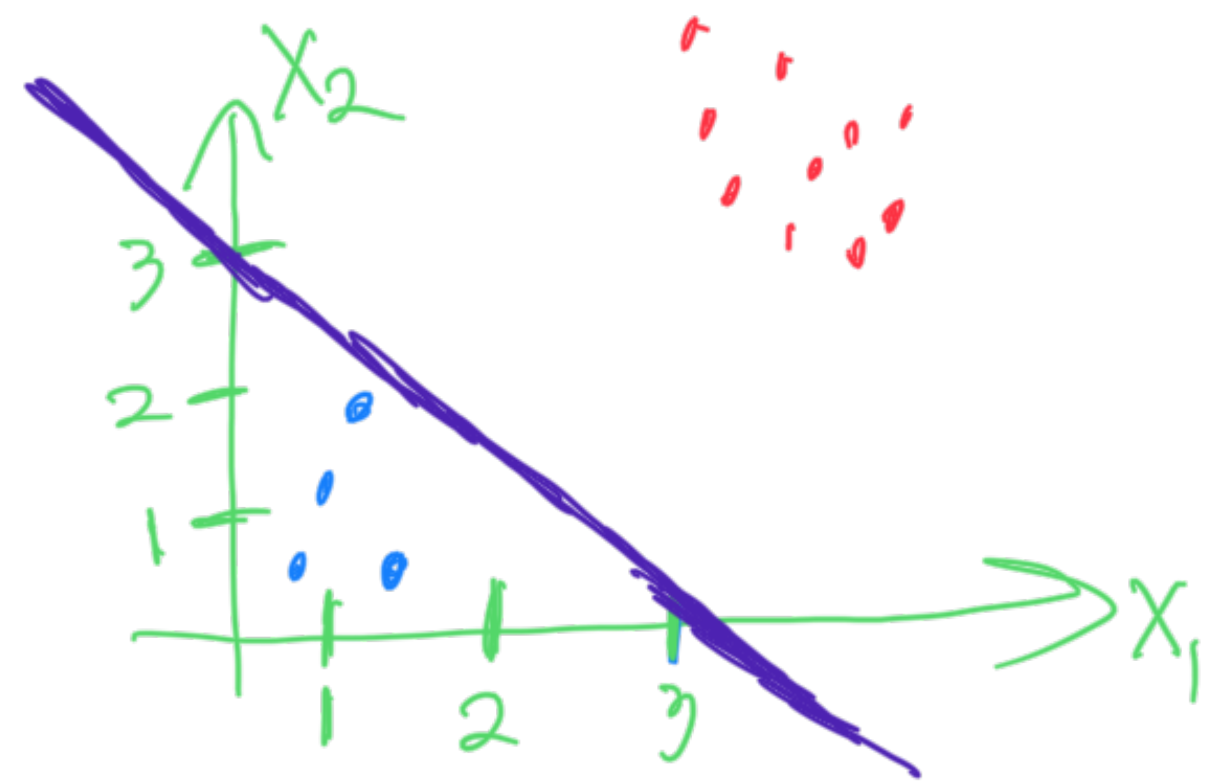
Find $\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \text{distance}(y_i, \hat{y}_i)$

$$= \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \text{distance}\left(y_i, \frac{1}{1+e^{-(\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n)}}\right)$$

0 or 1

between 0 and 1

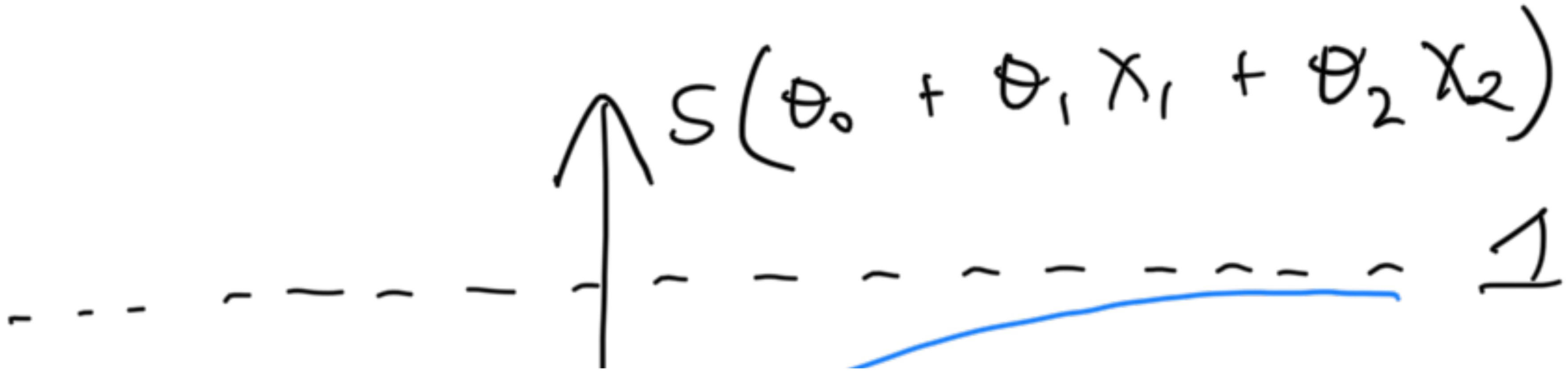
Example:

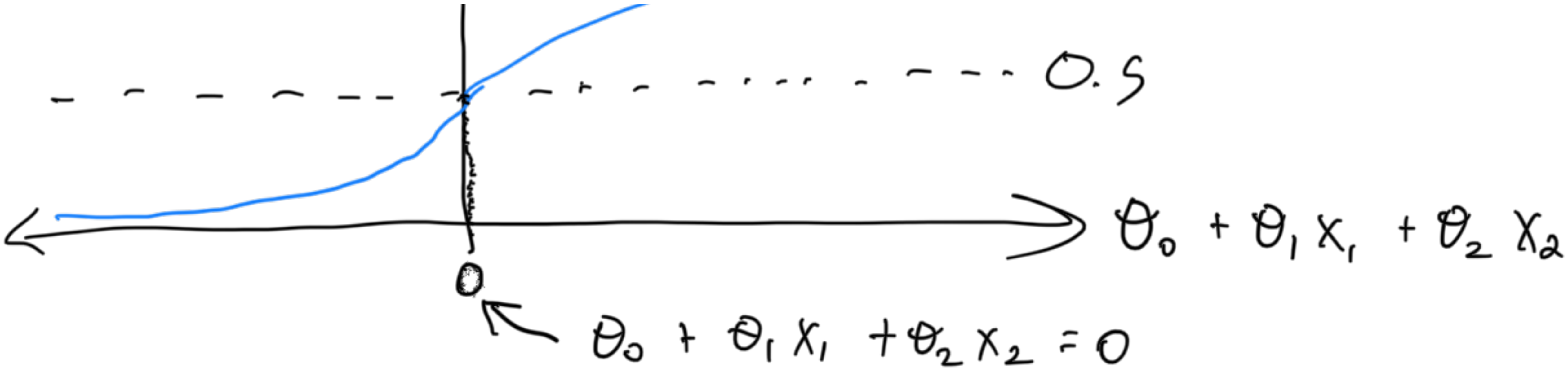


Let's say the Learned model is:

$$y = s(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$





If <u>decision boundary</u> = 0.5 ← set by ML engineer

(i.e., predict 1 if $\hat{y} \geq 0.5$, otherwise predict 0)

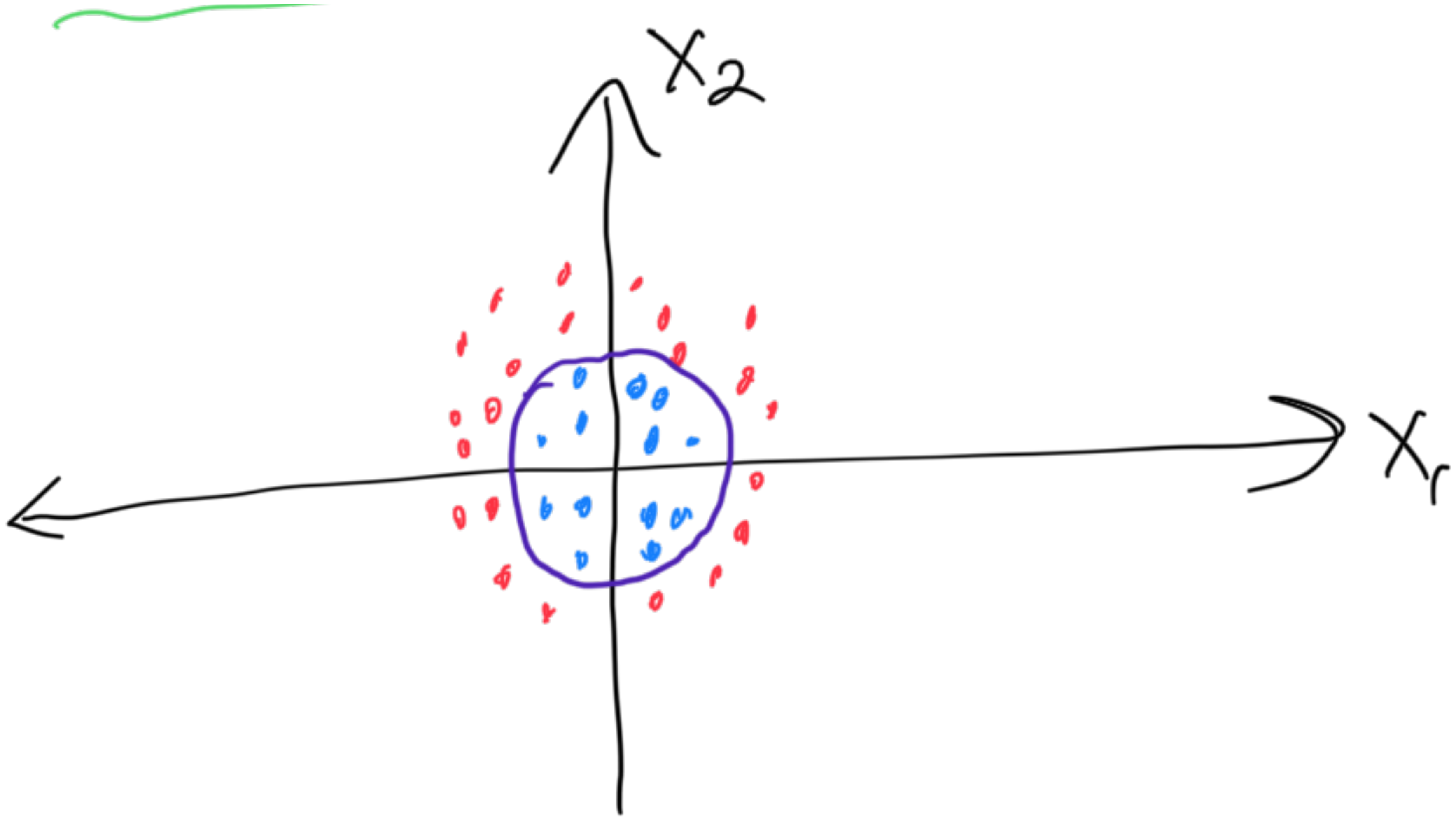
$$\underbrace{\theta_0}_{} \quad \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

X-axis in sigmoid plot above

Predict "y = 1" if $-3 + 1 \cdot x_1 + 1 \cdot x_2 \geq 0$

→ $\underbrace{x_1 + x_2 = 3}$ is the decision boundary

Example of Non-Linear Decision Boundary:



Learned model:

$$y = S(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

$$\theta = [-1 \quad 0 \quad 0 \quad 1 \quad 1]$$

Predict "$y=1$" if $-1 + x_1^2 + x_2^2 \geq 0$

$\hookrightarrow$ $x_1^2 + x_2^2 = 1$ is the decision bounary

---

Evaluating Supervised ML

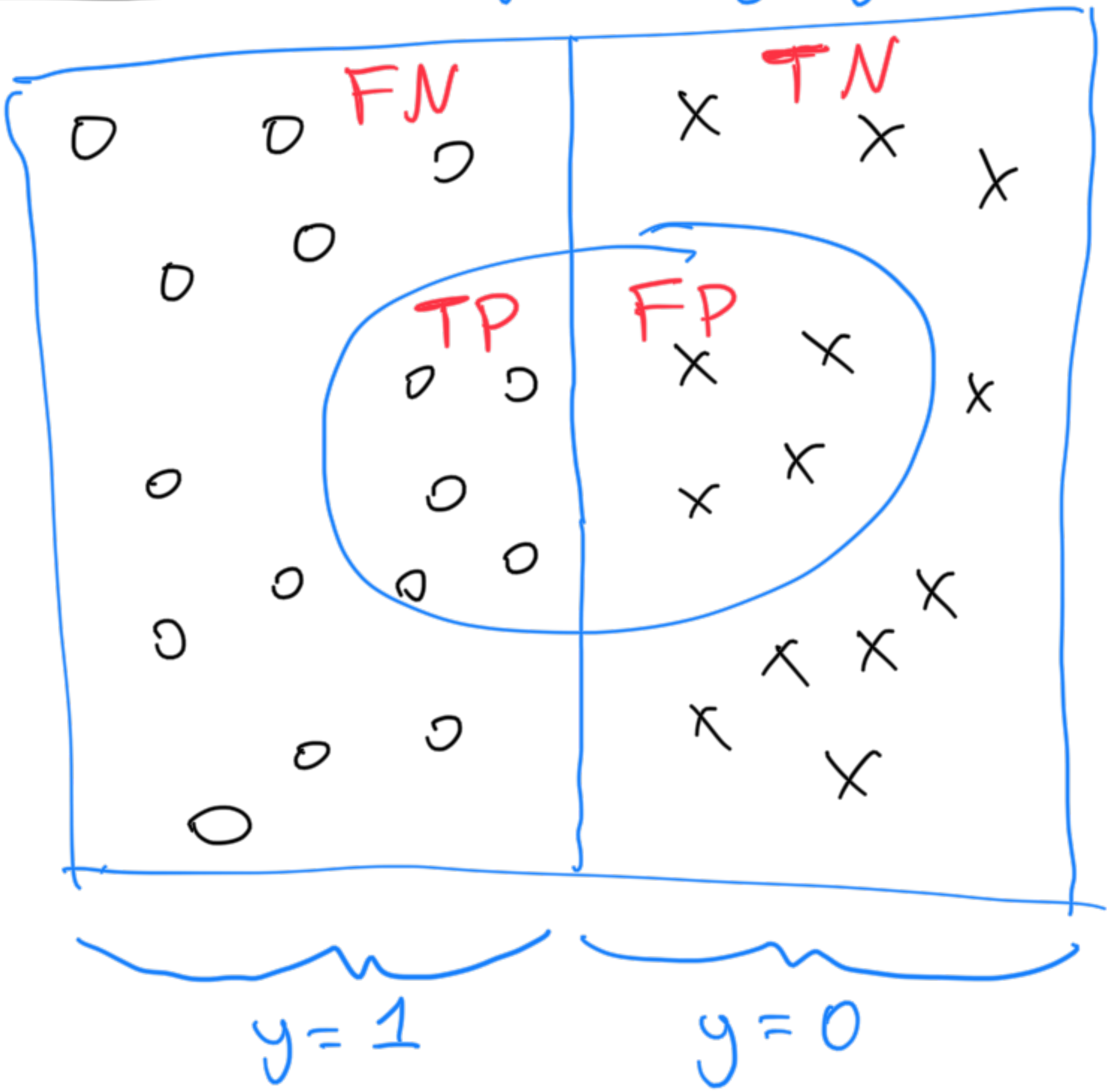
Regression: MSE satisfies the properties of a good loss function, and it's interpretable

$RMSE = \sqrt{MSE}$ → Sometimes people prefer this so

$RMSE = \sqrt{MSE}$ that you have original data units

Classification:

Venn Diagram:



TP = true positive

FP = false positive

FN = false negative

TN = true negative

O = true value is "positive" (i.e., $y = 1$)

X = true value is "negative" (i.e., $y = 0$)

inside circle: $\hat{y} = 1$

outside circle: $\hat{y} = 0$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Problem: What is the accuracy here?

(a classification model is called a "classifier")

y	$\hat{y}$
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
0	0
1	0

```
def classifier(X1, X2):
    return 0
```

Accuracy = 90%

$$\text{Precision}: \frac{TP}{TP + FP} = \frac{\text{correctly predicted positives}}{\text{predicted positives}}$$

(Positive Predictive Value)

Recall (Sensitivity) : $$\frac{TP}{TP + FN} = \frac{\text{Correctly predicted positives}}{\text{actual positives}}$$

Specificity : $$\frac{TN}{TN + FP} = \frac{\text{correctly predicted negatives}}{\text{actual negatives}}$$

Negative Predictive Value : $$\frac{TN}{TN + FN} = \frac{\text{correctly predicted negatives}}{\text{predicted negatives}}$$

Confusion Matrix :

$\hat{y}$

	0	1
	# TN	# FP

$\hat{y}$

	0	1	2	3
0				
1				

y	0	# TN	
	1	# FN	# TP

Multi-Class Classification

Macro Averaging

		y		
		0	1	2
$\hat{y}$	0	8	10	1
	1	5	60	50
	2	3	30	200

$$\text{Precision}_0 = 8/(8+10+1)$$

$$\text{Precision}_1 = 60/(5+60+50)$$

$$\text{Precision}_2 = 200/(3+30+200)$$

$$\text{Recall}_0 = 8/(8+5+3)$$

Overall Precision: $\dfrac{\text{Precision}_0 + \text{Precision}_1 + \text{Precision}_2}{3}$

$$= \boxed{0.60}$$

Micro Averaging

"Pooled" Confusion matrix:

	y "yes"	y "no"	
$\hat{y}$ "yes"	268 $(=200 + 60 + 8)$	99	$\rightarrow 11 + 55 + 33$
$\hat{y}$ "no"	99	635	$\rightarrow 340 + 212 + 83$
	$\rightarrow 8 + 40 + 51$		

$$\text{Overall precision} = \frac{268}{268 + 99} = \boxed{0.73}$$

* In practice, people like Macro Averaging

... Micro skews the result towards

because Micro sways the

the majority class