Classification

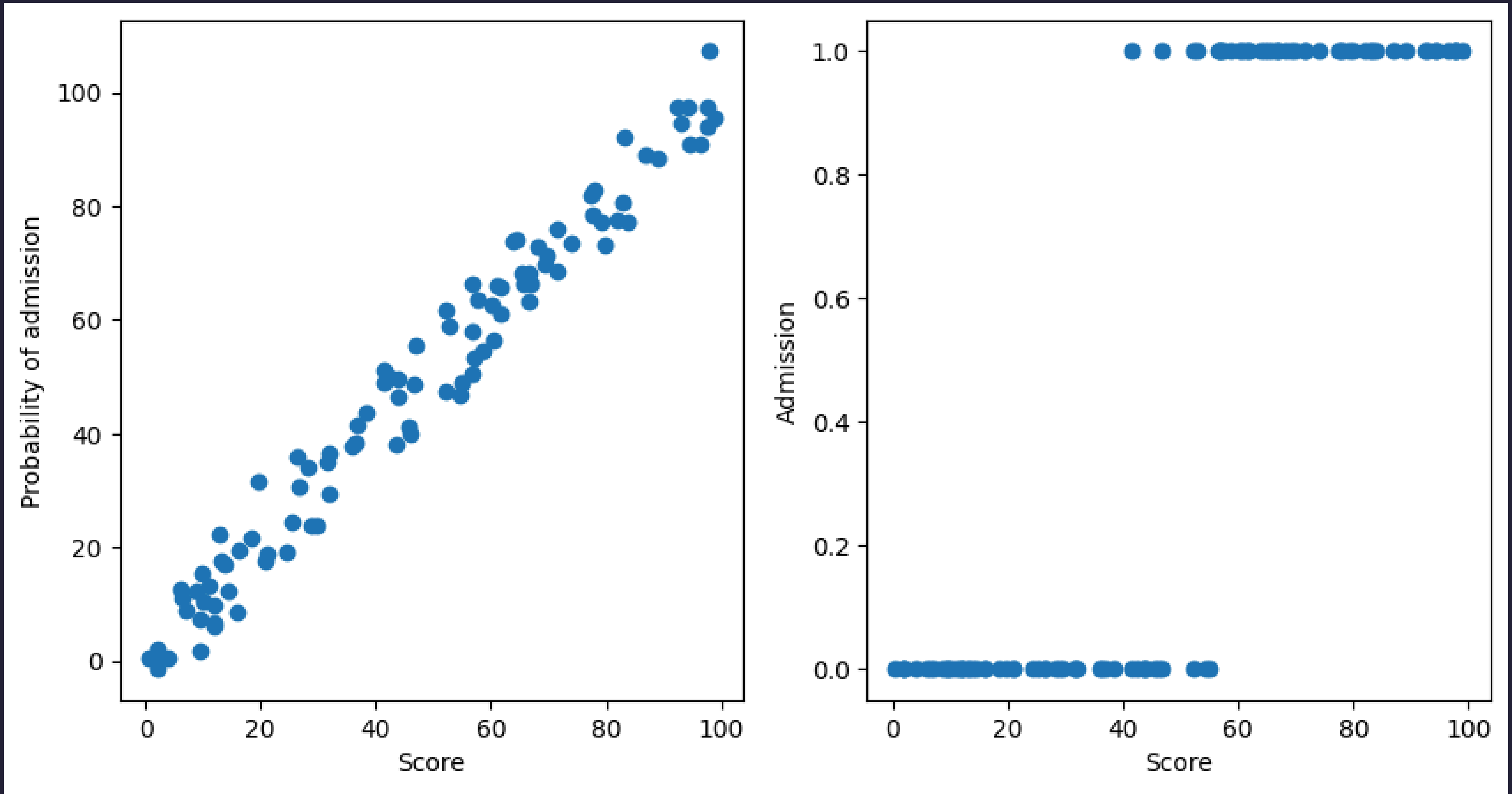
Regression

predict a number. Many possible outputs.

Classification

predict categories. Small number of possible outputs.

College admission

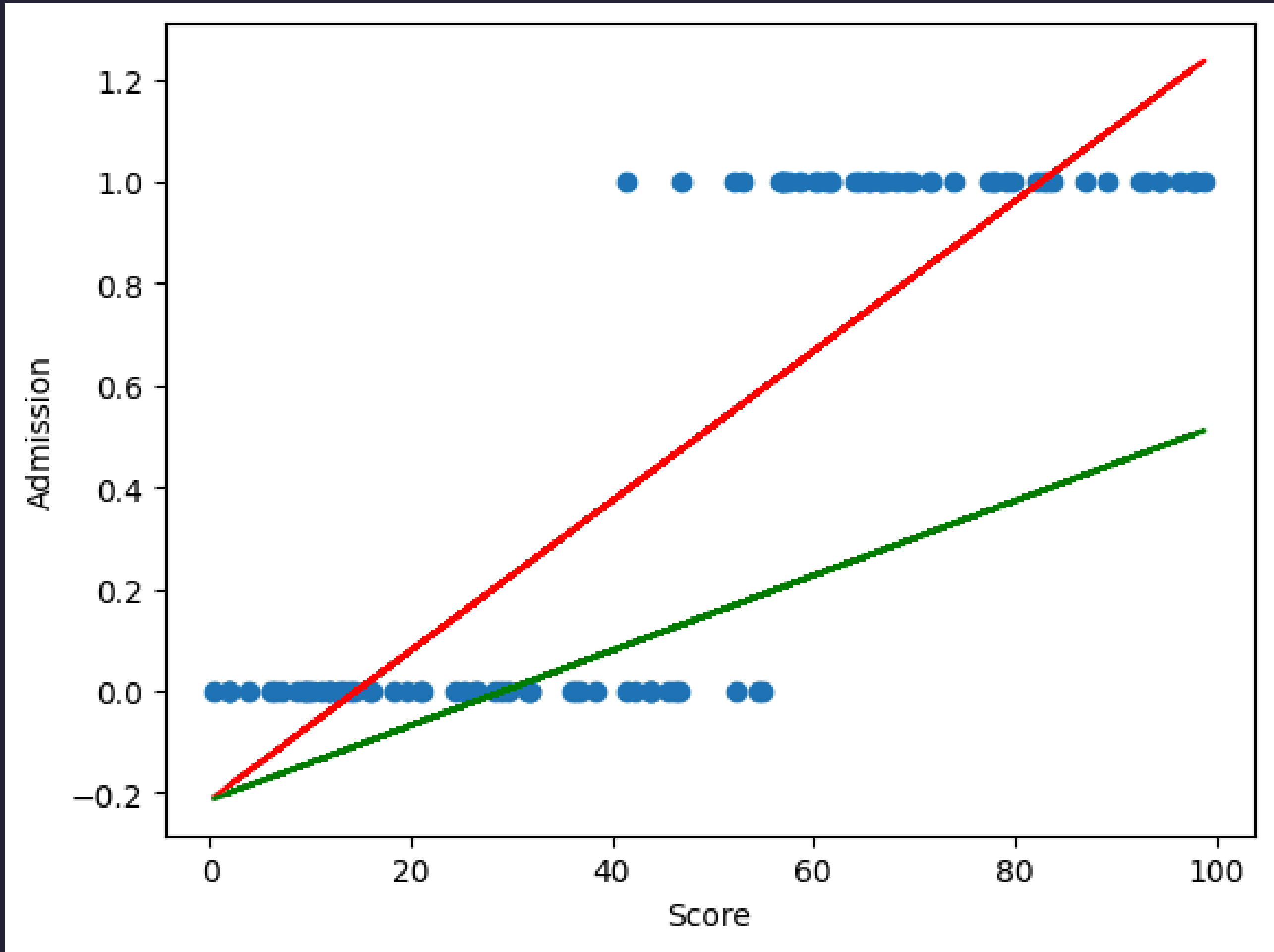


Why not use linear regression for classification?

$f(x) = wx + b$

if $f(x) \geq 0.5$, predict class 1

if $f(x) < 0.5$, predict class 0

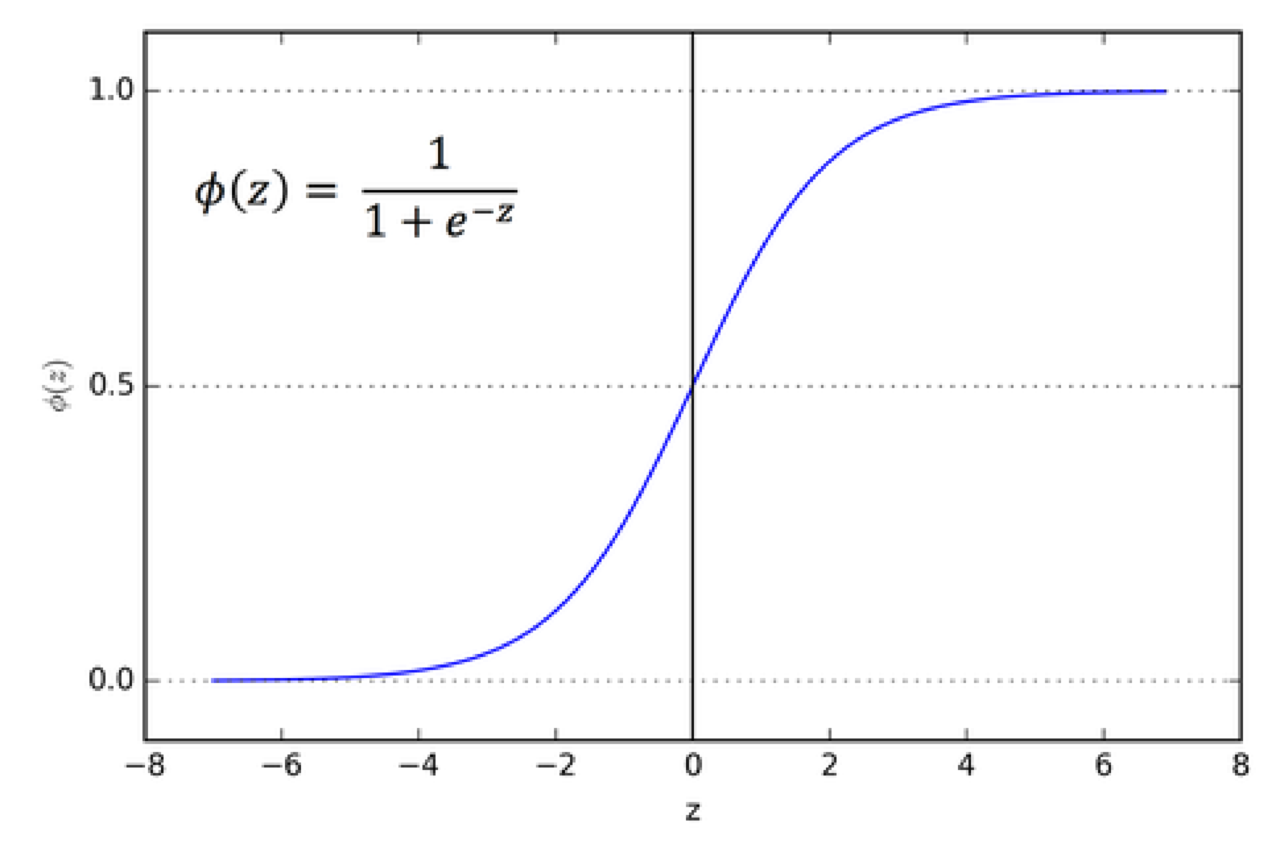


Logistic (sigmoid) function

$$g(z) = \frac{1}{1+e^{-z}}$$

$$z \to \infty, g(z) = \tfrac{1}{1+e^{-\infty}} = 1$$
$$z \to \infty, g(z) = \tfrac{1}{1+e^{\infty}} = 0$$
$$z = 0, g(z) = \tfrac{1}{1+e^{0}} = 0.5$$

$$0 \leq g(z) \leq 1$$



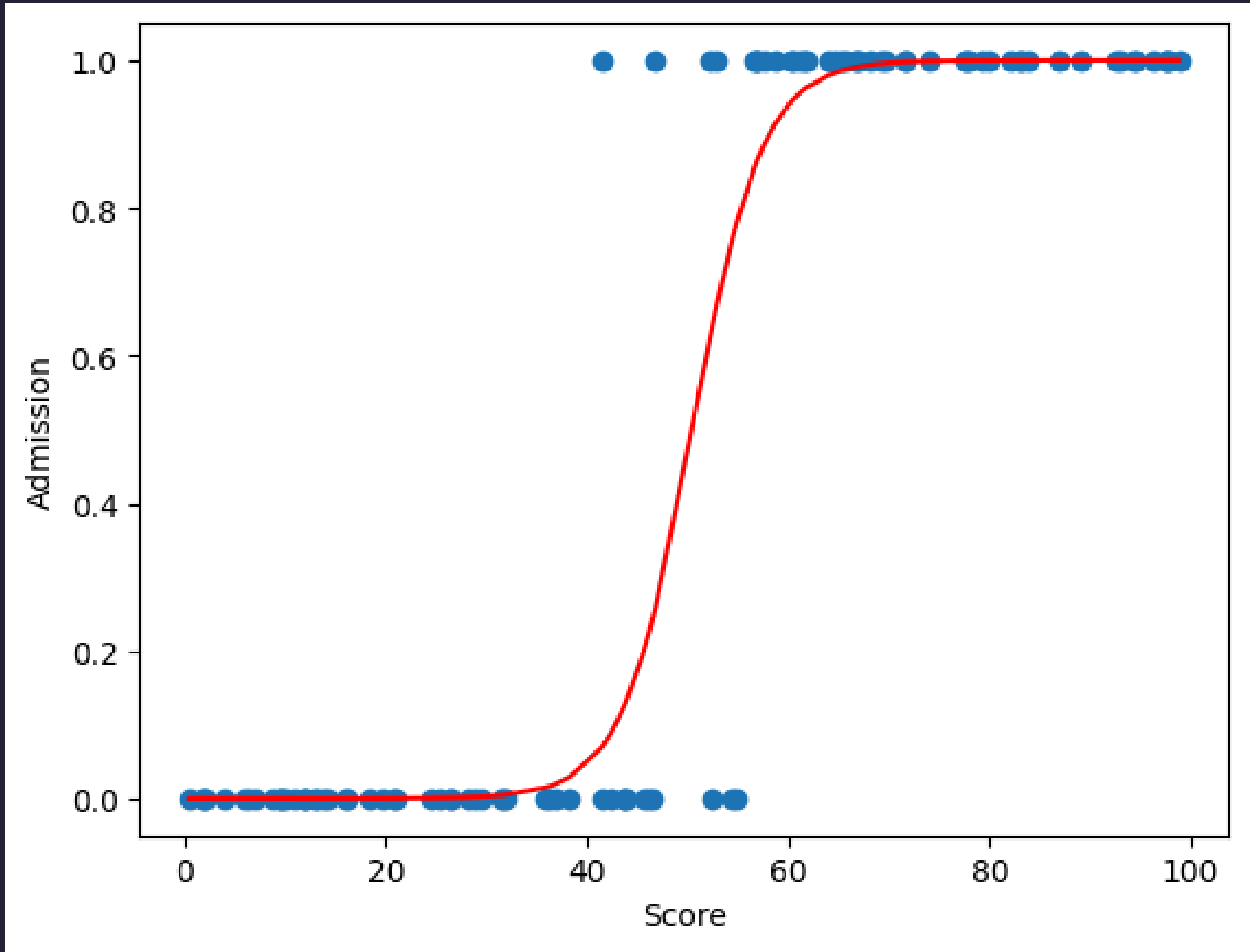
Logistic regression

$$f(x) = g(z) = g(wx + b) = \frac{1}{1 + e^{-(wx+b)}}$$

$$f(55) = 0.7$$

probability for 55 to be admitted: 0.7

probability for 55 to be rejected: 0.3



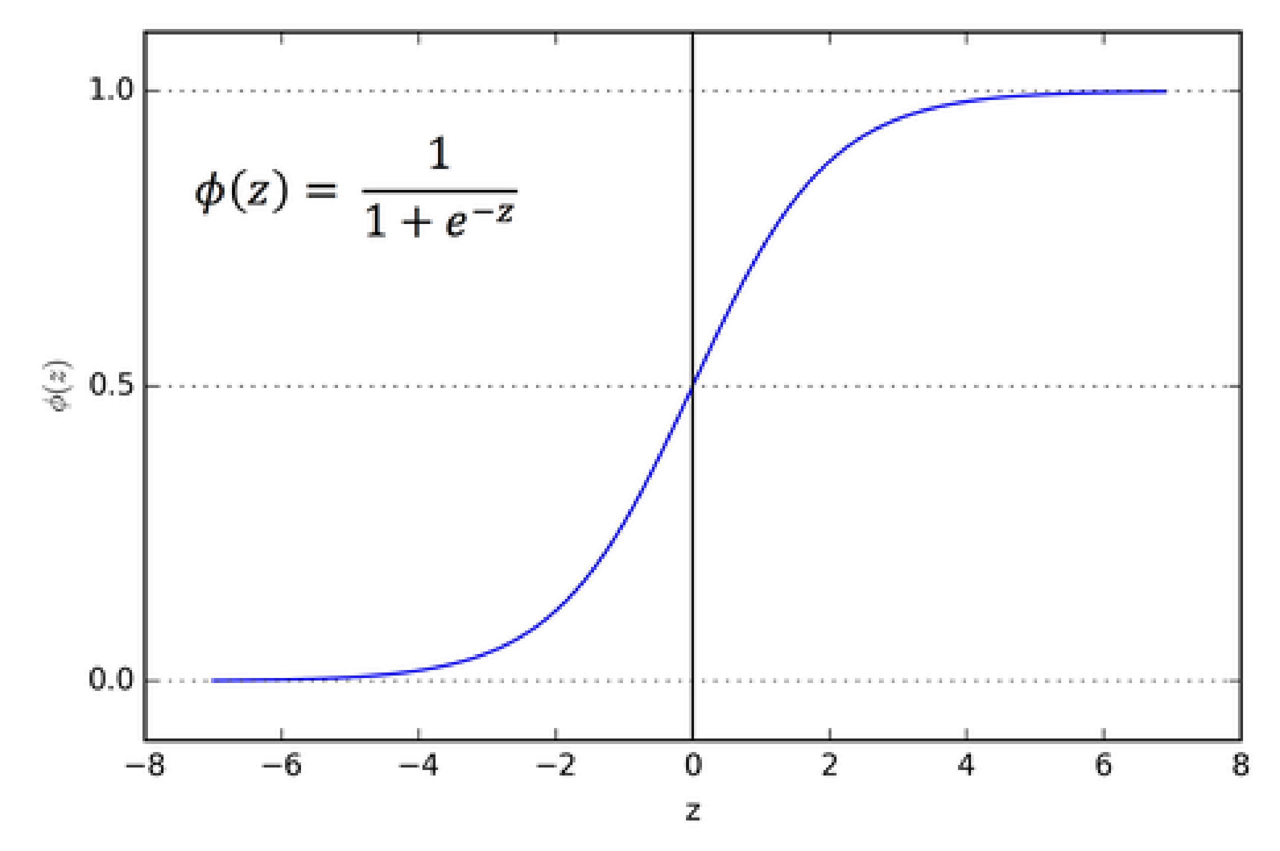
Is it class 1 or class 0?

$$f(x) = g(z) = P(y = 1|x; w, b) = 0.7$$

- $g(z) \geq 0.5$
- $z \geq 0$
- $wx + b \geq 0$

Decision boundary: threshold that separates the two classes

- $z = 0$



Linear decision boundary

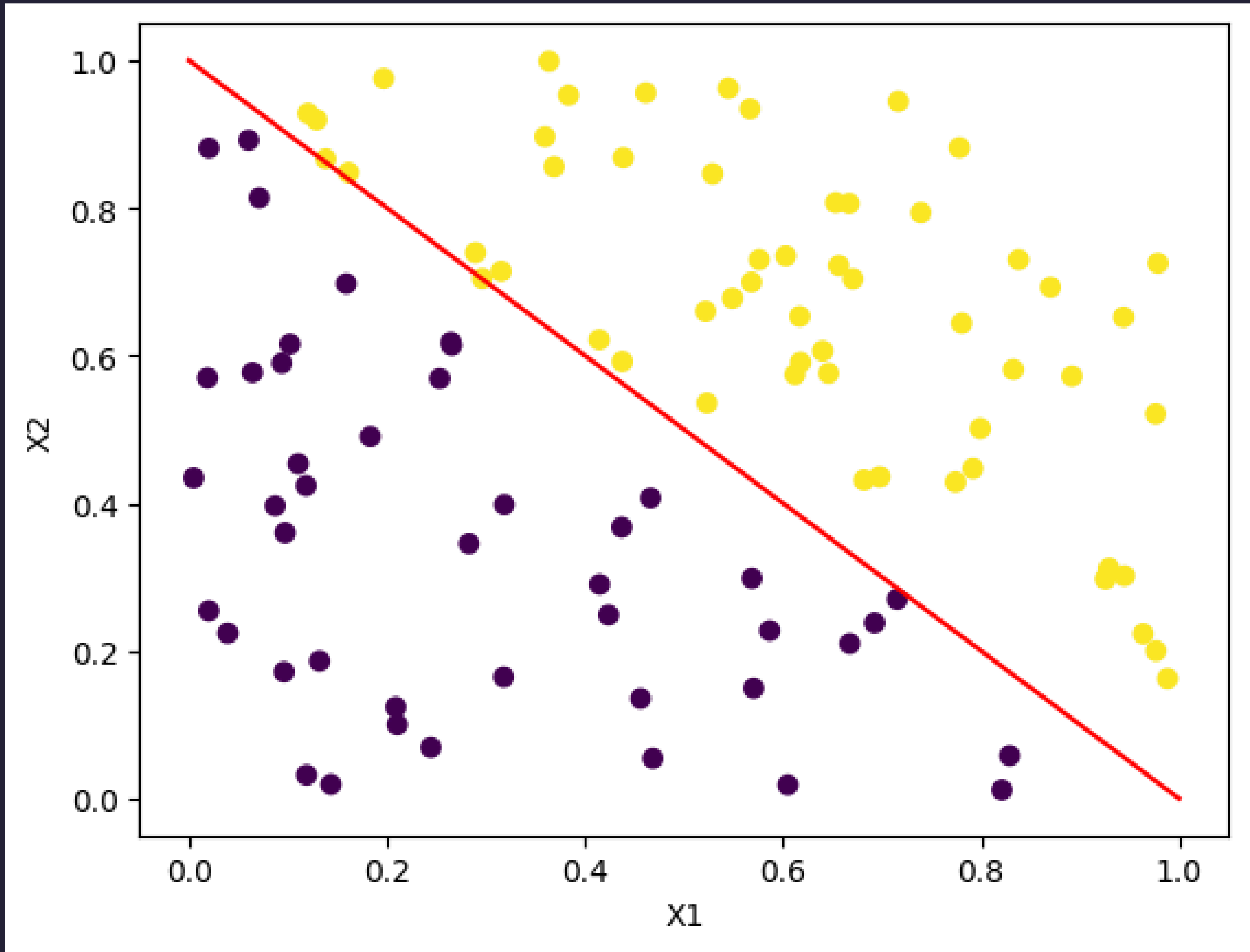
Two features: x_1 and x_2

w=[1,1], b=-1

$$f(x) = g(w_1 x_1 + w_2 x_2 + b)$$
$$= g(x_1 + x_2 - 1) = 0.5$$

$$z = x_1 + x_2 - 1 = 0$$

$$x_1 + x_2 = 1$$



Non-linear decision boundary

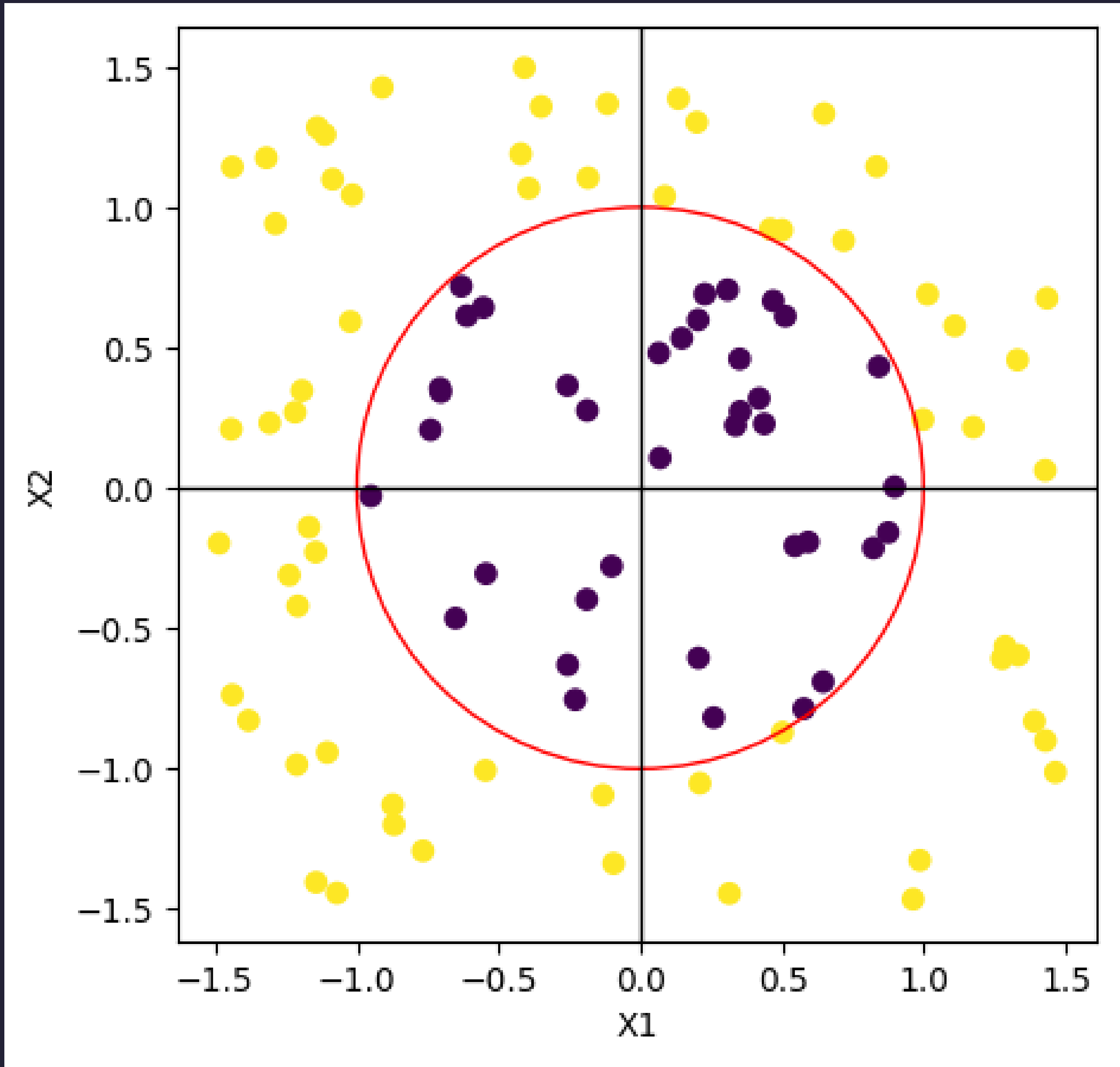
Two features: x_1 and x_2

w=[1,1], b=-1

$$f(x) = g(w_1 x_1^2 + w_2 x_2^2 + b)$$
$$= g(x_1^2 + x_2^2 - 1) = 0.5$$

$$z = x_1^2 + x_2^2 - 1 = 0$$

$$x_1^2 + x_2^2 = 1$$



💻 Logistic regression

Loss for logistic regression

$L(y, f(x))$

if $y = 1$,

- $L(1, f(x))$ should be small when $f(x)$ is close to 1
- $L(1, f(x))$ should be large when $f(x)$ is close to 0

if $y = 0$,

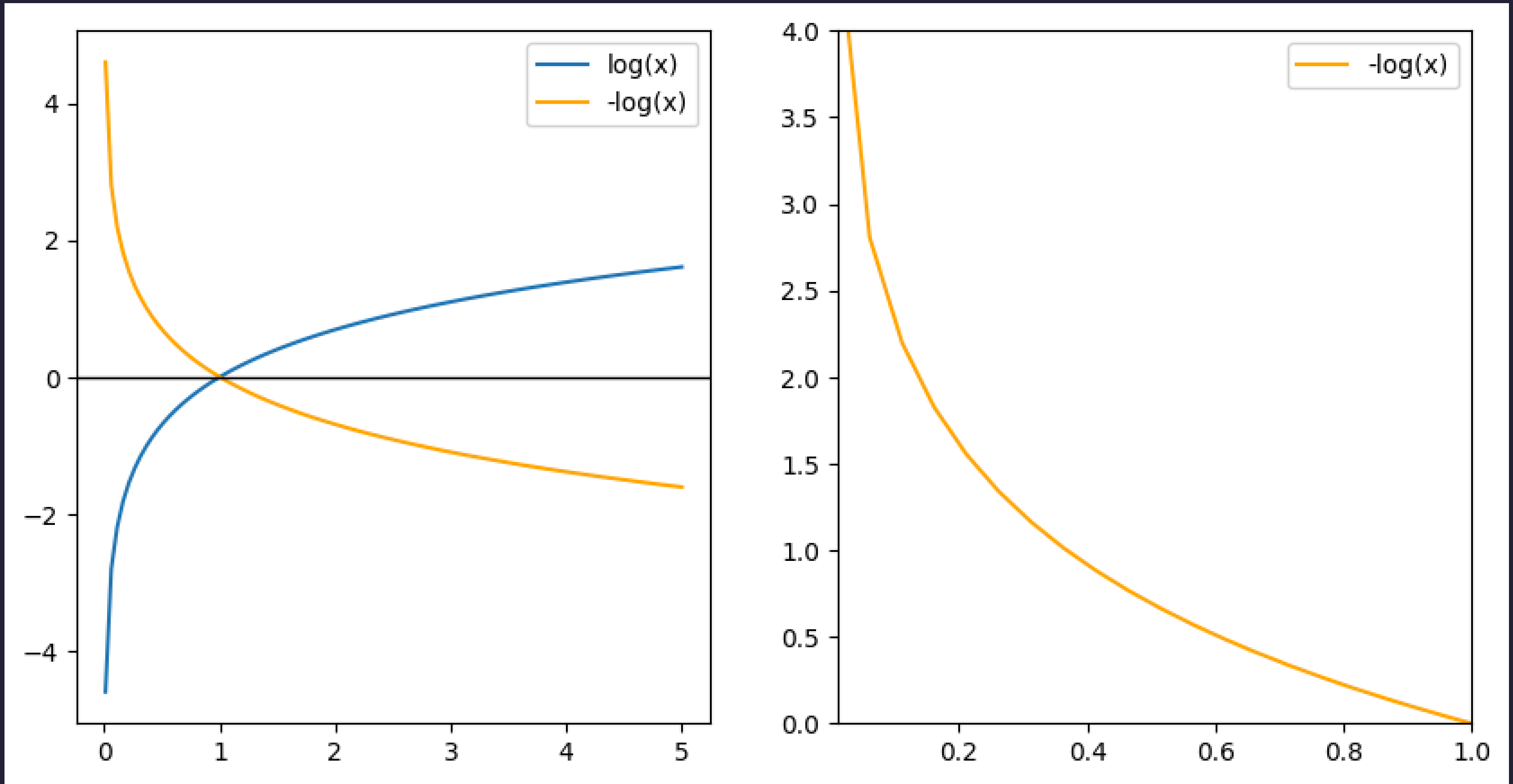
- $L(0, f(x))$ should be small when $f(x)$ is close to 0
- $L(0, f(x))$ should be large when $f(x)$ is close to 1

> The L2 loss for logistic regression is non-convex with many local minima.

Log(istic) loss function

$$L(y, f(x)) = \begin{cases} -\log(f(x)) & \text{if } y = 1 \\ -\log(1 - f(x)) & \text{if } y = 0 \end{cases}$$

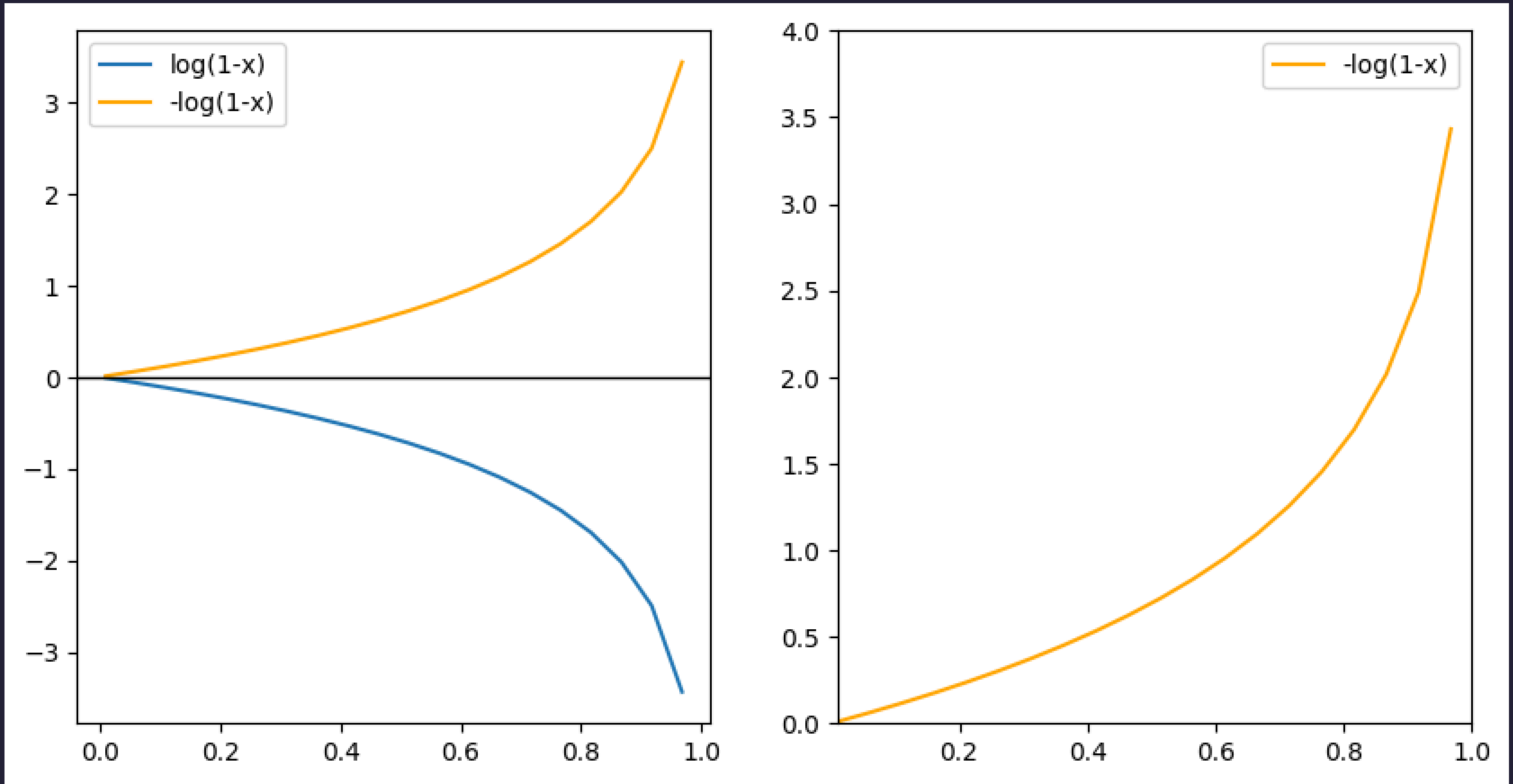
$$\text{if } y = 1, L(1, f(x)) = -\log(f(x))$$



Log(istic) loss function

$$L(y, f(x)) = \begin{cases} -\log(f(x)) & \text{if } y = 1 \\ -\log(1 - f(x)) & \text{if } y = 0 \end{cases}$$

$$\text{if } y = 0, L(1, f(x)) = -\log(f(x))$$



Combining the two cases

$$L(y, f(x)) = \begin{cases} -\log(f(x)) & \text{if } y = 1 \\ -\log(1 - f(x)) & \text{if } y = 0 \end{cases}$$

$$L(y, f(x)) = -y \log(f(x)) - (1 - y) \log(1 - f(x))$$

$$\text{if } y = 1, L(1, f(x)) = -1 * \log(f(x)) - (1 - 1) \log(1 - f(x)) = -\log(f(x))$$
$$\text{if } y = 0, L(1, f(x)) = -0 * \log(f(x)) - (1 - 0) \log(1 - f(x)) = -\log(1 - f(x))$$

Cost function

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(y^{(i)}, f(x^{(i)})) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(f(x^{(i)})) + (1-y^{(i)})\log(1-f(x^{(i)}))]$$

Gradient descent for logistic regression

$$w_j = w_j - \alpha \frac{\partial J(\vec{w},b)}{\partial w_j}, j = 1, 2, \ldots, k$$

$$w_1 = w_1 - \alpha \frac{\partial J(\vec{w},b)}{\partial w_1}$$

$$w_2 = w_2 - \alpha \frac{\partial J(\vec{w},b)}{\partial w_2}$$

…

$$w_k = w_k - \alpha \frac{\partial J(\vec{w},b)}{\partial w_k}$$

$$b = b - \alpha \frac{\partial J(\vec{w},b)}{\partial b}$$

Gradient for logistic regression

$f(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$

$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^{m} [y^{(i)} \log(f(\vec{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f(\vec{x}^{(i)}))]$

Partial derivative of the cost function with respect to w_j

$$\frac{\partial J(\vec{w}, b)}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (f(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

Partial derivative of the cost function with respect to b

$$\frac{\partial J(\vec{w}, b)}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (f(\vec{x}^{(i)}) - y^{(i)})$$

💻 Gradient descent for logistic regression

Evaluation metrics for `single` threshold

Confusion matrix

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

- True positive (TP): correctly predicted positive
- True negative (TN): correctly predicted negative
- False positive (FP): incorrectly predicted positive
- False negative (FN): incorrectly predicted negative

As thresold ⬆️, positive predictions (TP, FP) ⬇️ and negative predictions (TN, FN) ⬆️

Visualizing the confusion matrix

https://developers-dot-devsite-v2-prod.appspot.com/machine-learning/crash-course/classification/thresholding_cd2cec3b3711b6befffa498911d9a6be0fa233b9b7238880d23cdb7593116511.frame

Which mistake is more costly?

Spam detection:

- FP: non-spam email is classified as spam
- FN: spam email is classified as non-spam

Cancer detection:

- FP: non-cancerous tumor is classified as cancerous
- FN: cancerous tumor is classified as non-cancerous

Credit card fraud detection:

- FP: non-fraudulent transaction is classified as fraudulent
- FN: fraudulent transaction is classified as non-fraudulent

Accuracy

$$\frac{\text{correct predictions}}{\text{total predictions}} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Use when the classes are balanced

Avoid for imbalanced datasets

- 99% of the data is negative, and 1% is positive. A model that predicts all negative will have 99% accuracy.

Recall (True positive rate)

$$\frac{\text{correctly predicted positive}}{\text{actual positive}} = \frac{TP}{(TP + FN)}$$

Use when false negatives (FN) are more expensive than false positives (FP).

- spam email is classified as non-spam
- cancerous tumor is classified as non-cancerous
- fraudulent transaction is classified as non-fraudulent

False positive rate

$$\frac{\text{incorrectly predicted negative}}{\text{actual negative}} = \frac{FP}{(FP + TN)}$$

Use when false positives (FP) are more expensive than false negatives (FN).

- non-spam email is classified as spam
- non-cancerous tumor is classified as cancerous
- non-fraudulent transaction is classified as fraudulent

Precision

$$\frac{\text{correctly predicted positive}}{\text{predicted positive}} = \frac{TP}{(TP + FP)}$$

Use when it's very important for positive predictions to be accurate.

- spam email is classified as spam
- cancerous tumor is classified as cancerous
- fraudulent transaction is classified as fraudulent

F1 score

$$2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Use when you want a single metric that balances precision and recall.

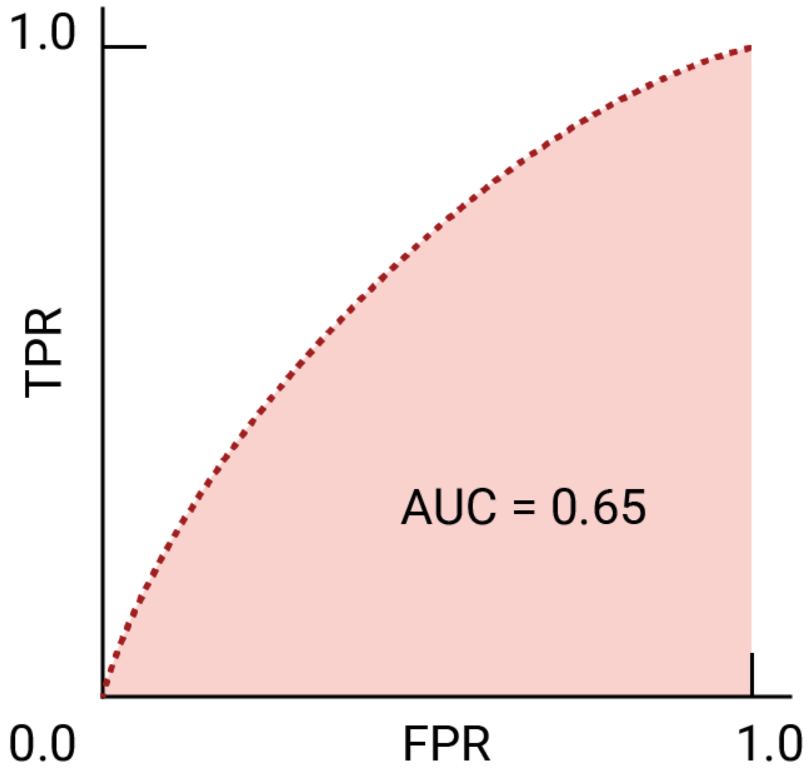
Evaluation metrics for `all` possible thresholds

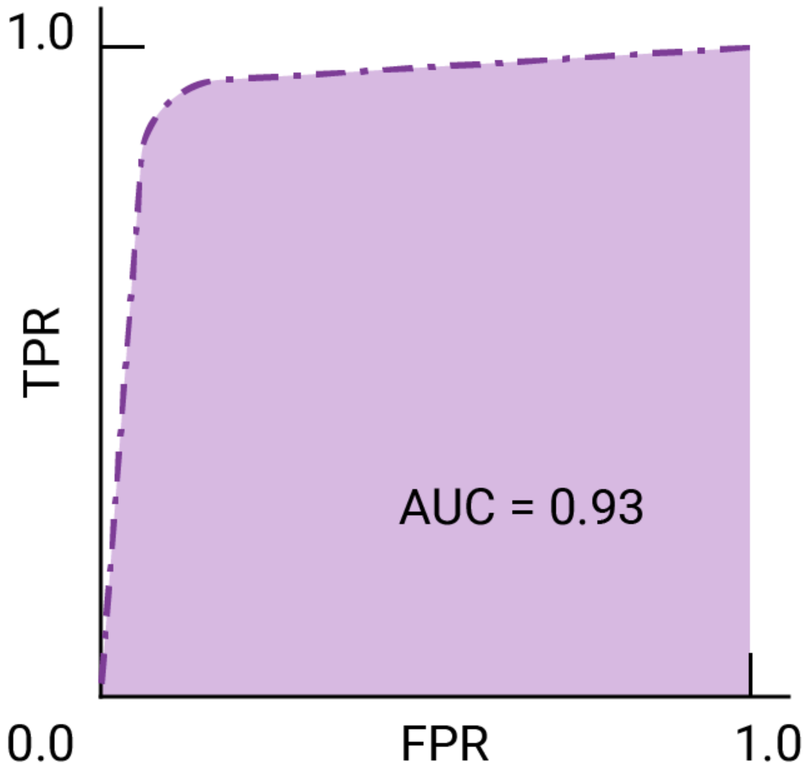
ROC: Receiver-operating characteristic curve

- False positive rate (FPR) vs. True positive rate (TPR) across all thresholds

AUC: Area under the ROC curve

- Probability that the model will rank the actual positive higher than the actual negative.
- e.g., a spam classifier with AUC of 1.0 always assigns a random spam email a higher probability of being spam than a random legitimate email.


1.0
TPR
AUC = 0.65
0.0
FPR
1.0

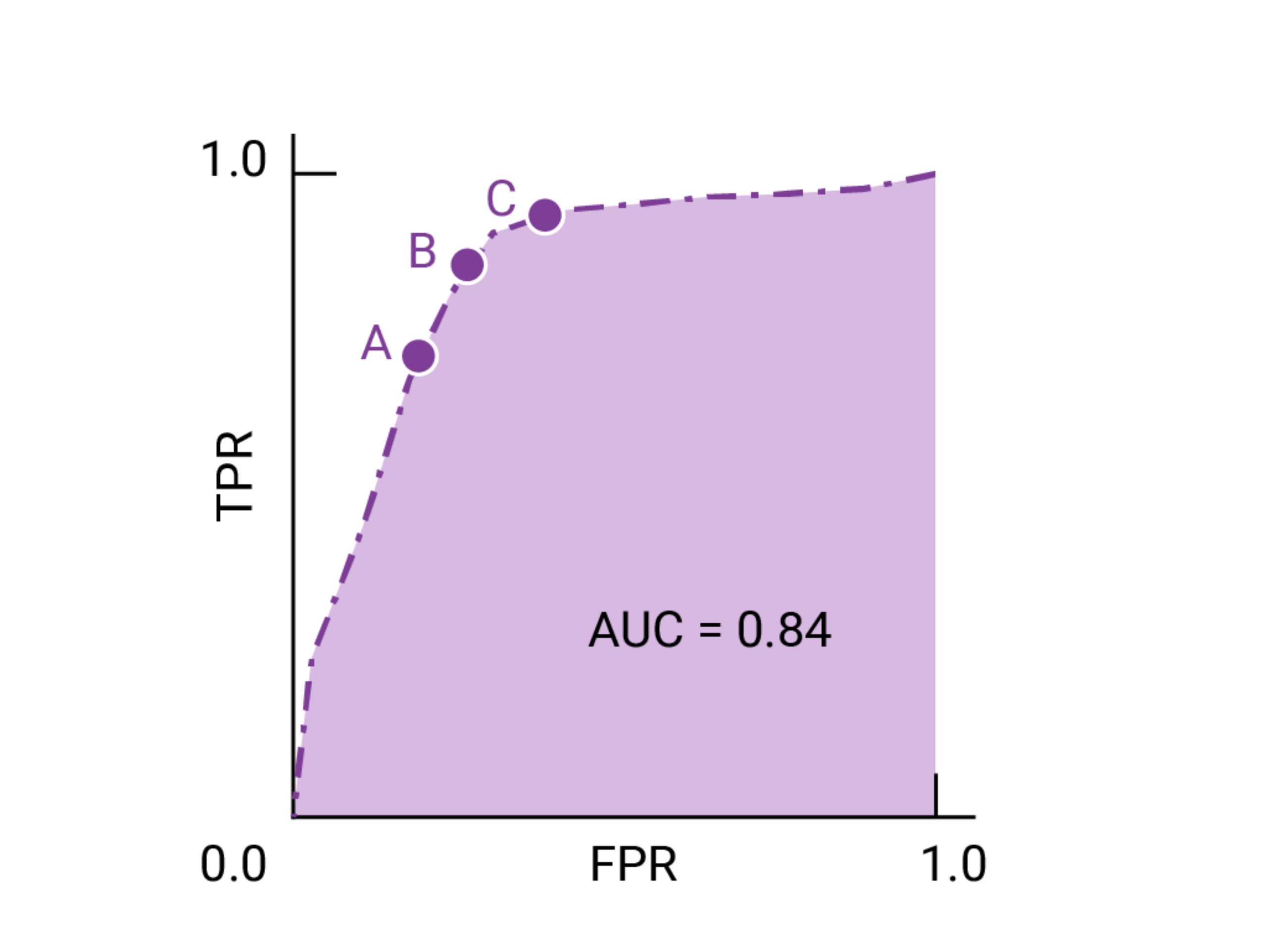

1.0
TPR
AUC = 0.93
0.0
FPR
1.0

Which threshold to choose?

`B` : highest TPR for a given FPR (closest to the top-left corner)

`A` : lowest FPR (when FP is costly)

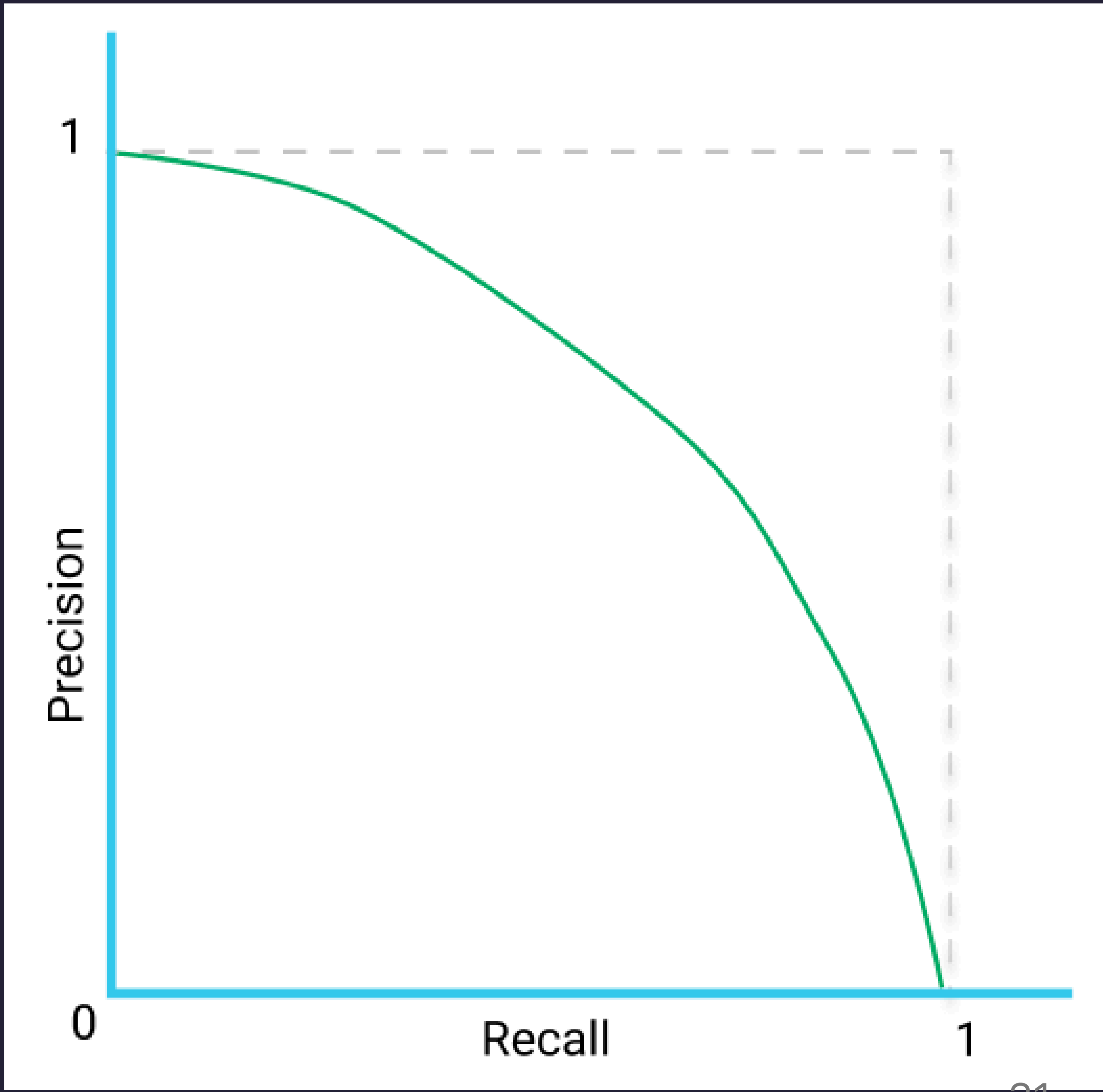
`C` : highest TPR (when FN is costly)



Precision-recall curve (PRC)

For imbalanced datasets, use PRC instead of ROC.

Precision vs. Recall across all thresholds.



Visualizing evaluation metrics

https://developers-dot-devsite-v2-prod.appspot.com/machine-learning/crash-course/classification/roc-and-auc_3689cac9917eb19cc4a8c29c3140b8e30ffacdd8fcfc99df2ec5a1879dbef187.frame