Binary Classification

f() ⟶ {Cat, Dog}

$$y = f(\mathbf{x}), \quad \text{Input: } \mathbf{x} \in \mathbb{R}^n \longrightarrow \text{Output: } y \in \{0, 1\}$$

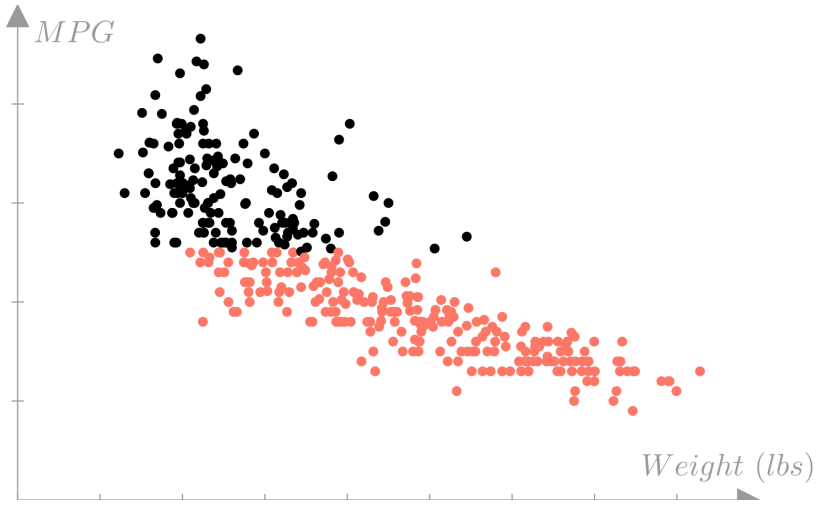
Linear Reg.

$\mathbb{R}$

Binary Classification

$$y = f(\mathbf{x}), \quad \text{Input: } \mathbf{x} \in \mathbb{R}^n \longrightarrow \text{Output: } y \in \{0, 1\}$$

$$\text{Input: } \mathbf{x}_i = \begin{bmatrix} \text{Weight} \\ \text{Horsepower} \\ \text{Displacement} \\ \text{0-60mph} \end{bmatrix}, \quad \text{Output: } y_i = \begin{cases} 1 : \text{Meets target } (MPG \geq 30) \\ 0 : \text{Fails to meet target } (MPG < 30) \end{cases}$$



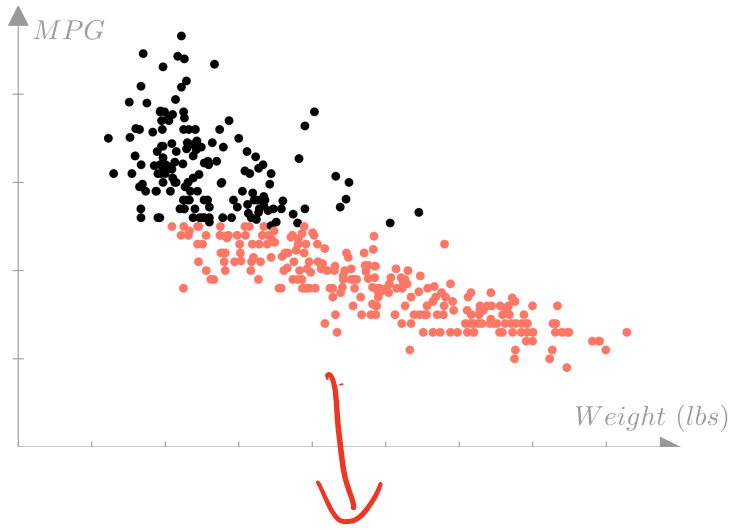
$$\text{Honda Accord: } \begin{bmatrix} \text{Weight:} & \text{2500 lbs} \\ \text{Horsepower:} & \text{123 HP} \\ \text{Displacement:} & \text{2.4 L} \\ \text{0-60mph:} & \text{7.8 Sec} \end{bmatrix} \longrightarrow 1 \text{ (Meets target)}$$

$$\text{Dodge Aspen: } \begin{bmatrix} \text{Weight:} & \text{3800 lbs} \\ \text{Horsepower:} & \text{155 HP} \\ \text{Displacement:} & \text{3.2 L} \\ \text{0-60mph:} & \text{6.8 Sec} \end{bmatrix} \longrightarrow 0 \text{ (Does not meet target)}$$

Binary Classification

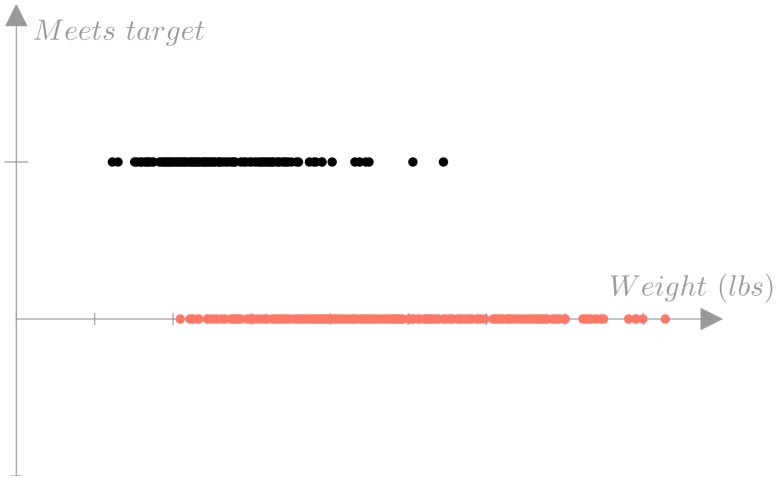
$$y = f(\mathbf{x}), \quad \text{Input: } \mathbf{x} \in \mathbb{R}^n \longrightarrow \text{Output: } y \in \{0, 1\}$$

$$\text{Input: } \mathbf{x}_i = \begin{bmatrix} \text{Weight} \\ \text{Horsepower} \\ \text{Displacement} \\ \text{0-60mph} \end{bmatrix}, \quad \text{Output: } y_i = \begin{cases} 1 : \text{Meets target } (MPG \geq 30) \\ 0 : \text{Fails to meet target } (MPG < 30) \end{cases}$$



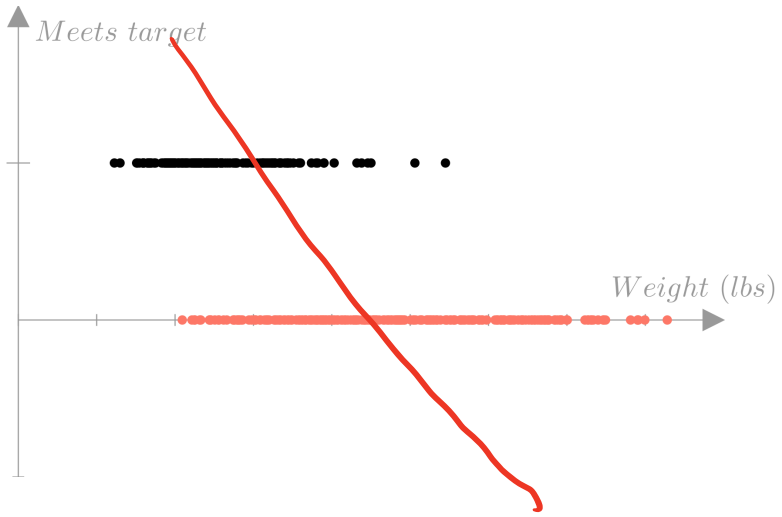
$$\text{Honda Accord: } \begin{bmatrix} \text{Weight:} & \text{2500 lbs} \\ \text{Horsepower:} & \text{123 HP} \\ \text{Displacement:} & \text{2.4 L} \\ \text{0-60mph:} & \text{7.8 Sec} \end{bmatrix} \longrightarrow 1 \text{ (Meets target)}$$

$$\text{Dodge Aspen: } \begin{bmatrix} \text{Weight:} & \text{3800 lbs} \\ \text{Horsepower:} & \text{155 HP} \\ \text{Displacement:} & \text{3.2 L} \\ \text{0-60mph:} & \text{6.8 Sec} \end{bmatrix} \longrightarrow 0 \text{ (Does not meet target)}$$



Can we do binary classification with linear regression?

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b = \sum_{i=1}^{d} x_i w_i + b$$



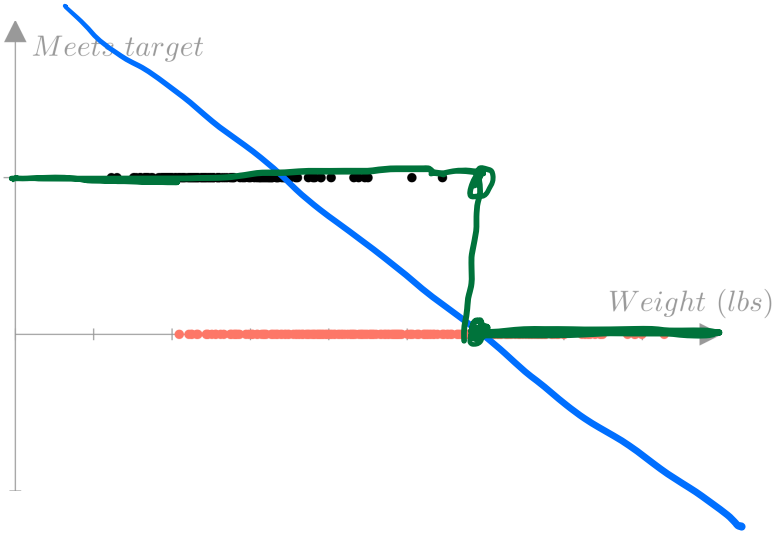
Binary predictions with thresholding

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ b \end{bmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

$$x^T w \equiv \sum_{i=1}^{d} x_i w_i + b$$

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} \longrightarrow f(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x}^T\mathbf{w} \geq 0 \\ 0 \text{ if } \mathbf{x}^T\mathbf{w} < 0 \end{cases}$$

Linear fcs.



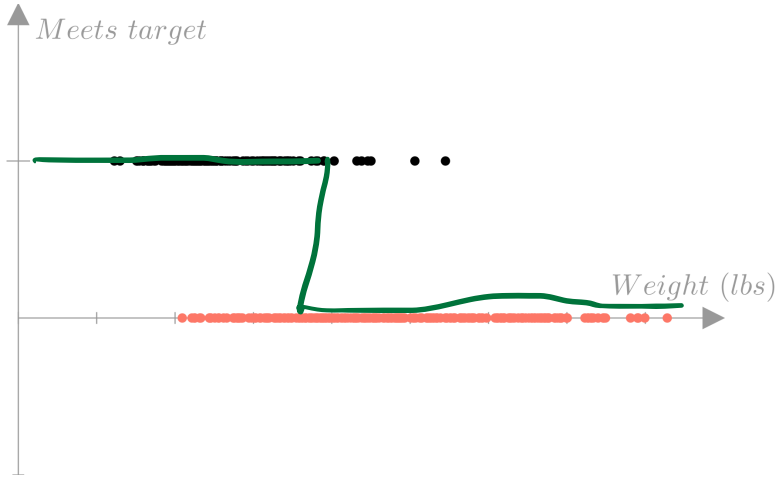
$$f(\mathbf{x}) = \mathbb{I}(\mathbf{x}^T\mathbf{w} \geq 0)$$

Indicator func.

Binary predictions with thresholding

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} \quad \longrightarrow \quad f(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x}^T\mathbf{w} \geq 0 \\ 0 \text{ if } \mathbf{x}^T\mathbf{w} < 0 \end{cases}$$

$$f(\mathbf{x}) = \mathbb{I}(\mathbf{x}^T\mathbf{w} \geq 0)$$



$$\text{Meets target} = f(\mathbf{x}) =$$

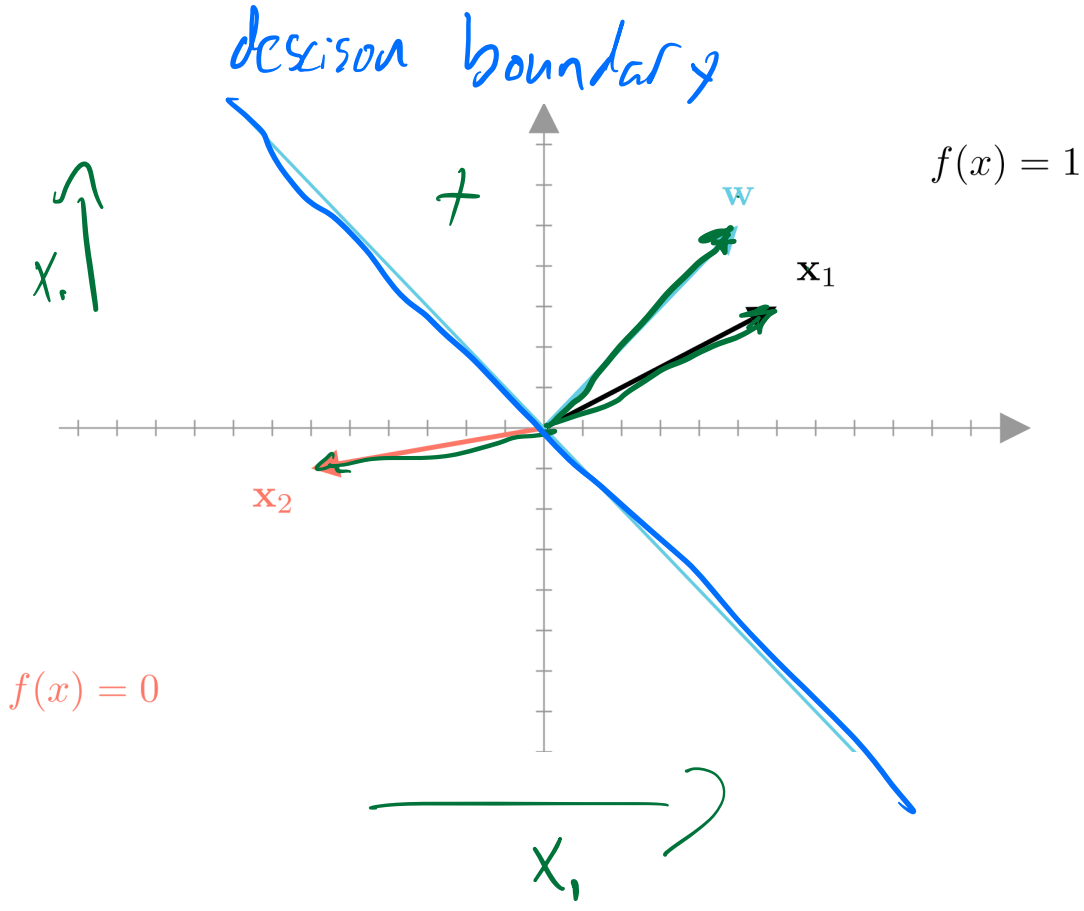
$$\big((\text{weight})w_1 + (\text{horsepower})w_2 + (\text{displacement})w_3 + (\text{0-60mph})w_4 + b\big) \geq 0$$

$$f(\mathbf{x}) = \left(\begin{bmatrix} \text{Weight} \\ \text{Horsepower} \\ \text{Displacement} \\ \text{0-60mph} \\ 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ b \end{bmatrix} \geq 0 \right)$$

Thresholds in higher dimensions

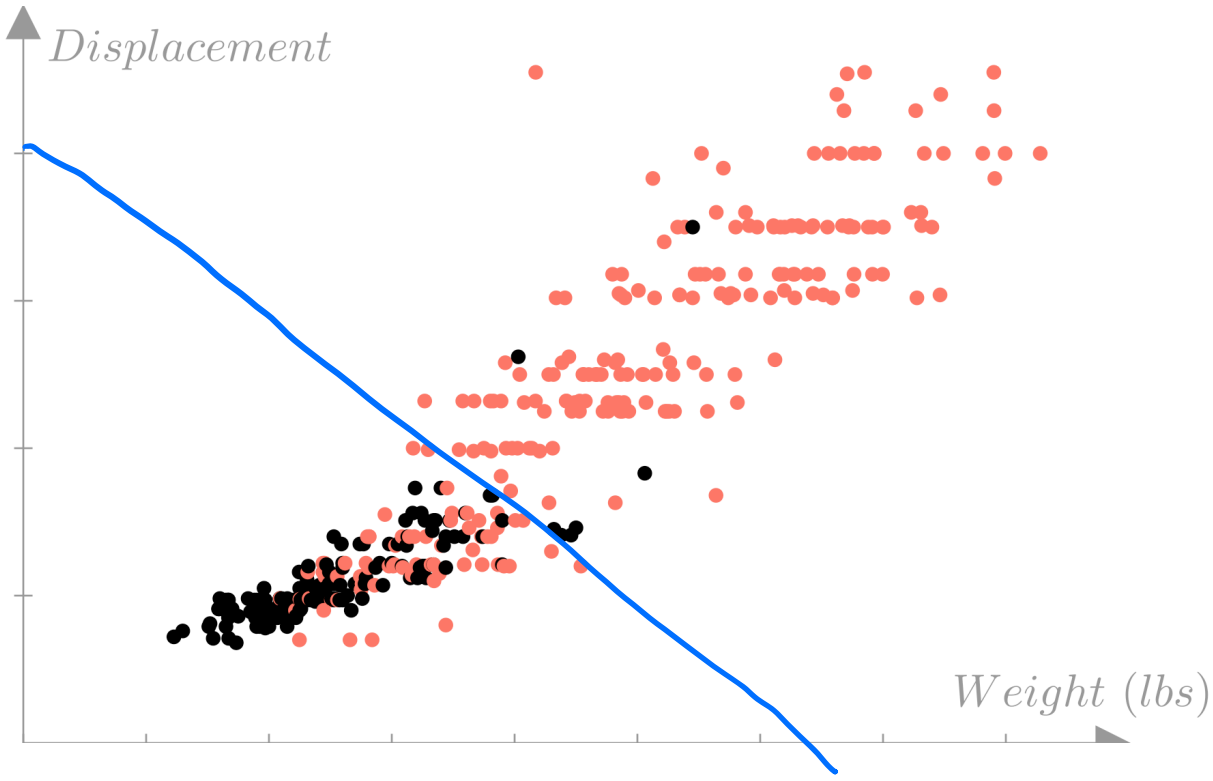
$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} \quad \longrightarrow \quad f(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x}^T\mathbf{w} \geq 0 \\ 0 \text{ if } \mathbf{x}^T\mathbf{w} < 0 \end{cases}$$

$$\mathbf{x}^T\mathbf{w} = ||\mathbf{x}||_2||\mathbf{w}||_2 \cos\theta$$



Decision boundaries

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} \quad \longrightarrow \quad f(\mathbf{x}) = \begin{cases} 1 \text{ if } \mathbf{x}^T\mathbf{w} \geq 0 \\ 0 \text{ if } \mathbf{x}^T\mathbf{w} < 0 \end{cases}$$



Accuracy as a loss

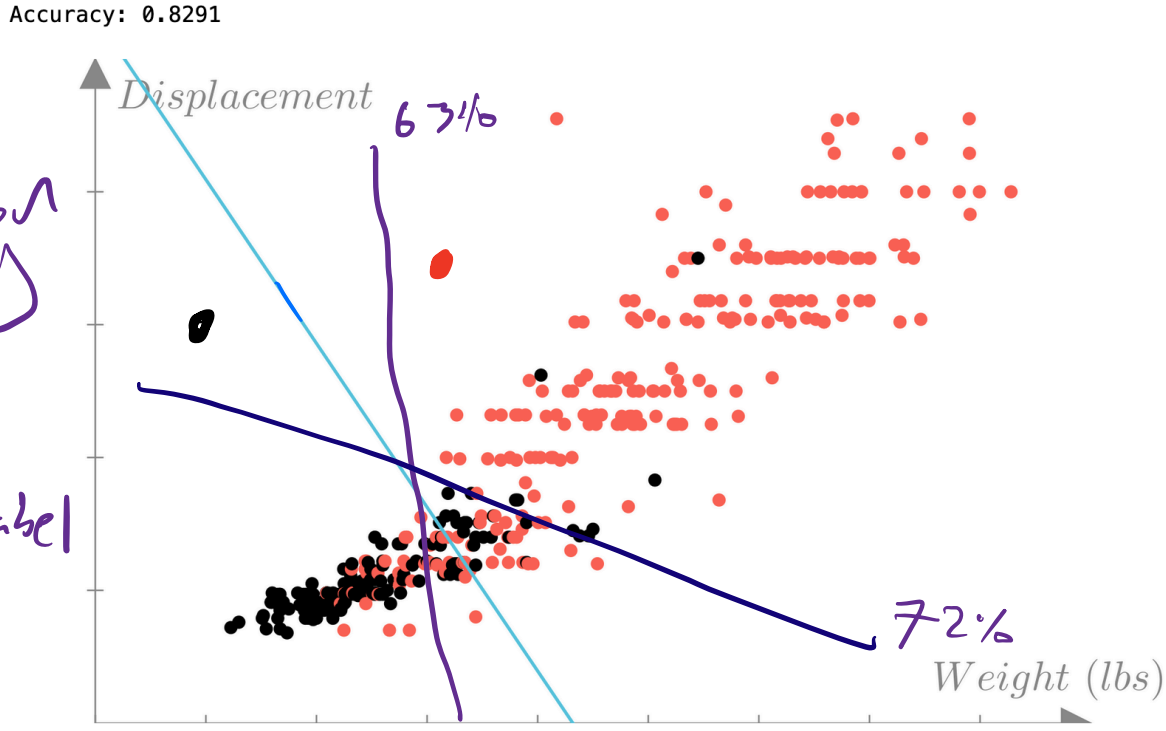
Accuracy: $\dfrac{\text{\# of correct predictions}}{\text{Total predictions}}$

Prediction function $\mathbb{I}(x^T w \geq 0)$

$$\textbf{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\big(f(\mathbf{x}_i) = y_i\big)$$

label

Dataset



Can we use accuracy to find optimal parameters?

Gradient descent

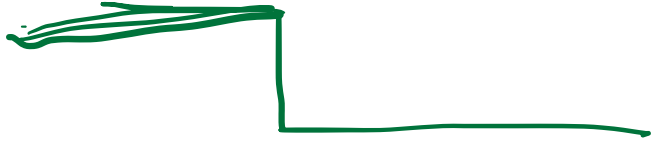
While $\nabla f(\mathbf{w}^{(i)}) \neq \mathbf{0}:\quad \mathbf{w}^{(i+1)} \leftarrow \mathbf{w}^{(i)} - \nabla f(\mathbf{w}^{(i)})$

Loss

$$\textbf{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\big(f(\mathbf{x}_i) = y_i\big)$$

$$\frac{d}{dw}\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\big(f(x_i)=y_i\big)$$

$$\frac{d}{dw}\mathbb{I}\big(f_w(x_i)=y_i\big)$$



The Bernoulli distribution

Probability of **heads:** q, Probability of **tails:** $1 - q$

$$p(y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases} \quad q \in [0, 1], \; y \in \{0, 1\}$$

The Bernoulli distribution

Probability of **heads:** q, Probability of **tails:** $1 - q$

$$p(y) = \begin{cases} q & \text{if } y = 1 \\ 1 - q & \text{if } y = 0 \end{cases} \quad q \in [0, 1],\ y \in \{0, 1\}$$

$y=1$

$y=0$

$$p(y) = q^y (1 - q)^{1-y}$$

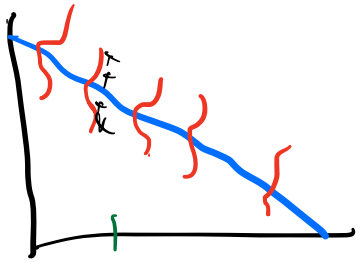
$$\log p(y) = y \log q + (1 - y) \log(1 - q)$$

A probabilistic model for binary classification

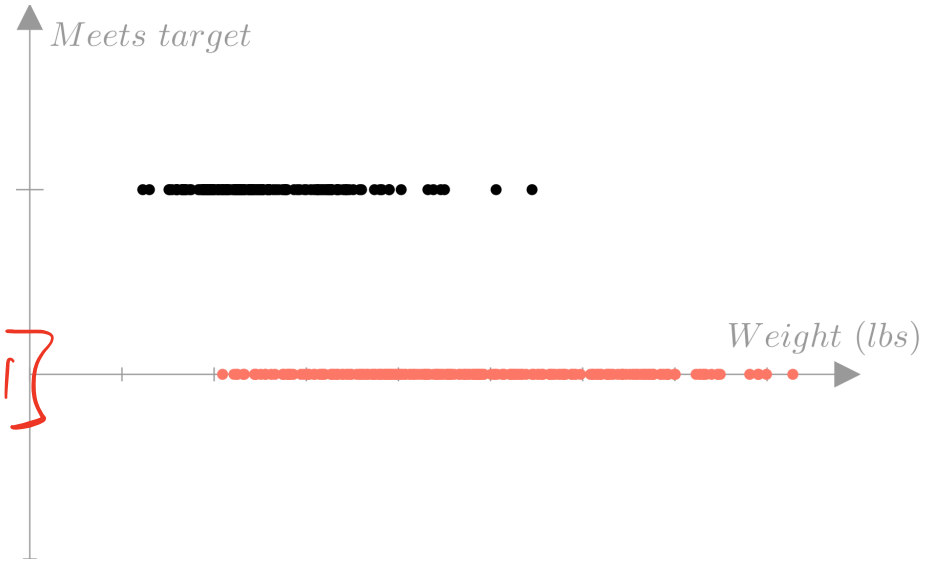
Linear regression: Normal observations

$$y_i \sim \mathcal{N}(\mathbf{x}_i^T\mathbf{w},\ \sigma^2)$$

mean



Bernoulli observations?



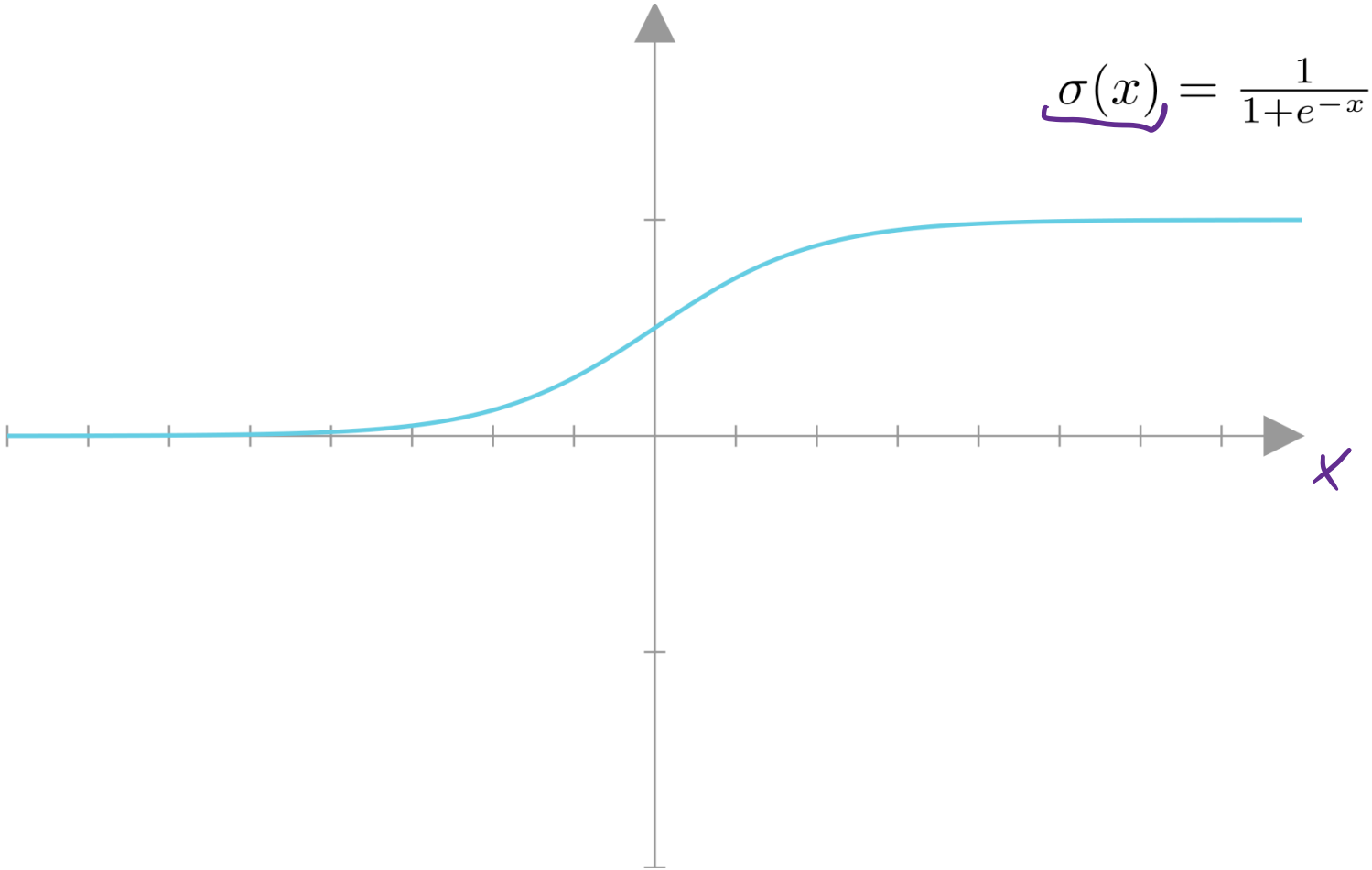
$\mathbf{x}^T\mathbf{w} \notin [0,1] \longrightarrow y_i \sim \textbf{Bernoulli}(\mathbf{q} = ?)$

$x^T w \in [0,1]$

Need $g(x)$: $\mathbb{R} \Longrightarrow [0,1]$

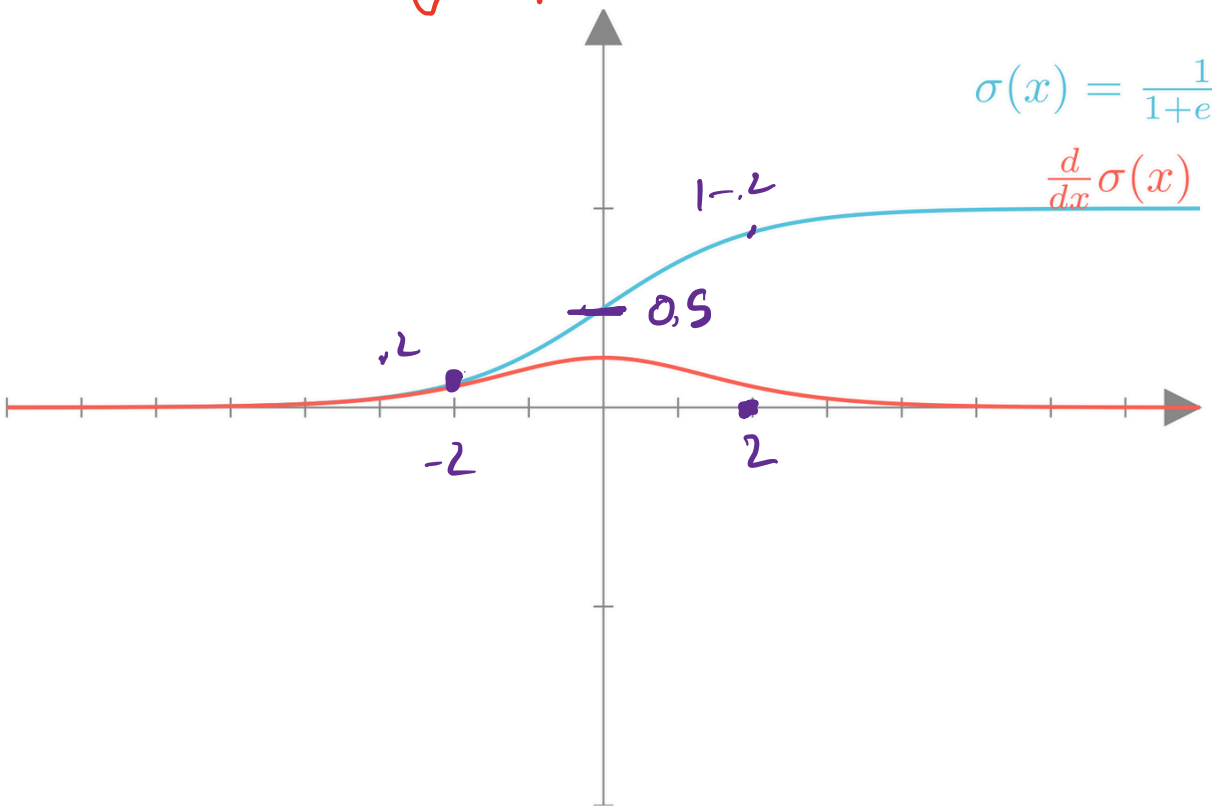
Input: $x \in \mathbb{R} \longrightarrow$ **Output:** $y \in [0,1]$

The sigmoid function



Properties of the sigmoid function

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$



$$\sigma(0) = 0.5$$

$$1 - \sigma(x) = \sigma(-x)$$

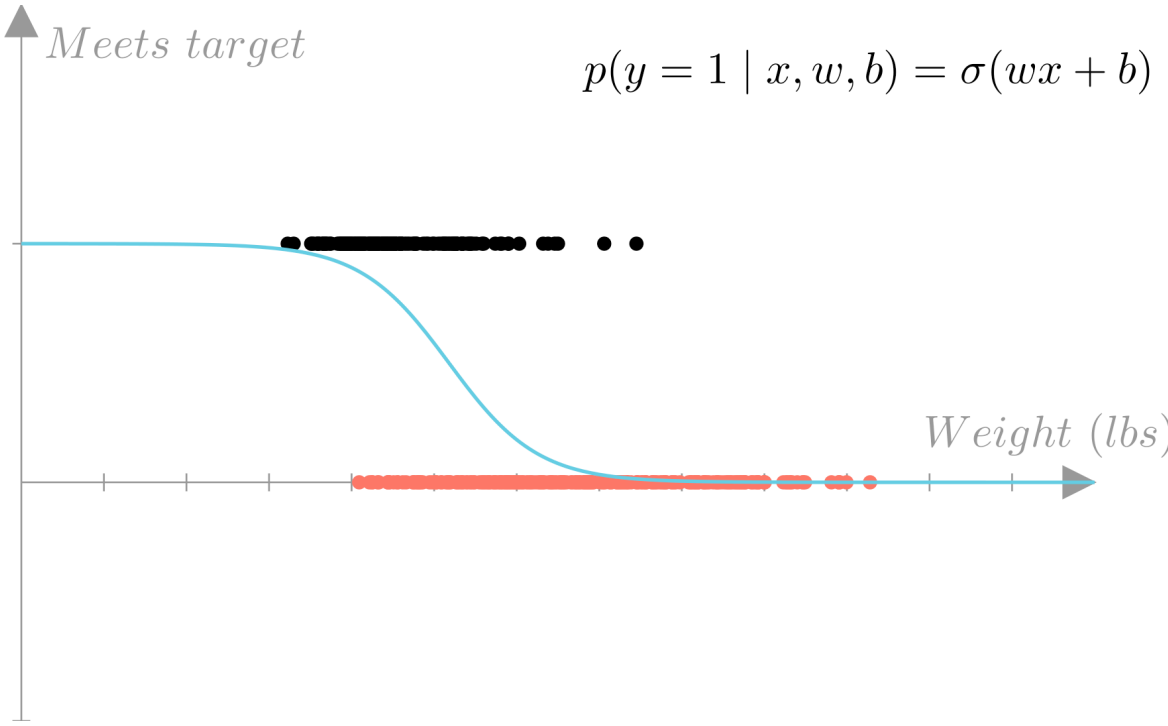
Logistic regression

Parameters

$$y_i \sim \textbf{Bernoulli}(\sigma(\mathbf{x_i^T w}))$$

$$p(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{x}_i^T \mathbf{w}),$$

$$p(y_i = 0 \mid \mathbf{x}_i, \mathbf{w}) = 1 - \sigma(\mathbf{x}_i^T \mathbf{w}) = \sigma(-\mathbf{x}_i^T \mathbf{w})$$



$$p(y = 1 \mid x, w, b) = \sigma(wx + b)$$

$$\textbf{Prediction function: } f(\mathbf{x}) = \begin{cases} 1 \text{ if } p(y = 1 \mid \mathbf{x}, \mathbf{w}) \geq p(y = 0 \mid \mathbf{x}, \mathbf{w}) \\ 0 \text{ otherwise} \end{cases}$$

Logistic regression

Prediction function: $f(\mathbf{x}) = \begin{cases} 1 \text{ if } p(y=1 \mid \mathbf{x}, \mathbf{w}) \geq p(y=0 \mid \mathbf{x}, \mathbf{w}) \\ 0 \text{ otherwise} \end{cases}$

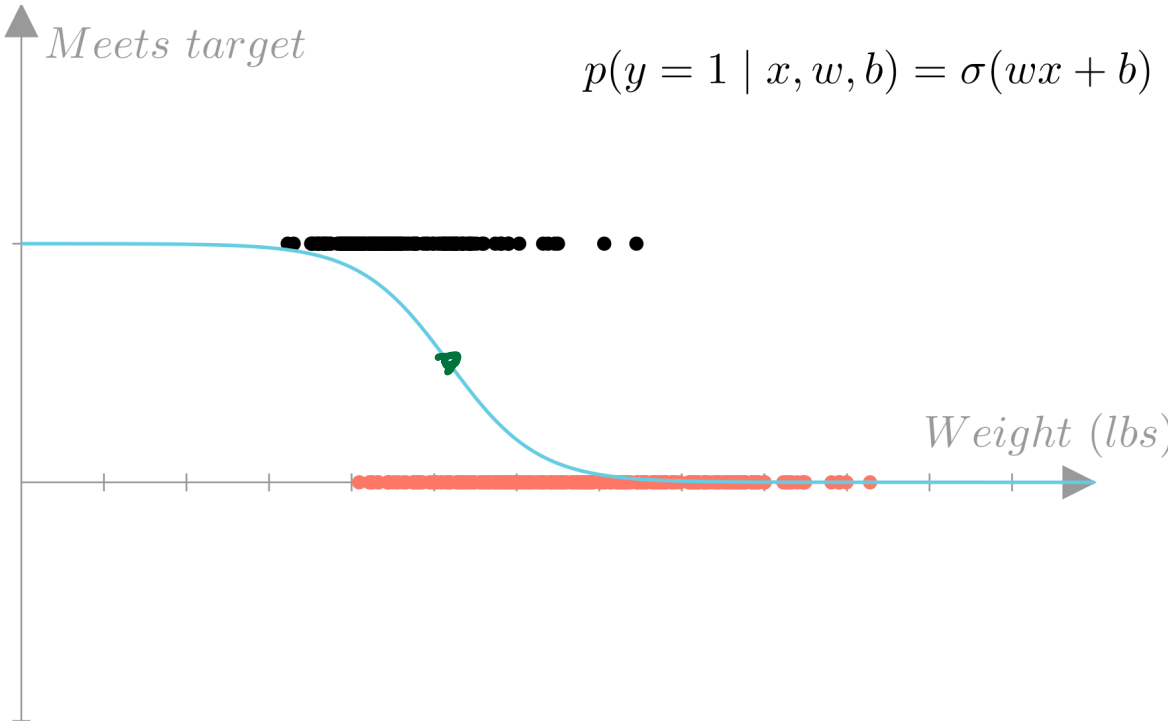
$P(y=1|x,w) > P(y=0|x,w)$

$\equiv P(y=1|x,w) > 0.5$

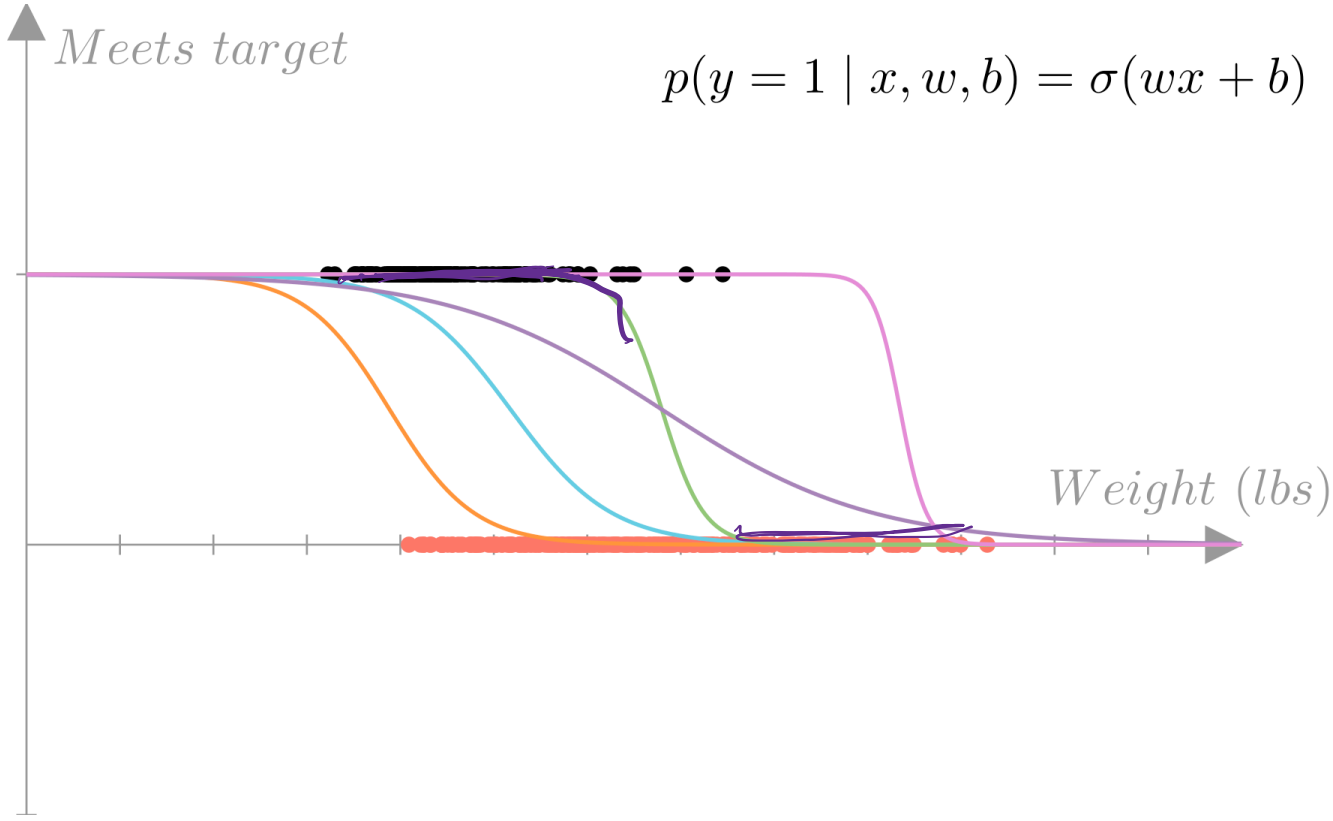
$$p(y=1 \mid \mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^T\mathbf{w}) \geq 0.5$$

$\sigma(0) = 0.5$

$$p(y_i = 1) \geq 0.5 \quad \longrightarrow \quad \mathbf{x}^T\mathbf{w} \geq 0$$



Maximum likelihood for logistic regression



$$p(y = 1 \mid x, w, b) = \sigma(wx + b)$$

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(y_1, \ldots, y_N \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{w})$$

Maximum likelihood for logistic regression

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{y} \mid \mathbf{X}, \mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmax}}\, p(y_1, \ldots, y_N \mid \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{w})$$

$$\equiv \prod_{i=1} P(y_i \mid x_i, w)$$

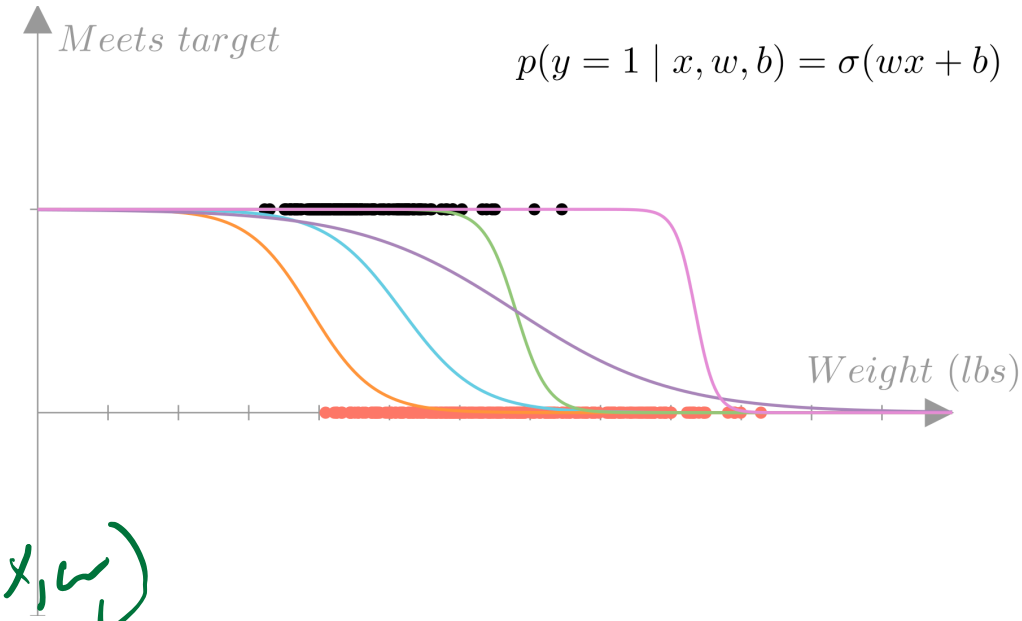
$$\mathbf{Loss}(\mathbf{w}) = \mathbf{NLL}(\mathbf{w}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \mathbf{w})$$

$$p(y_i = 1 \mid \mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{x}_i^T \mathbf{w}), \quad p(y_i = 0 \mid \mathbf{x}_i, \mathbf{w}) = 1 - \sigma(\mathbf{x}_i^T \mathbf{w}) = \sigma(-\mathbf{x}_i^T \mathbf{w})$$

sum over our dataset

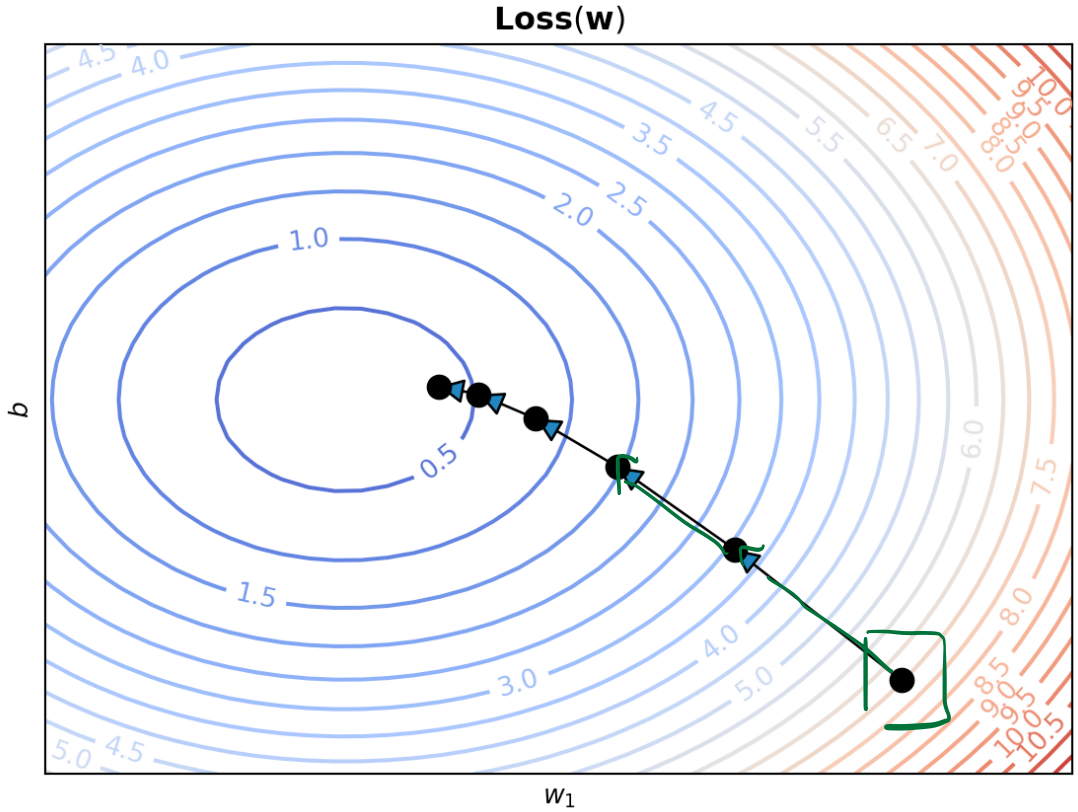
$$\mathbf{NLL}(\mathbf{w}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \Big[y_i \log \sigma(\mathbf{x}_i^T \mathbf{w}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}_i^T \mathbf{w})) \Big]$$

$P(y=1 \mid x, w)$ $\quad P(y=0 \mid x, w)$



Gradient descent

While $\nabla f(\mathbf{w}^{(i)}) \neq \mathbf{0}$: $\mathbf{w}^{(i+1)} \leftarrow \mathbf{w}^{(i)} - \nabla f(\mathbf{w}^{(i)})$



Deriving the gradient

$$\mathbf{NLL}(\mathbf{w}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \left[y_i \log \sigma(\mathbf{x}_i^T \mathbf{w}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}_i^T \mathbf{w})) \right]$$

$$\frac{d}{dw} - \sum_{i=1}^{N} \left[y_i \log \sigma(x_i^T w) + (1 - y_i) \log(1 - \sigma(x_i^T w)) \right]$$

$$= -\sum_{i=1}^{N} \frac{d}{dw} \left[y_i \log \sigma(x_i^T w) + (1 - y_i) \log(1 - \sigma(x_i^T w)) \right]$$

$$= -\sum_{i=1}^{N} \left[y_i \frac{d}{dw} \log \sigma(x_i^T w) + (1 - y_i) \frac{d}{dw} \log(1 - \sigma(x^T w)) \right]$$

$$= \frac{1}{\sigma(x_i^T w)} \frac{d}{dw} \sigma(x^T w)$$

$$\frac{d}{dx} \sigma(x) = \sigma(x)(1 - \sigma(x))$$

$$= \frac{1}{\sigma(x_i^T w)} \left(\sigma(x^T w)(1 - \sigma(x^T w)) \right) \frac{d}{dw} x^T w$$

$$= \left(1 - \sigma(x^T w)\right) x^T$$

Deriving the gradient

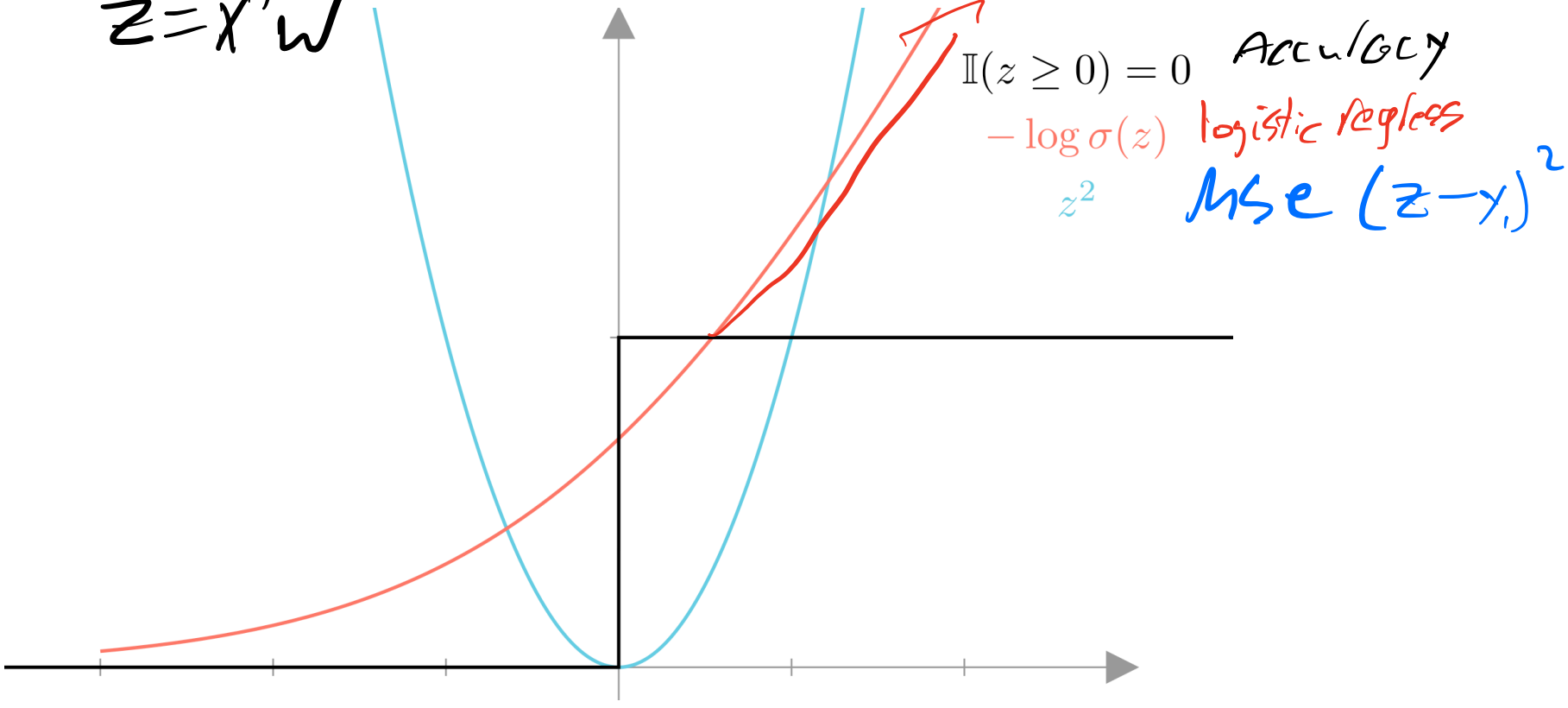
$$\mathbf{NLL}(\mathbf{w}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \left[y_i \log \sigma(\mathbf{x}_i^T \mathbf{w}) + (1 - y_i) \log(1 - \sigma(\mathbf{x}_i^T \mathbf{w})) \right]$$

$$\nabla_{\mathbf{w}} \mathbf{NLL}(\mathbf{w}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \Big(1 - \sigma\big((2y_i - 1)\mathbf{x}_i^T \mathbf{w}\big) \Big) \Big((2y_i - 1)\mathbf{x}_i \Big)$$

Comparing loss functions

$y_i = 1$

$z = x^T w$



accuracy

logistic regress

$mse\ (z - y_i)^2$

Multi-class classification

f() → {Cat, Dog, Mouse,…}

$$y = f(\mathbf{x}), \quad \text{Input: } \mathbf{x} \in \mathbb{R}^n \longrightarrow \text{Output: } y \in \{1, 2, \ldots, C\}$$

Ordering irrelevant!

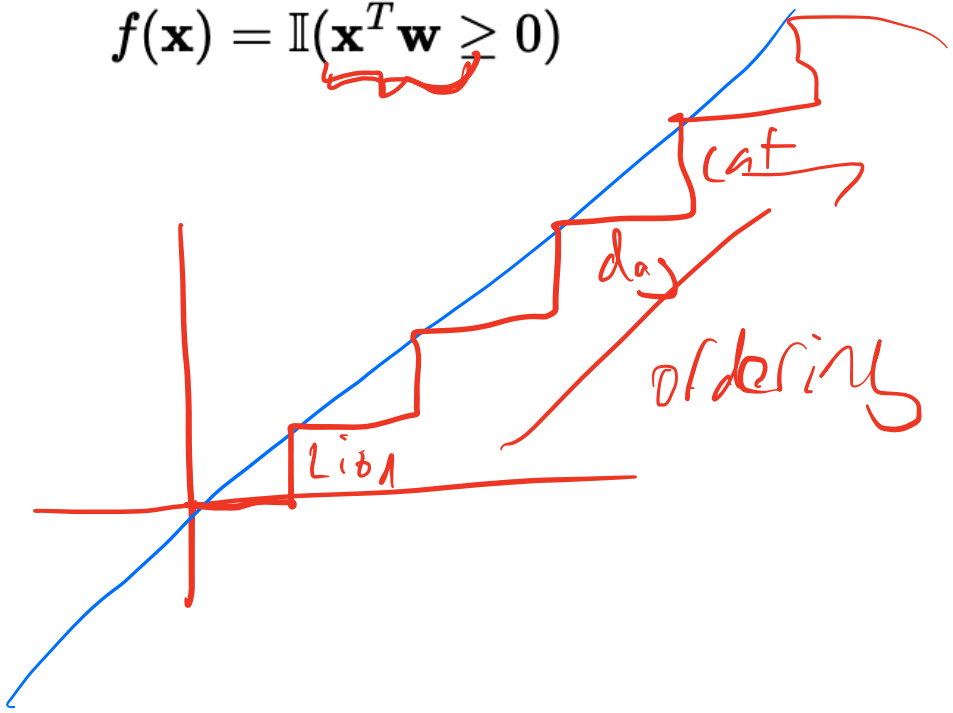
1: Cat, 2: Dog, 3: Mouse

1: Dog, 2: Mouse, 3: Cat

Multi-class prediction functions

Binary thresholding

$$f(\mathbf{x}) = \mathbb{I}(\mathbf{x}^T\mathbf{w} \geq 0)$$



Multi-class thresholding

$$f(\mathbf{x}) = \underset{c \in \{1 \ldots C\}}{\operatorname{argmax}}\, \mathbf{x}^T\mathbf{w}_c$$

C different function

Multi-class prediction functions

Multi-class thresholding

$$f(\mathbf{x}) = \underset{c \in \{1 \ldots C\}}{\operatorname{argmax}} \mathbf{x}^T \mathbf{w}_c$$

For convenience, we can also define a matrix that contains all C parameter vectors:

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_C^T \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & \ldots & W_{1d} \\ W_{21} & W_{22} & \ldots & W_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ W_{C1} & W_{C2} & \ldots & W_{Cd} \end{bmatrix}$$

With this notation, our prediction function becomes:

$$f(\mathbf{x}) = \underset{c \in \{1 \ldots C\}}{\operatorname{argmax}} (\mathbf{x}^T \mathbf{W}^T)_c, \quad \mathbf{W} \in \mathbb{R}^{C \times d}$$

Multi-class decision boundaries

$$f(\mathbf{x}) = \underset{c \in \{0,1\}}{\operatorname{argmax}} \, (\mathbf{x}^T \mathbf{W}^T)_c = \mathbb{I}(\mathbf{x}^T \mathbf{w}_1 - \mathbf{x}^T \mathbf{w}_0 \geq 0)$$

Categorical distribution

$$p(y = c) = q_c, \quad y \in \{1 \dots C\}$$

$$\mathbf{q} \in \mathbb{R}^C \quad q_c \geq 0 \; \forall c \in \{1 \dots C\} \quad \sum_{c=1}^{C} q_c = 1$$

A probabilistic model for multi-class classification

$$y_i \sim \mathbf{Categorical}(\mathbf{q} = ?), \quad \mathbf{q} = \mathbf{x}_i^T \mathbf{W}^T ?$$

$$\mathbf{x}^T \mathbf{W}^T \in \mathbb{R}^C, \quad q_c \ngeq 0 \; \forall c \in \{1 \dots C\}, \quad \sum_{c=1}^{C} q_c \neq 1$$

$$\textbf{Need} \;\; f(\mathbf{x}) : \; \mathbb{R}^C \longrightarrow [0, \infty)^C, \; \sum_{i=1}^{C} f(\mathbf{x})_c = 1$$

Categorical distribution

$$p(y = c) = q_c, \quad y \in \{1 \dots C\}$$

$$\mathbf{q} \in \mathbb{R}^C \quad q_c \geq 0 \; \forall c \in \{1 \dots C\} \quad \sum_{c=1}^{C} q_c = 1$$

$$p(y) = \prod q_c^{\mathbb{I}(y=c)} = q_y$$

$$\log p(y) =$$

Softmax function

$$\mathbf{softmax}(\mathbf{x})_c = \frac{e^{x_c}}{\sum_{j=1}^{C} e^{x_j}}$$

$$\mathbf{softmax}(\mathbf{x}) = \begin{bmatrix} \frac{e^{x_1}}{\sum_{j=1}^{C} e^{x_j}} \\ \frac{e^{x_2}}{\sum_{j=1}^{C} e^{x_j}} \\ \vdots \\ \frac{e^{x_C}}{\sum_{j=1}^{C} e^{x_j}} \end{bmatrix}$$

Softmax function

$$\mathbf{softmax}(\mathbf{x})_c = \frac{e^{x_c}}{\sum_{j=1}^{C} e^{x_j}}$$

$$\sum_{i=1}^{C} \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}} = \frac{\sum_{i=1}^{C} e^{x_i}}{\sum_{j=1}^{C} e^{x_j}} = 1$$

$$\underset{c \in \{1,\ldots,C\}}{\mathbf{argmax}}\ \mathbf{x}_c = \underset{c \in \{1,\ldots,C\}}{\mathbf{argmax}}\ \mathbf{softmax}(\mathbf{x})_c$$

A probabilistic model for multi-class classification: Multinomial Logistic regression

$$y_i \sim \mathbf{Categorical}\big(\mathrm{softmax}(\mathbf{x}^T\mathbf{W})\big)$$

$$p(y_i = c \mid \mathbf{x}, \mathbf{W}) = \mathrm{softmax}(\mathbf{x}^T\mathbf{W})_c = \frac{e^{\mathbf{x}^T\mathbf{w}_c}}{\sum_{j=1}^{C} e^{\mathbf{x}^T\mathbf{w}_j}}$$

Maximum likelihood estimation for Multinomial Logistic regression

Model

$$y_i \sim \textbf{Categorical}\big(\text{softmax}(\mathbf{x}^T\mathbf{W})\big)$$

$$p(y_i = c \mid \mathbf{x}, \mathbf{W}) = \text{softmax}(\mathbf{x}^T\mathbf{W})_c = \frac{e^{\mathbf{x}^T\mathbf{w}_c}}{\sum_{j=1}^{C} e^{\mathbf{x}^T\mathbf{w}_j}}$$

Loss

$$\textbf{Loss}(\mathbf{W}) = \textbf{NLL}(\mathbf{W}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \mathbf{W})$$

$$\textbf{NLL}(\mathbf{W}, \mathbf{X}, \mathbf{y}) =$$

Maximum likelihood estimation for Multinomial Logistic regression

Model

$$y_i \sim \textbf{Categorical}\big(\text{softmax}(\mathbf{x}^T\mathbf{W})\big)$$

$$p(y_i = c \mid \mathbf{x}, \mathbf{W}) = \text{softmax}(\mathbf{x}^T\mathbf{W})_c = \frac{e^{\mathbf{x}^T\mathbf{w}_c}}{\sum_{j=1}^{C} e^{\mathbf{x}^T\mathbf{w}_j}}$$

Loss

$$\textbf{Loss}(\mathbf{W}) = \textbf{NLL}(\mathbf{W}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \log p(y_i \mid \mathbf{x}_i, \mathbf{W})$$

$$\textbf{NLL}(\mathbf{W}, \mathbf{X}, \mathbf{y}) =$$

$$\mathbf{NLL}(\mathbf{W}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \log\ \text{softmax}(\mathbf{x}_i^T \mathbf{W}^T)_{y_i} = -\sum_{i=1}^{N} \log \frac{e^{\mathbf{x}_i^T \mathbf{w}_{y_i}}}{\sum_{j=1}^{C} e^{\mathbf{x}_i^T \mathbf{w}_j}}$$

$$\mathbf{NLL}(\mathbf{W}, \mathbf{X}, \mathbf{y}) = -\sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w}_{y_i} - \log \sum_{j=1}^{C} e^{\mathbf{x}_i^T \mathbf{w}_j} \right)$$