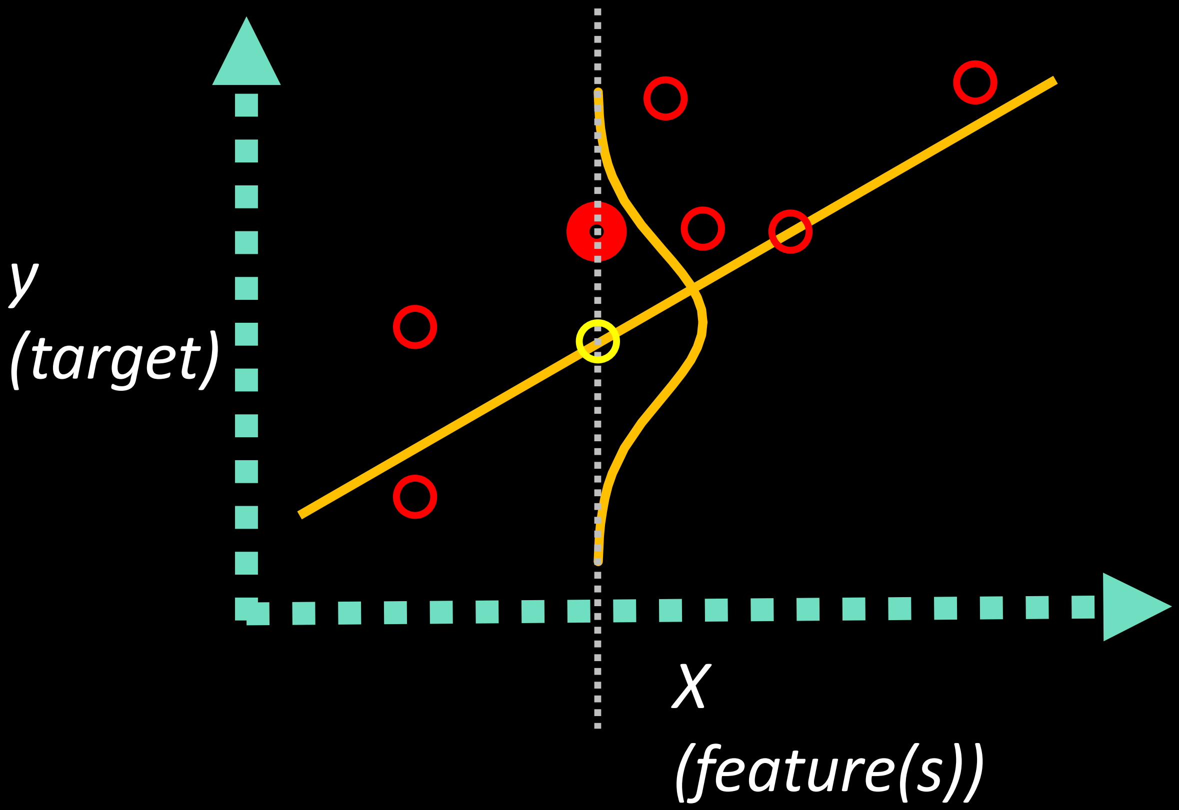
Recap: Linear Regression

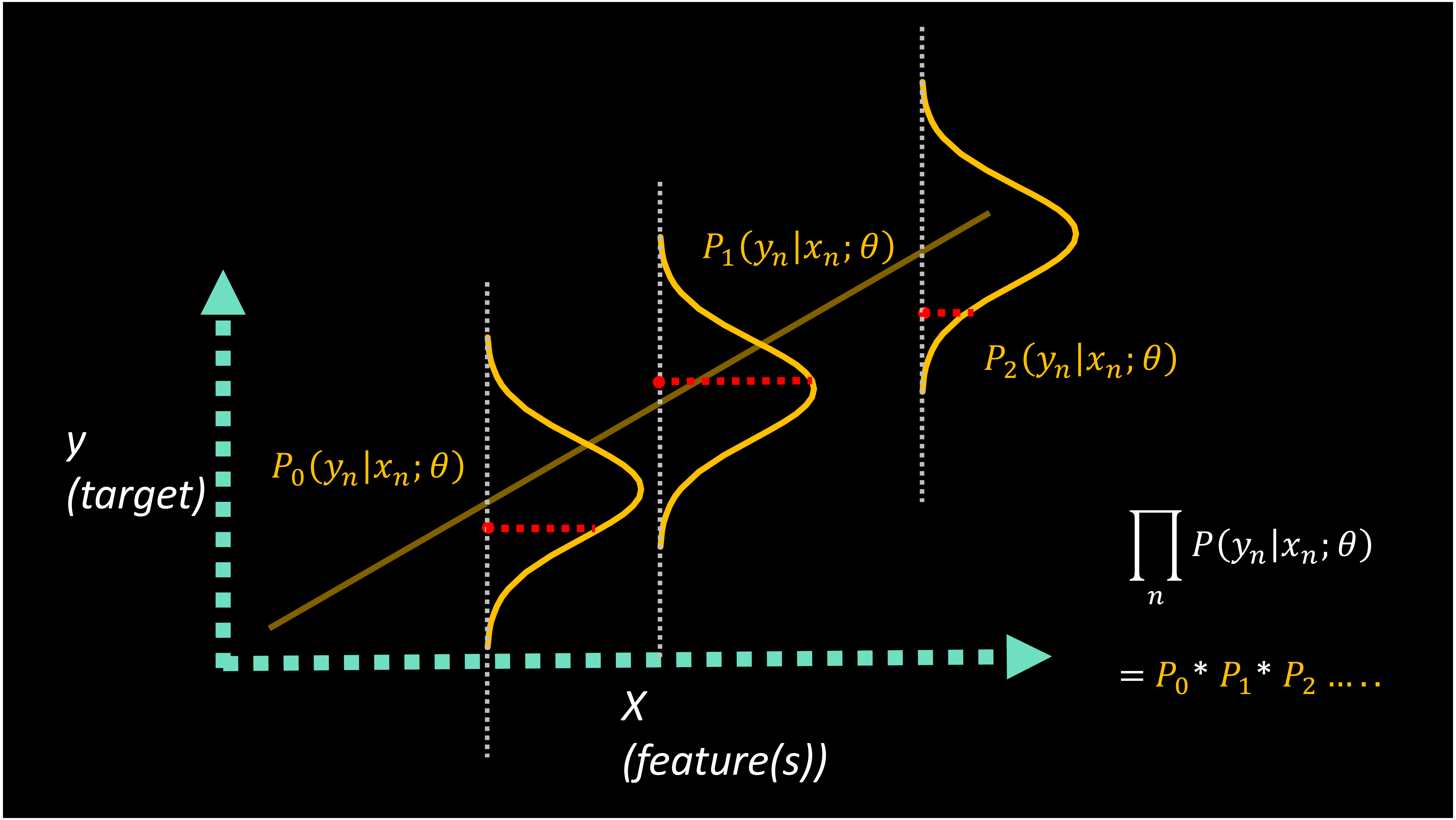
Linear Regression: Predicts continuous outcomes



Linear regression is a method of predicting an outcome based on input data.

A line represents the best-fit linear relationship between weight and size.

This line can be used to make predictions: given a new weight value, we can estimate the corresponding size.



$P_1(y_n|x_n;\theta)$
$P_2(y_n|x_n;\theta)$
$P_0(y_n|x_n;\theta)$
y
(target)
$\prod_n P(y_n|x_n;\theta)$
$= P_0 * P_1 * P_2 \ldots$
X
(feature(s))

Maximum Likelihood Estimation (MLE)

$$\theta_{MLE} = argmax_{\theta} P(y|X;\theta))$$

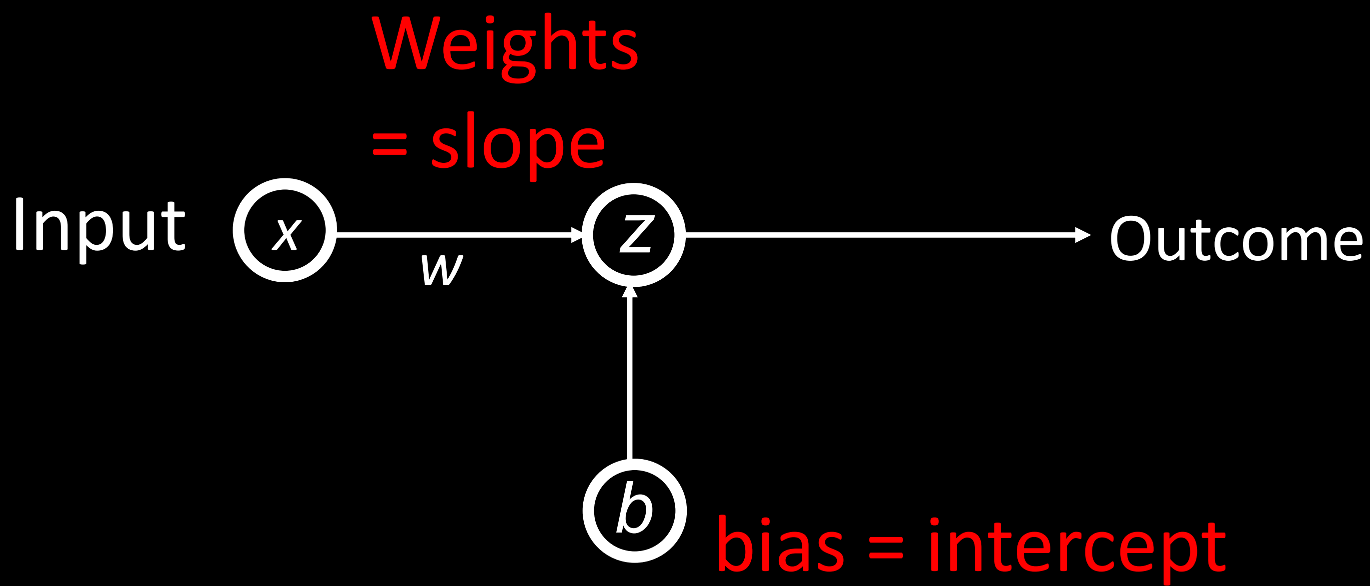
$P(y|X)$ (true) probability of seeing stroke given blood pressure

$P(y|X;\theta)$ Probability of seeing stroke given blood pressure from model with θ

$argmax_{\theta}$ Find the parameters θ to maximize the given probability $P(y|X,\theta)$

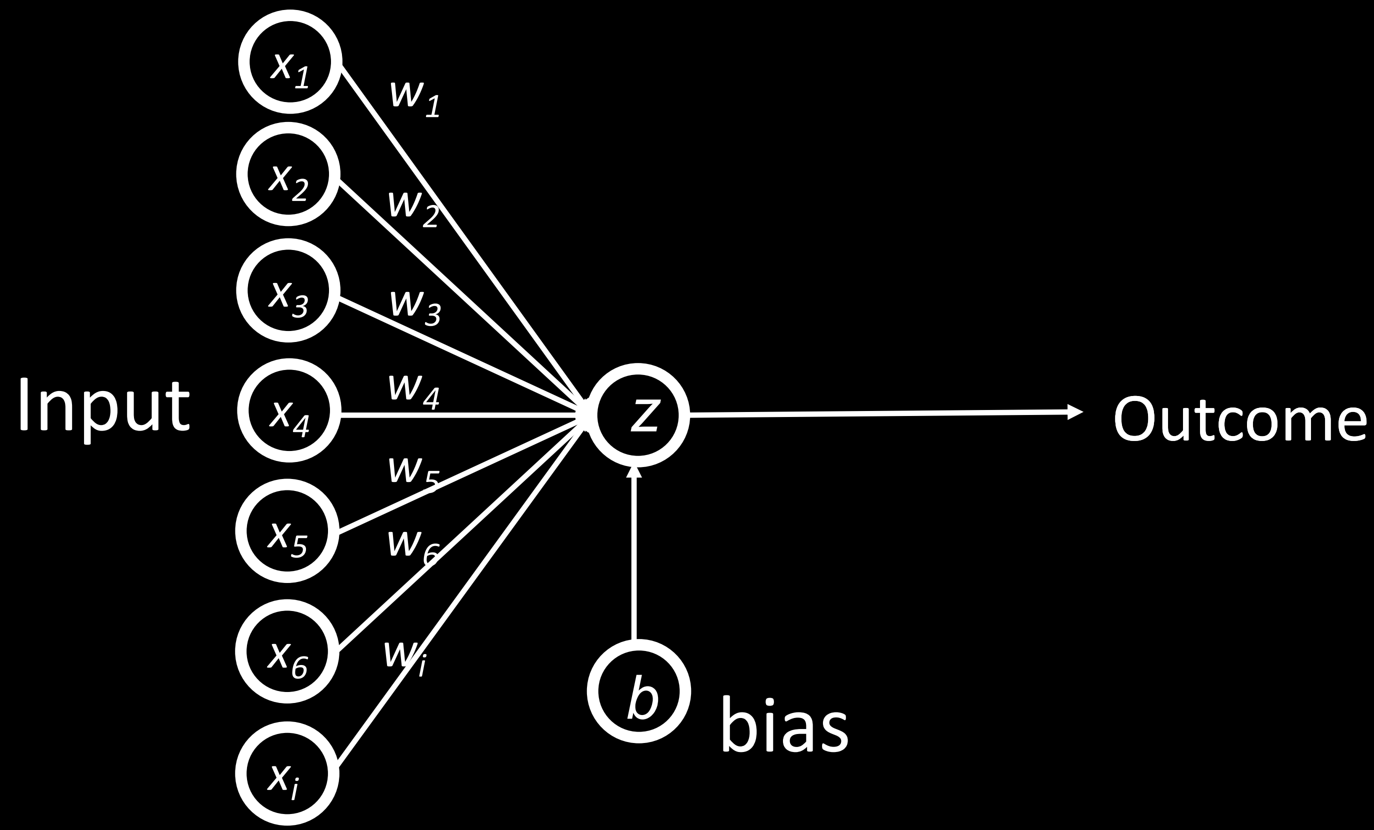
Graph of Linear Regression

$$y = WX + b$$

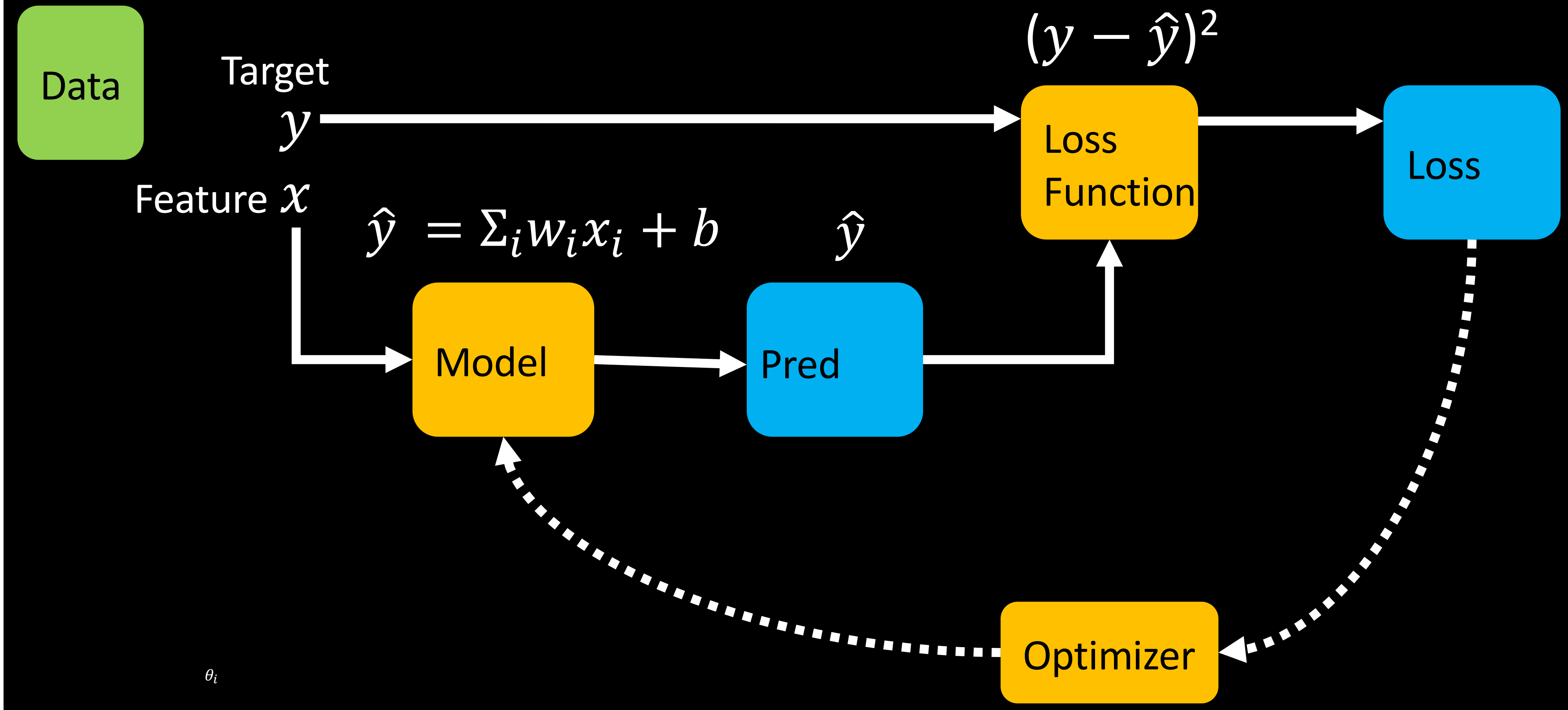


Graph of Linear Regression

$$y = w_1x_1 + w_2x_2 + \ldots.. + b$$



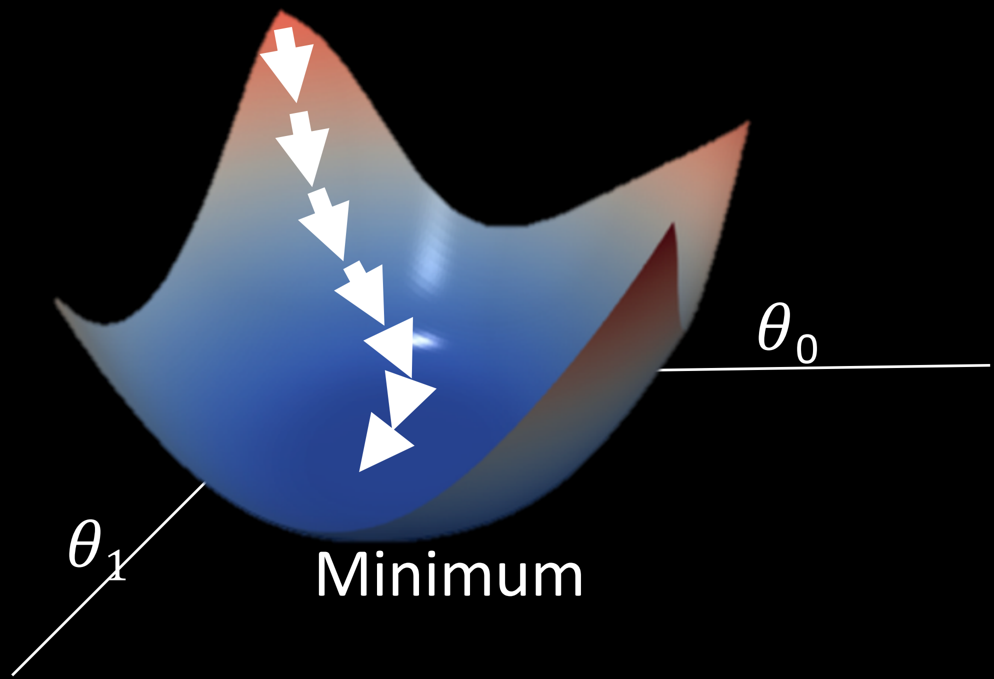
Flow Chart for Linear Regression



Gradient Descent

Moves in the direction of steepest descent (negative gradient) to minimize the objective function.

Steps:
1. Start with initial parameter values
2. Calculate the gradient of the objective function
3. Update parameters in the opposite direction of the gradient
4. Repeat steps 2-3 until convergence

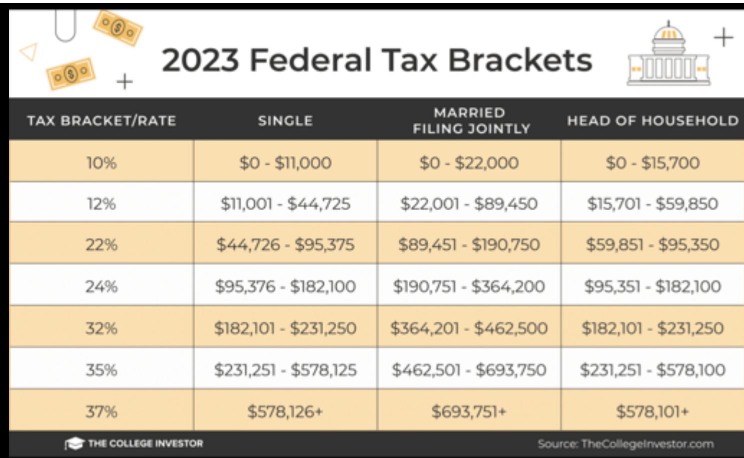


Classification: Logistic Regression

Classification vs Regression

	TNM 8th - Primary tumor characteristics
T_x	Tumor in sputum/bronchial washings but not be assessed in imaging or bronchoscopy
T_0	No evidence of tumor
T_{is}	Carcinoma in situ
T_1	≤ 3 cm surrounded by lung/visceral pleura, not involving main bronchus
$T_{1a(mi)}$	Minimally invasive carcinoma
T_{1a}	≤ 1 cm
T_{1b}	> 1 to ≤ 2 cm
T_{1c}	> 2 to ≤ 3 cm
T_2	> 3 to ≤ 5 cm *or* involvement of main bronchus without carina, regardless of distance from carina or invasion visceral pleural *or* atelectasis or post obstructive pneumonitis extending to hilum
T_{2a}	>3 to ≤4cm
T_{2b}	>4 to ≤5cm
T_3	>5 to ≤7cm in greatest dimension *or* tumor of any size that involves chest wall, pericardium, phrenic nerve *or* satellite nodules in the same lobe
T_4	> 7cm in greatest dimension *or* any tumor with invasion of mediastinum, diaphragm, heart, great vessels, recurrent laryngeal nerve, carina, trachea, oesophagus, spine *or* separate tumor in different lobe of ipsilateral lung
N_1	Ipsilateral peribronchial and/or hilar nodes and intrapulmonary nodes
2	Ipsilateral mediastinal and/or subcarinal nodes
3	Contralateral mediastinal or hilar; ipsilateral/contralateral scalene/ supraclavicular
M_1	Distant metastasis
M_{1a}	Tumor in contralateral lung or pleural/pericardial nodule/malignant effusion
M_{1b}	Single extrathoracic metastasis, including single non-regional lymphnode
M_{1c}	Multiple extrathoracic metastases in one or more organs

Human has a tendency (or need) to categorize things for our own perception



2023 Federal Tax Brackets

TAX BRACKET/RATE	SINGLE	MARRIED FILING JOINTLY	HEAD OF HOUSEHOLD
10%	$0 - $11,000	$0 - $22,000	$0 - $15,700
12%	$11,001 - $44,725	$22,001 - $89,450	$15,701 - $59,850
22%	$44,726 - $95,375	$89,451 - $190,750	$59,851 - $95,350
24%	$95,376 - $182,100	$190,751 - $364,200	$95,351 - $182,100
32%	$182,101 - $231,250	$364,201 - $462,500	$182,101 - $231,250
35%	$231,251 - $578,125	$462,501 - $693,750	$231,251 - $578,100
37%	$578,126+	$693,751+	$578,101+

THE COLLEGE INVESTOR

Source: TheCollegeInvestor.com

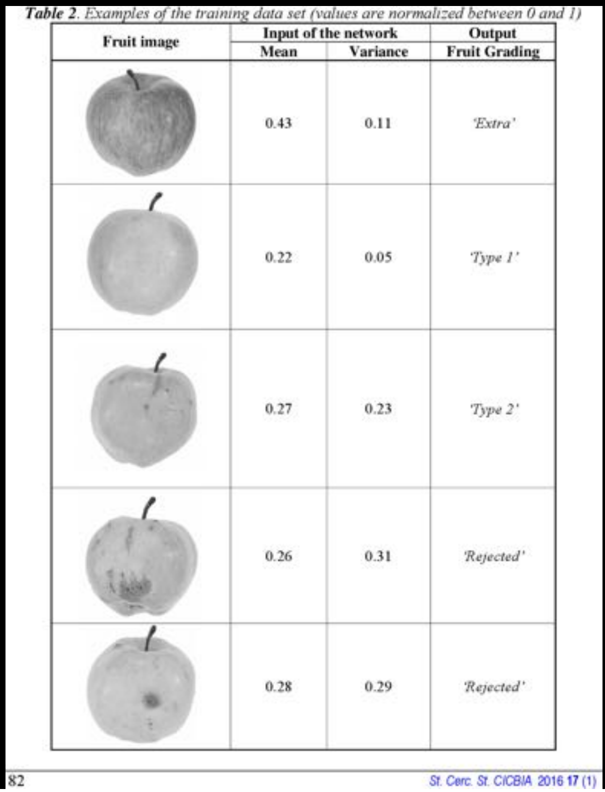
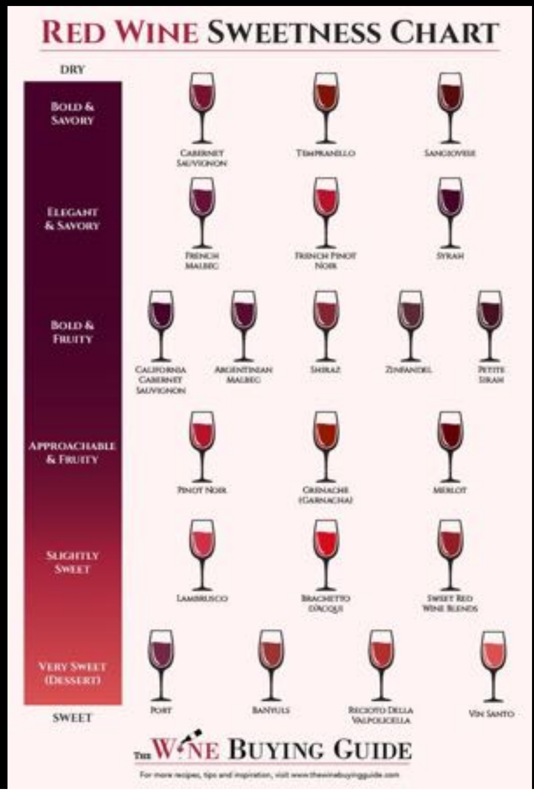


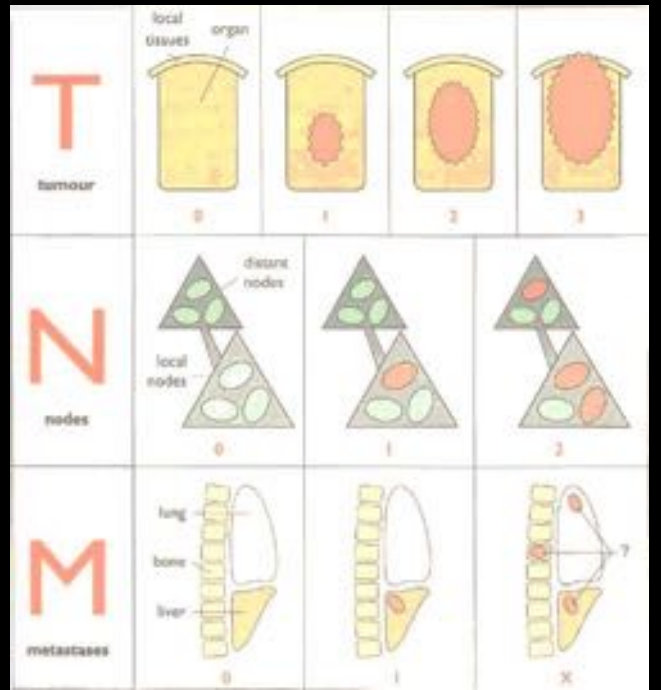
Table 2. Examples of the training data set (values are normalized between 0 and 1)

Fruit image	Input of the network Mean	Input of the network Variance	Output Fruit Grading
	0.43	0.11	'Extra'
	0.22	0.05	'Type 1'
	0.27	0.23	'Type 2'
	0.26	0.31	'Rejected'
	0.28	0.29	'Rejected'

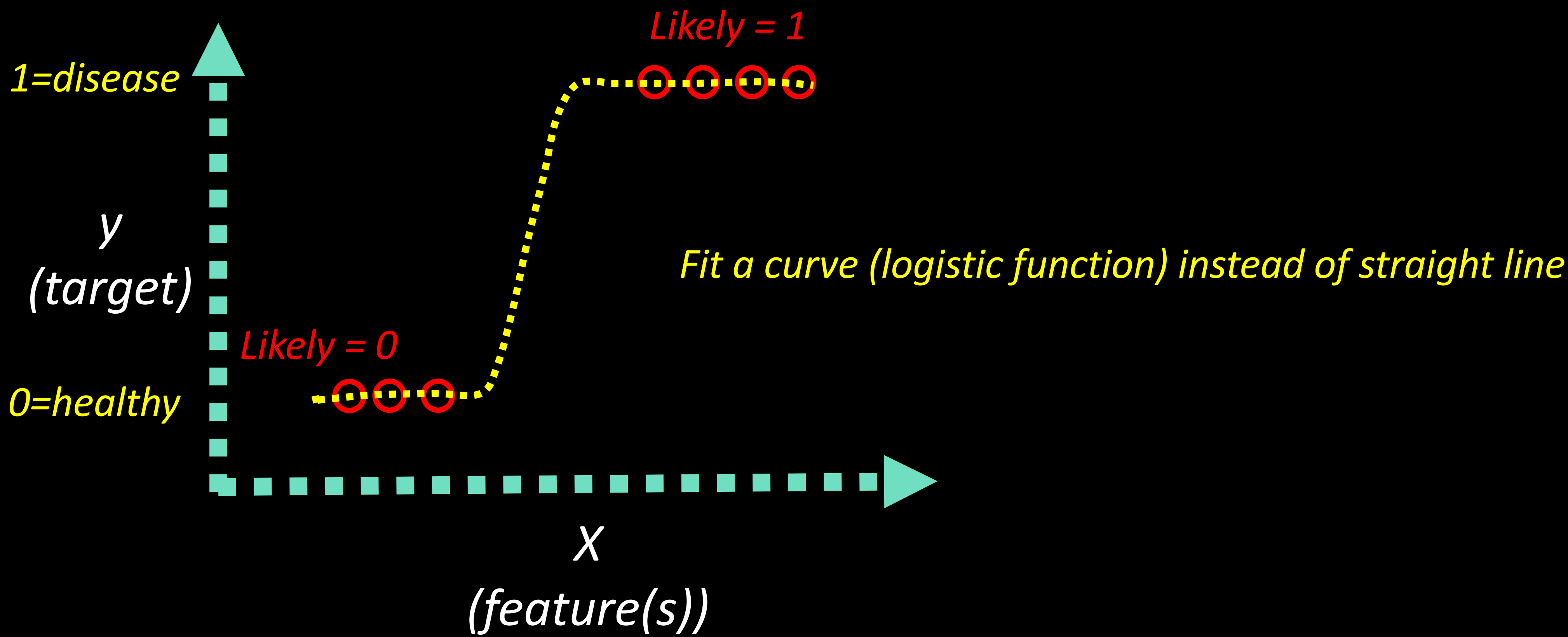
82

St. Cerc. St. CICBIA 2016 17 (1)

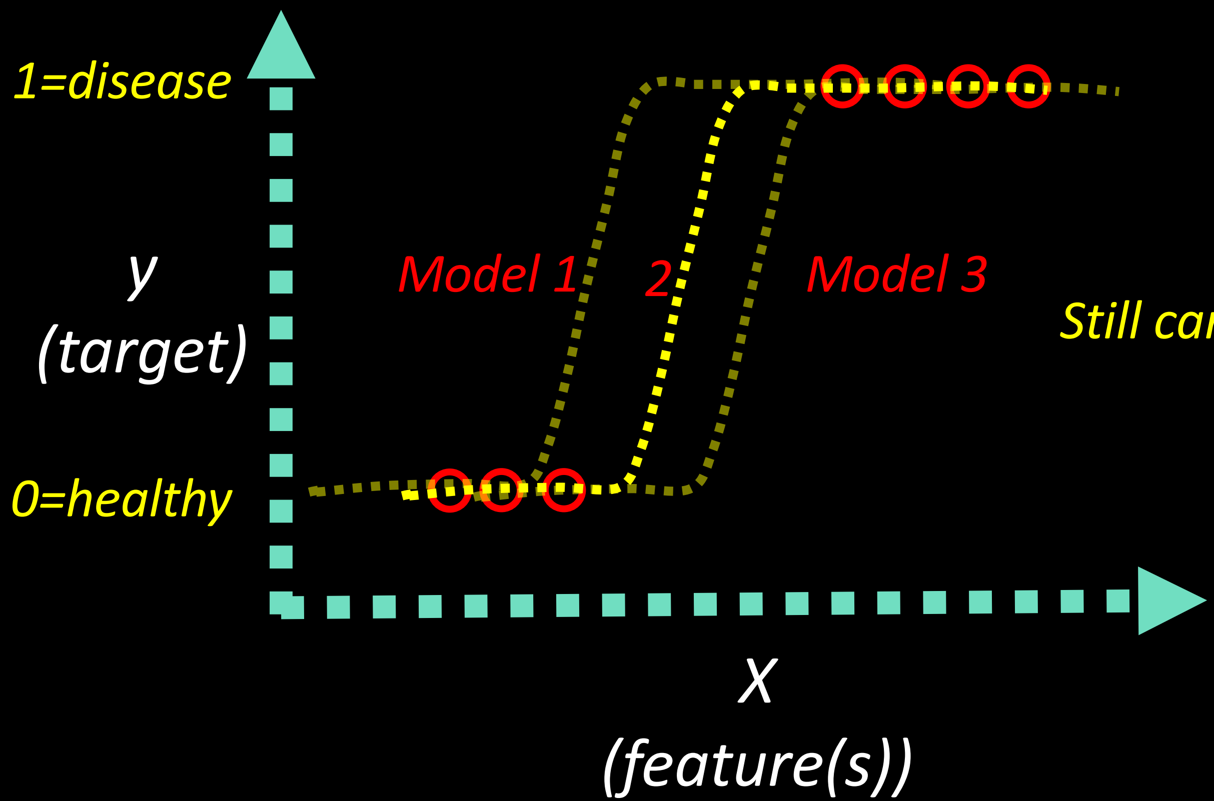




Logistic Regression: Predicts categorical outcomes



Logistic Regression: Predicts categorical outcomes



Still can apply maximum likelihood

Regression -> classification, how?

regression $\Sigma_i w_i x_i + b$ $-\infty \sim \infty$ y_i $-\infty \sim \infty$

classification $0 < P(Y = truck) < 1$ $P(Y = car) = 1 - P(Y = truck)$

We want to find a way to map $-\infty \sim \infty$ to 0-1

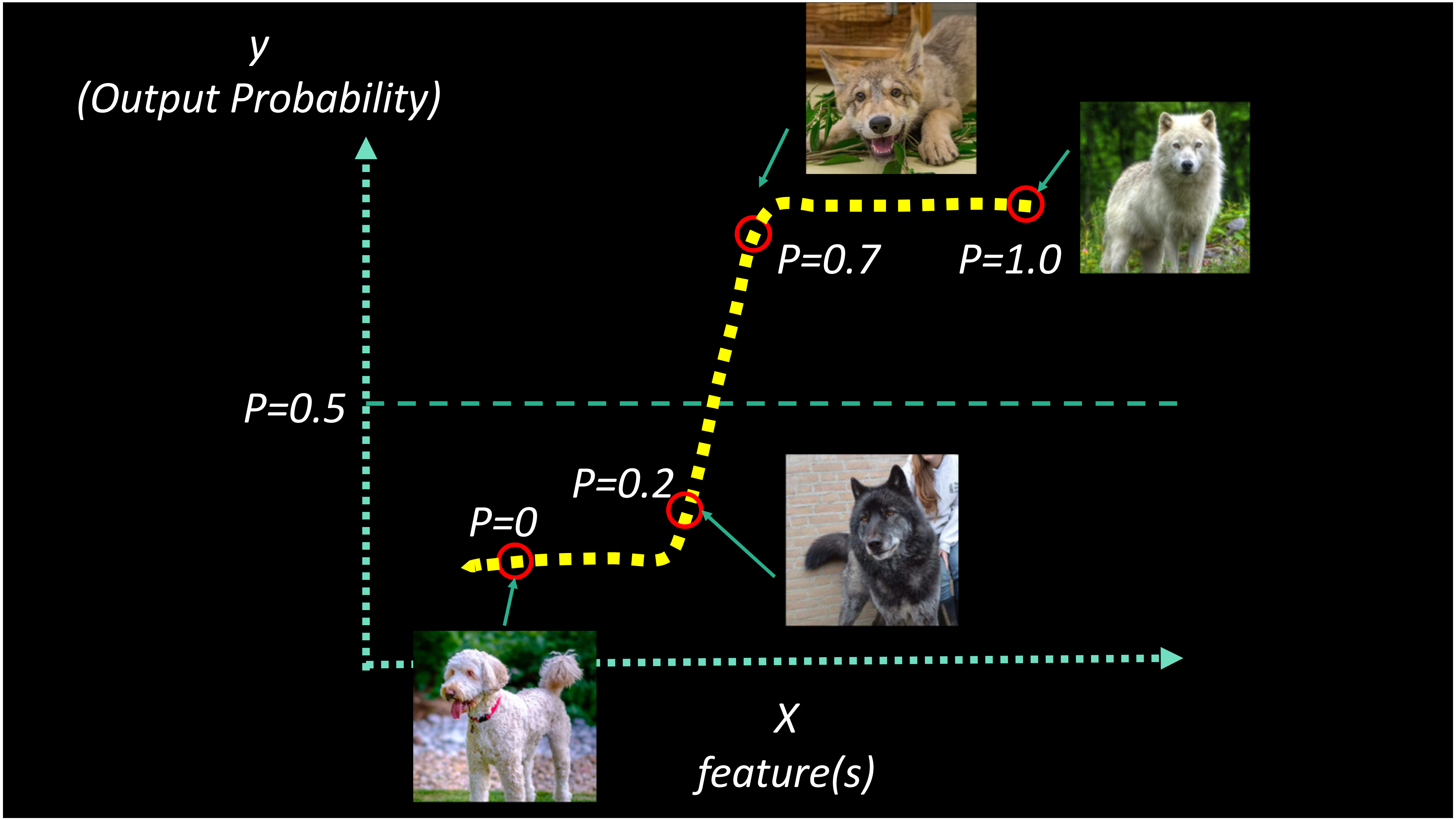
Chance of wolf (=1)

0 (very sure)

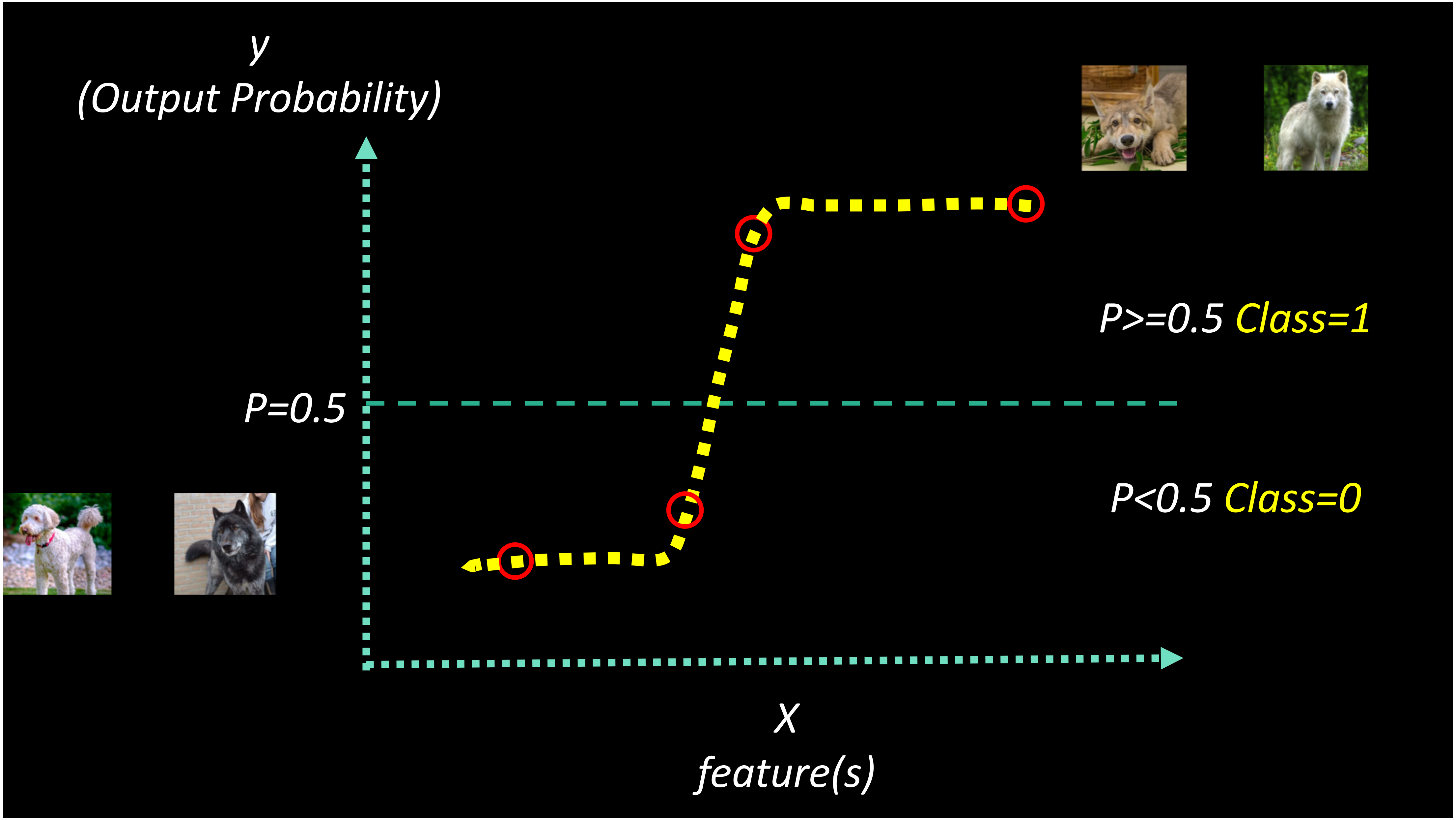
0.2 (not sure)

0.7 (not sure)

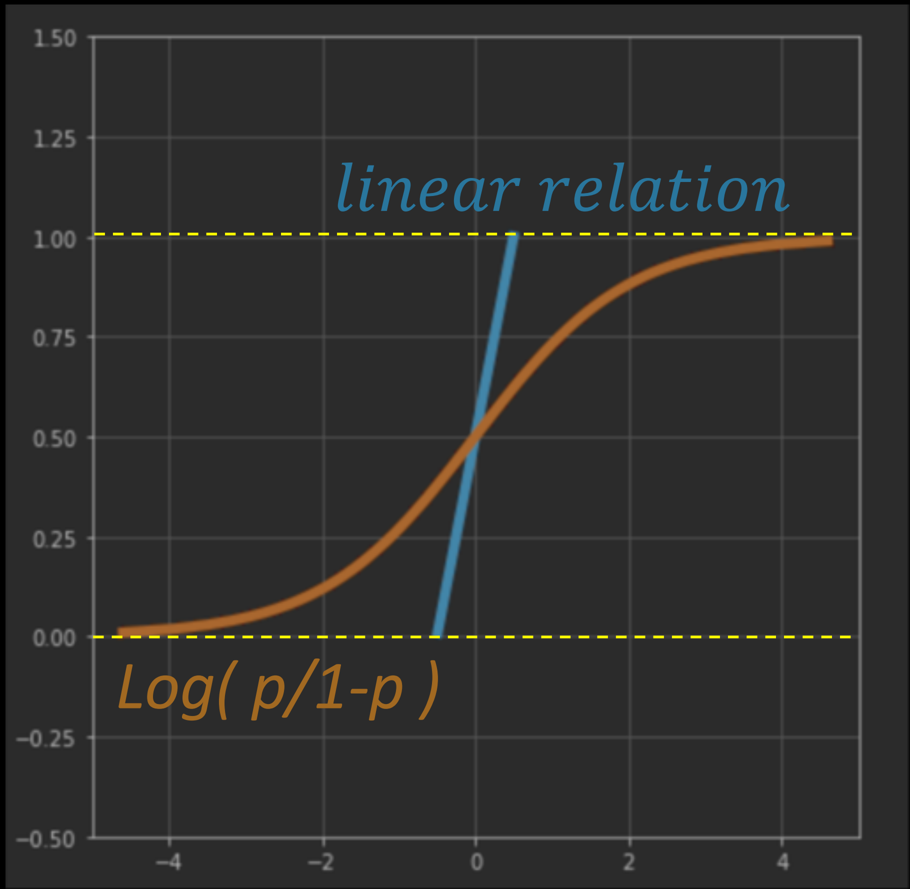
1 (very sure)



y
(Output Probability)
P=0.7
P=1.0
P=0.5
P=0.2
P=0
X
feature(s)


y
(Output Probability)
P=0.5
P>=0.5 Class=1
P<0.5 Class=0
X
feature(s)

Probability
0-1



feature $-\infty \sim \infty$

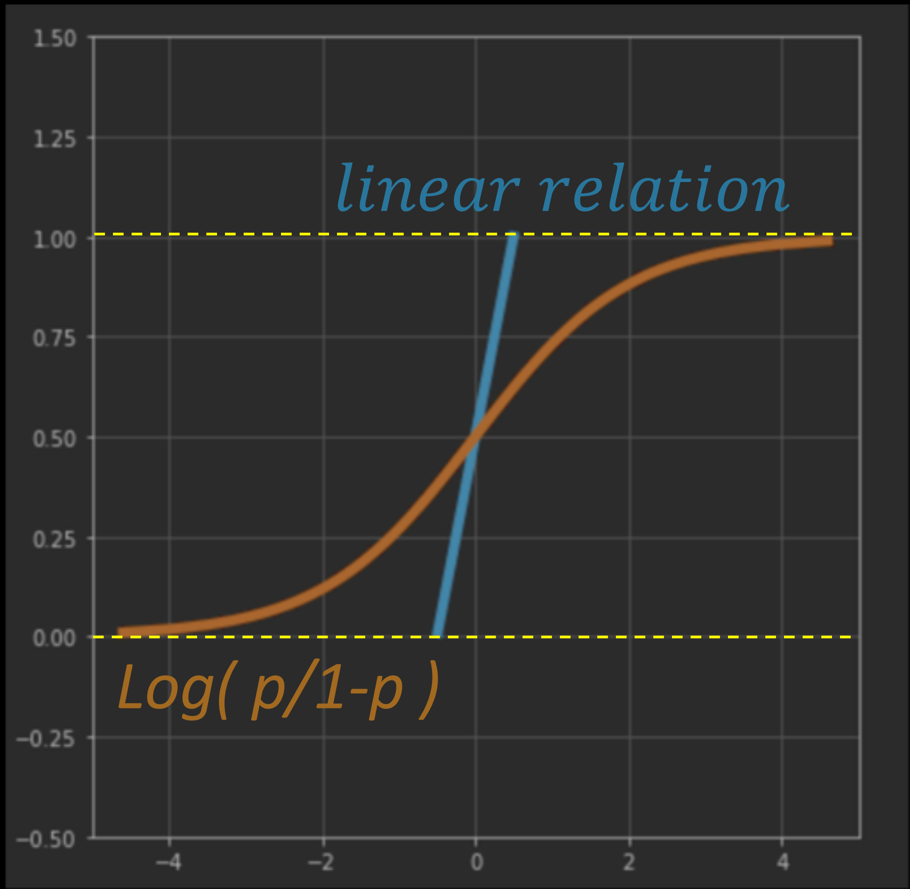
How to predict probability?

Remember we got output of range

$-\infty \sim \infty$

From regression

Probability
0-1



feature $-\infty \sim \infty$

Probability p $0 \sim 1$

Logit $\log(\frac{p}{1-p})$ $-\infty \sim \infty$

same range!

linear regression $\Sigma_i w_i x_i + b$ $-\infty \sim \infty$

(Odd Ratio) $\frac{p}{1-p}$ $0 \sim \infty$

	No. of case patients	No. of control subjects	Odds ratio	95% confidence interval	*P* for trend*
Nonsmokers	117	1750	1.0	Referent	
Duration of tobacco use, y					
0.1–20.0	3	33	1.3	0.4–4.5	
20.1–32.0	7	33	3.4	1.4–8.0	
32.1–44.0	21	33	13.3	7.2–24.9	
⩾44.1	30	30	19.1	10.4–35.1	<.0001
Average consumption of tobacco, g/day					
0.1–3.5	2	10	2.2	0.5–10.4	
3.6–5.0	22	54	7.9	4.3–14.3	
5.1–10.7	6	18	4.8	1.9–12.6	
⩾10.8	31	47	12.4	7.2–21.4	.1
Cumulative consumption of tobacco, g/day × y					
1–71	3	31	1.3	0.4–4.4	
72–157	13	34	7.6	3.8–15.4	
158–382	15	32	8.0	4.0–15.7	
⩾383	30	32	18.3	10.2–32.8	.0006
Age at start of tobacco use, y					
⩽19	27	48	9.6	5.6–16.7	
20–26	20	52	6.3	3.5–11.2	
⩾27	14	29	8.2	4.1–16.3	.4

*Test for linear trend, two-sided *P* value (considered statistically significant for $P<.05$).

Now we have a good target for regression:

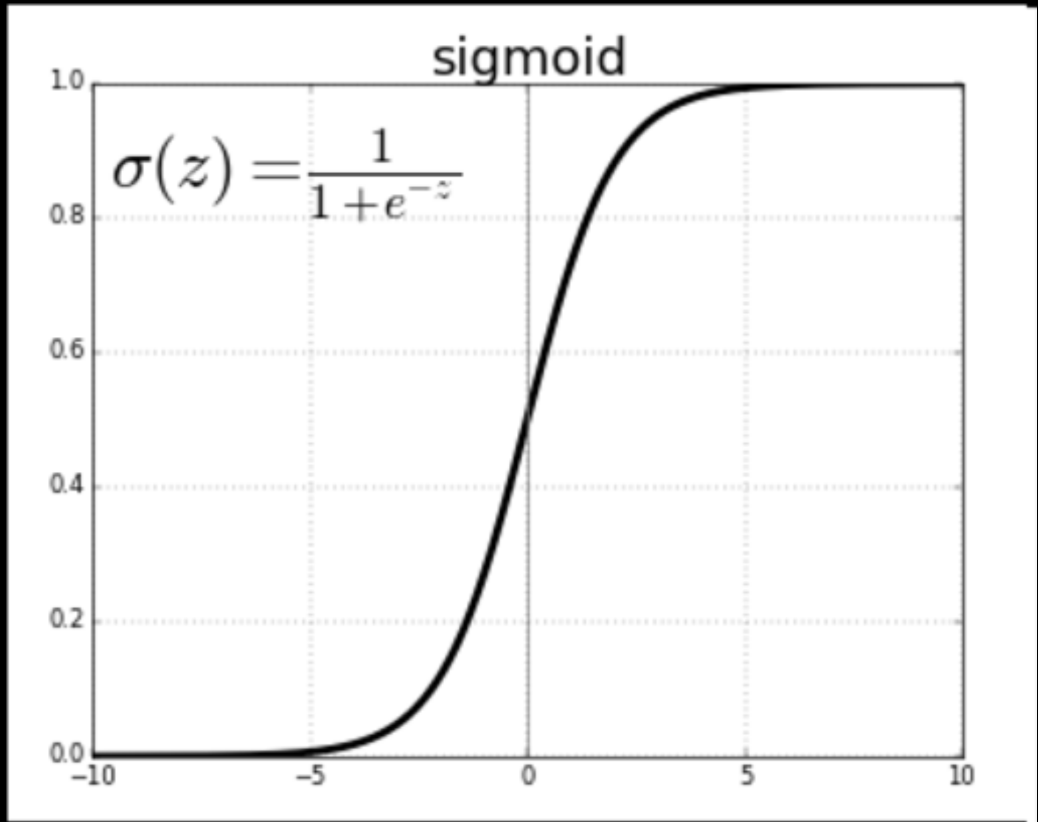
$\Sigma_i w_i x_i + b$ $-\infty \sim \infty$ *to* *Logit* $\log(\frac{p}{1-p})$ $-\infty \sim \infty$

$$p = \frac{1}{1+exp-(\Sigma_i w_i x_i + b)} = \text{Sigmoid}(w_i x_i + b) = S(\beta X)$$

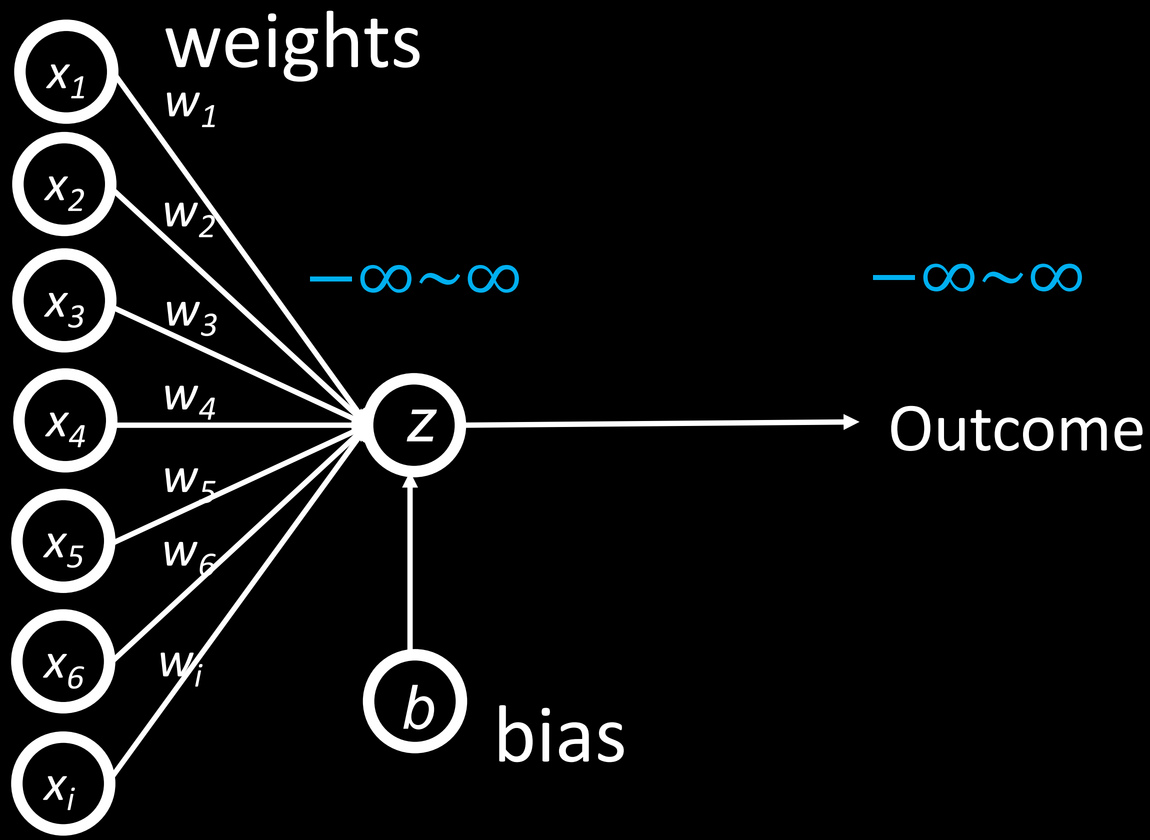
$$P = S(\beta X)$$

Logistic Regression = Regression Between Probability and Sigmoid of (weighted features)

```python
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Create x values from -6 to 6
x = np.linspace(-6, 6, 1000)

# Calculate sigmoid values
y_sigmoid = sigmoid(x)
# Plot sigmoid
plt.plot(x, y_sigmoid)
```

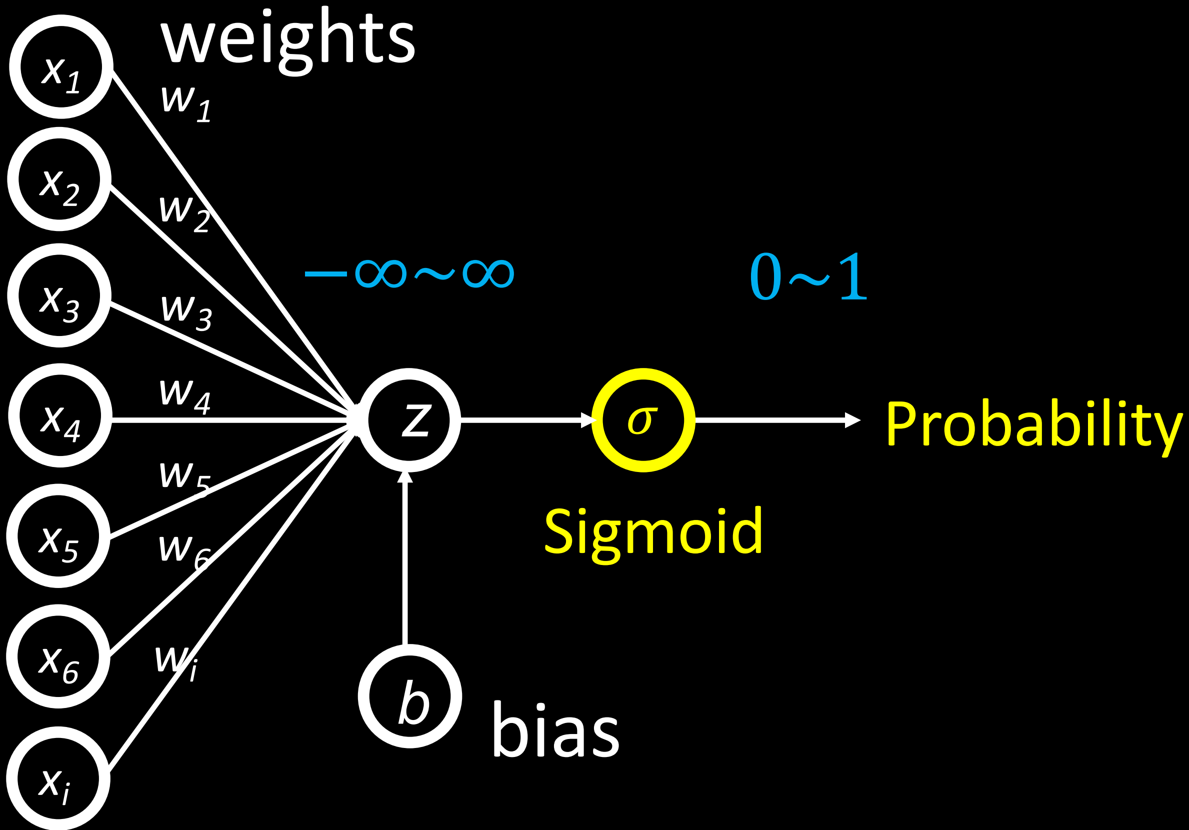


Graph of Linear Regression

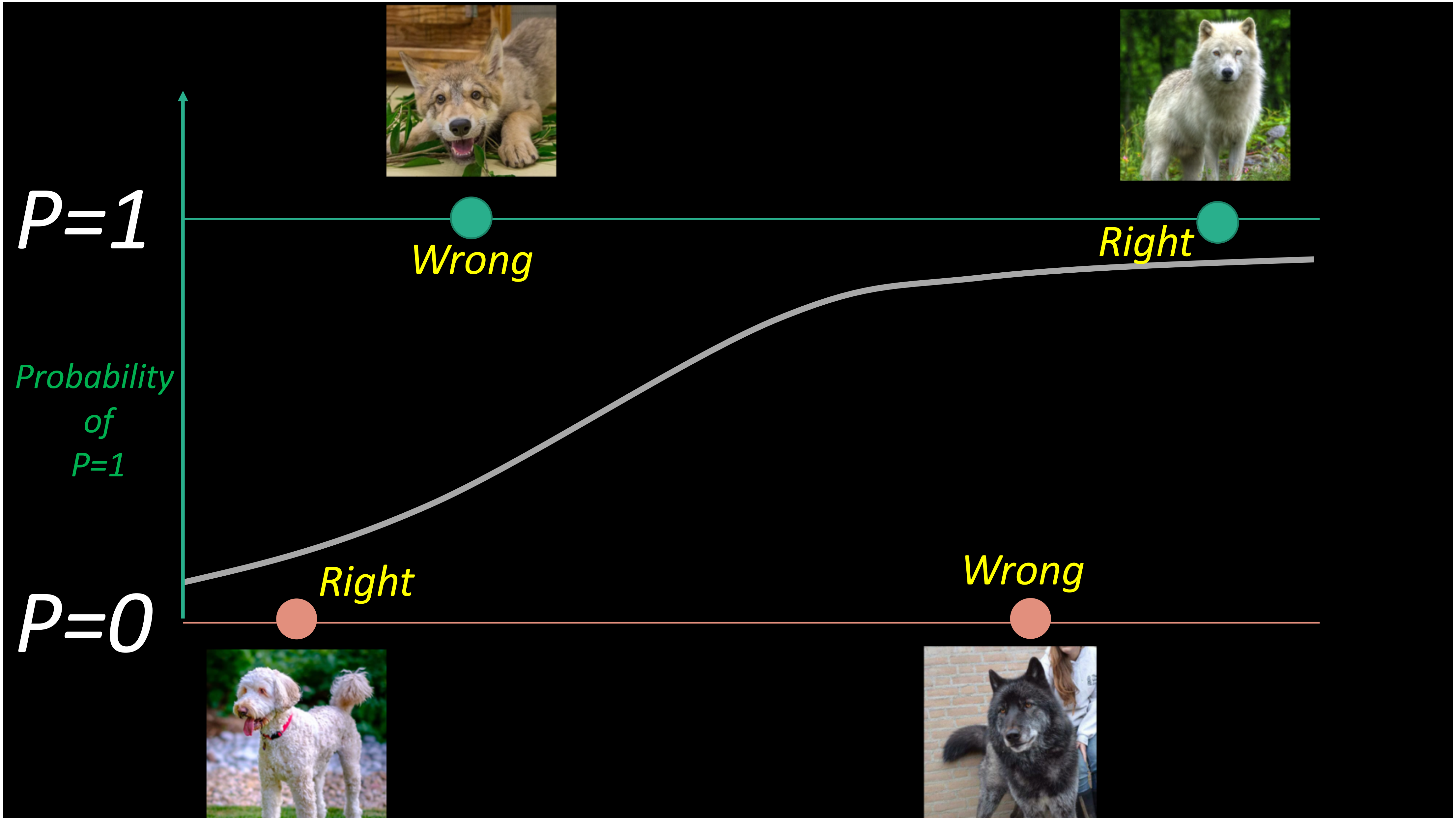


Graph of Logistic Regression

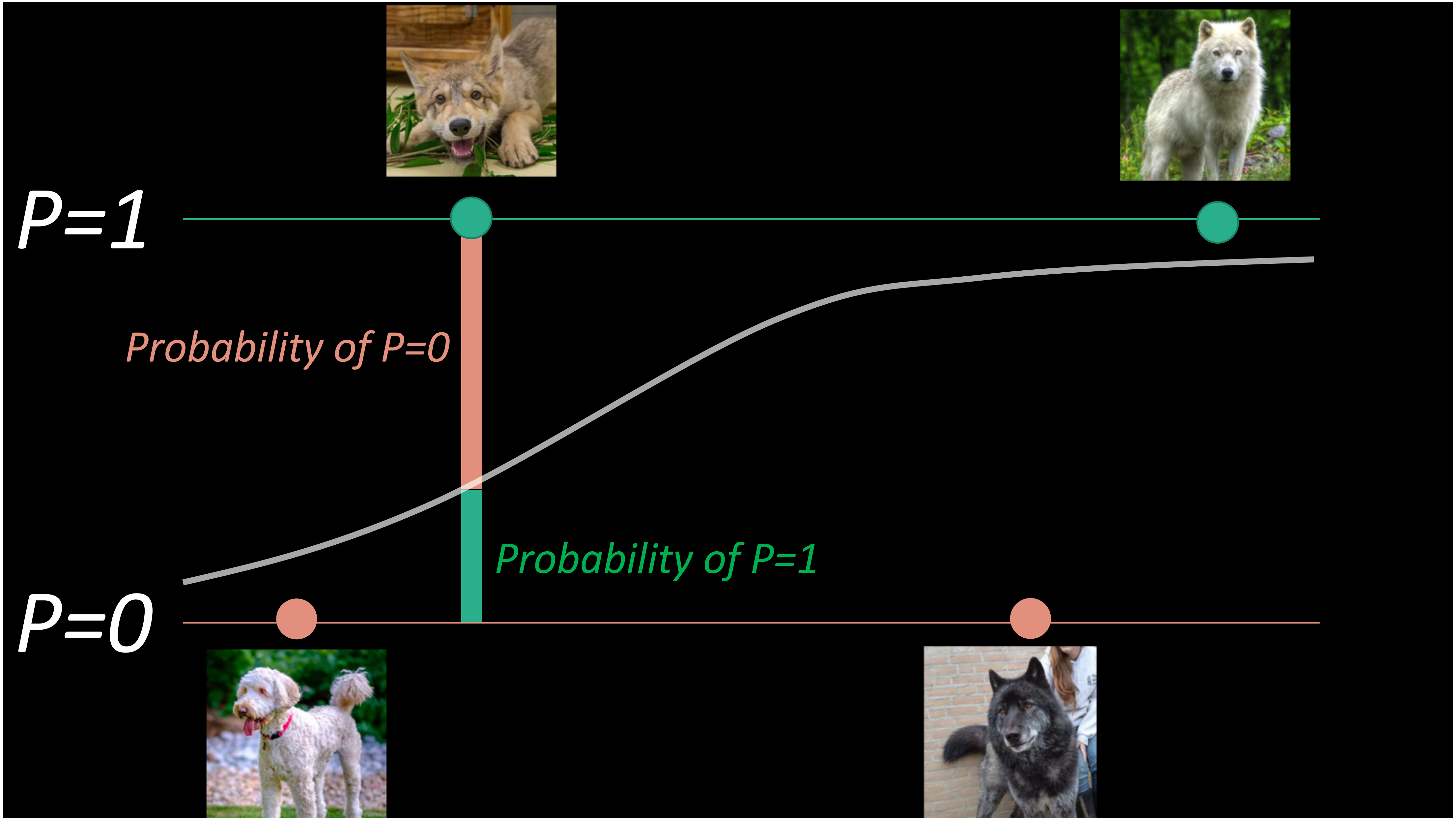
$$p = \frac{1}{1+exp-(\Sigma_i w_i x_i + b)} = \text{Sigmoid}(\Sigma_i w_i x_i + b) = S(WX)$$



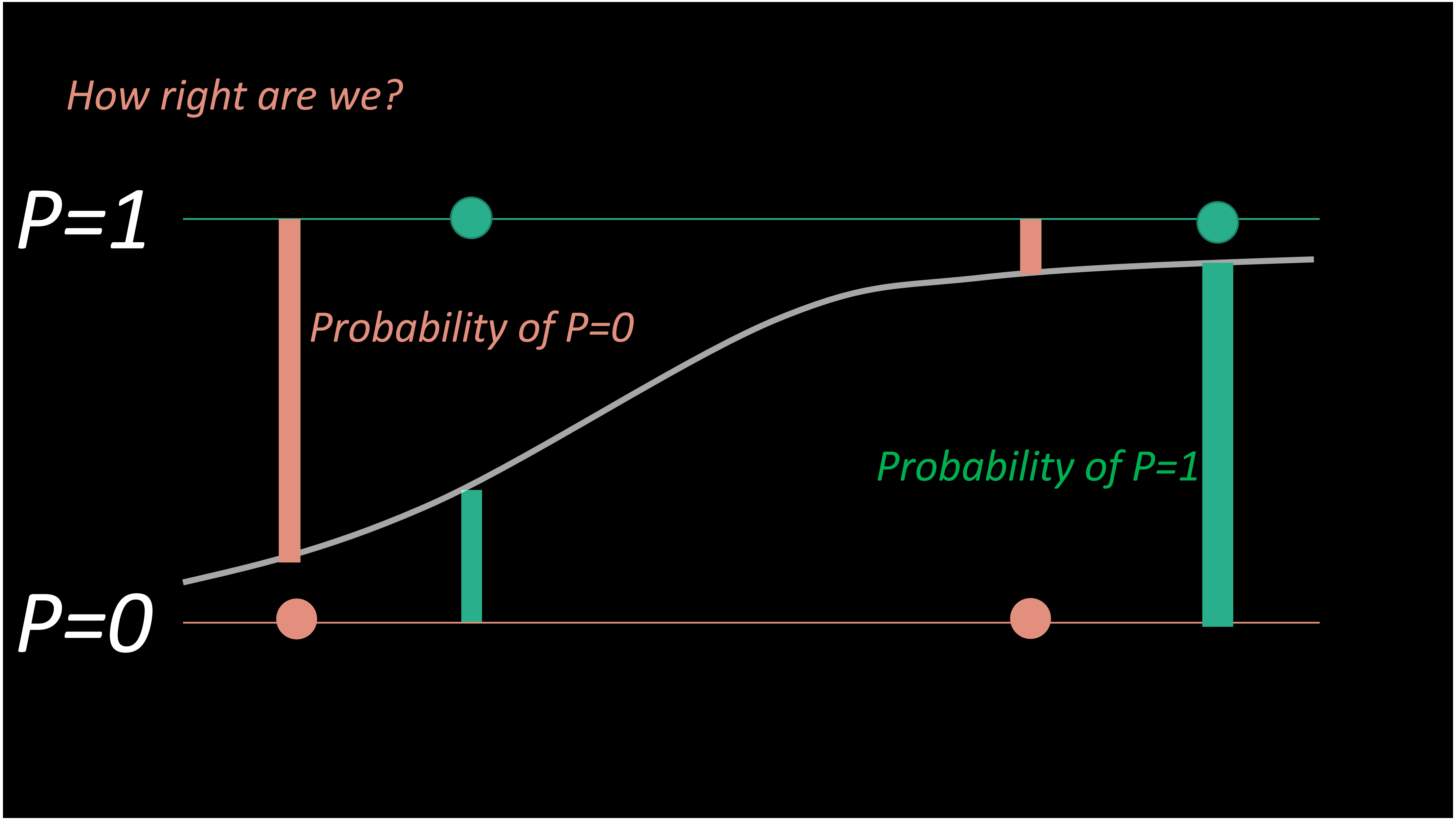
Loss Function for Classification

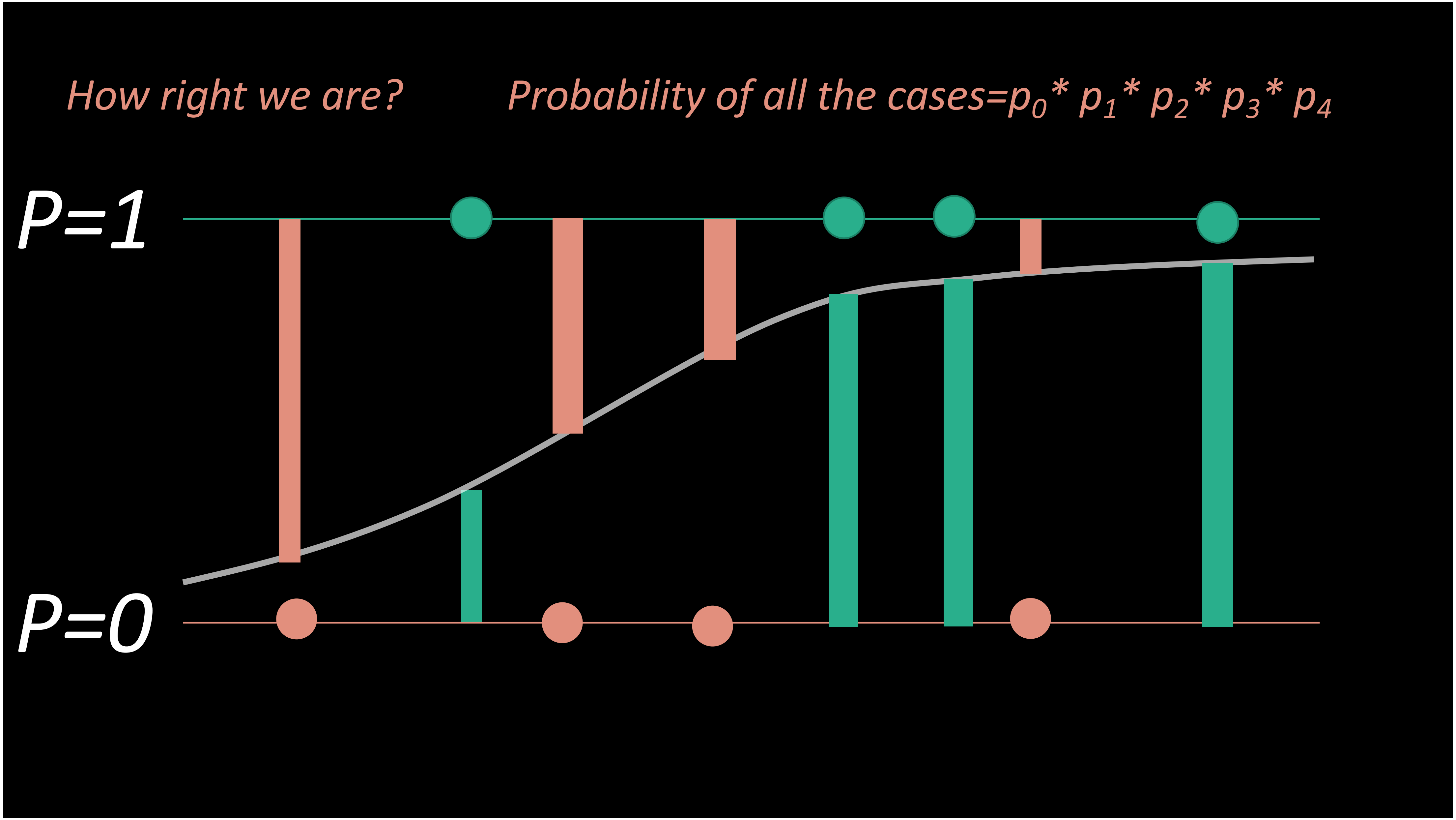


P=1
Wrong
Right
Probability
of
P=1
Right
Wrong
P=0



P=1
Probability of P=0
Probability of P=1
P=0


How right are we?
P=1
Probability of P=0
Probability of P=1
P=0


How right we are?
Probability of all the cases=$p_0 * p_1 * p_2 * p_3 * p_4$
P=1
P=0

Probability of all the cases

$(+)\ p_0 * p_1 * p_2 * \ldots \ldots$

$(-)\ p_0 * p_1 * p_2 * \ldots \ldots$

Remember

$$\prod_n P(y_n | x_n; \theta)$$

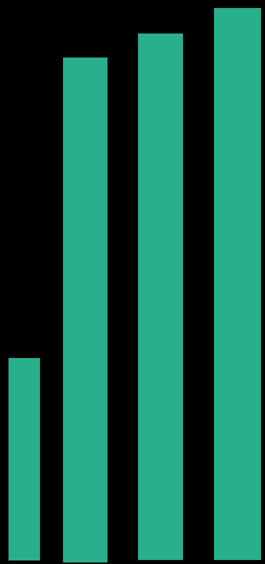
VS

$$\sum_i log P(y_n | x_n; \theta)$$

Binary Cross Entropy (BCE) loss

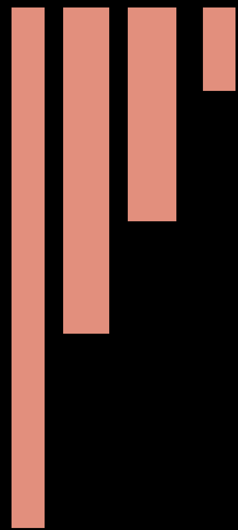
y = 1

BCE= $-\sum_n \overset{1}{y} \cdot \log(p(\hat{y}))$



y = 0

BCE= $-\sum_n (1 \overset{1}{-} y) \cdot \log(p(1-\hat{y}))$



Binary Cross Entropy (BCE) loss

$$BCE =$$

$$-\sum_{n}[\mathrm{y} \cdot \log(p(\hat{y})) \quad + \quad (1-y) \cdot \log(p(1-\hat{y}))]$$

Loss if y=1

Loss if y=0

Compare these Models

Assuming Truth = 1

Very Correct

P(0)	P(1)
0.3	0.7

Correct

P(0)	P(1)
0.45	0.55

Should be rewarded

Wrong

P(0)	P(1)
0.7	0.3

Very Wrong

P(0)	P(1)
0.9	0.1

Should be punished

Compare these Models

Assuming Truth = 1

Very Correct

P(0)	P(1)
0.3	0.7

Should be rewarded more

Correct

P(0)	P(1)
0.45	0.55

Should be rewarded less

Wrong

P(0)	P(1)
0.7	0.3

Should be punished less

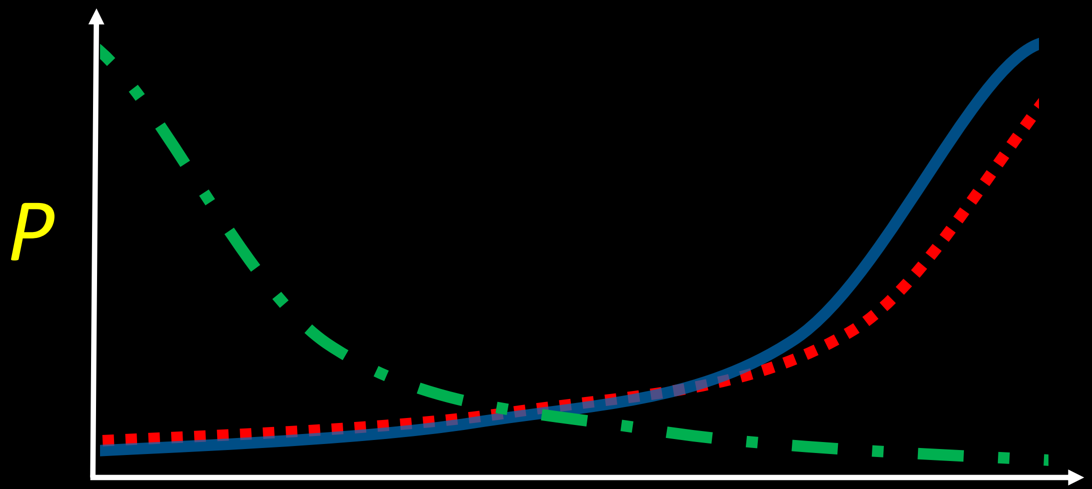
Very Wrong

P(0)	P(1)
0.9	0.1

Should be punished more

Again, we can use Maximum Likelihood Estimation (MLE)

$$\theta_{ML} = argmax_{\theta} P(y|X;\theta)$$



Q, Real Probability

$P(y|X;\theta)$, good prediction

$P(y|X;\theta)$, bad prediction

Cross Entropy:

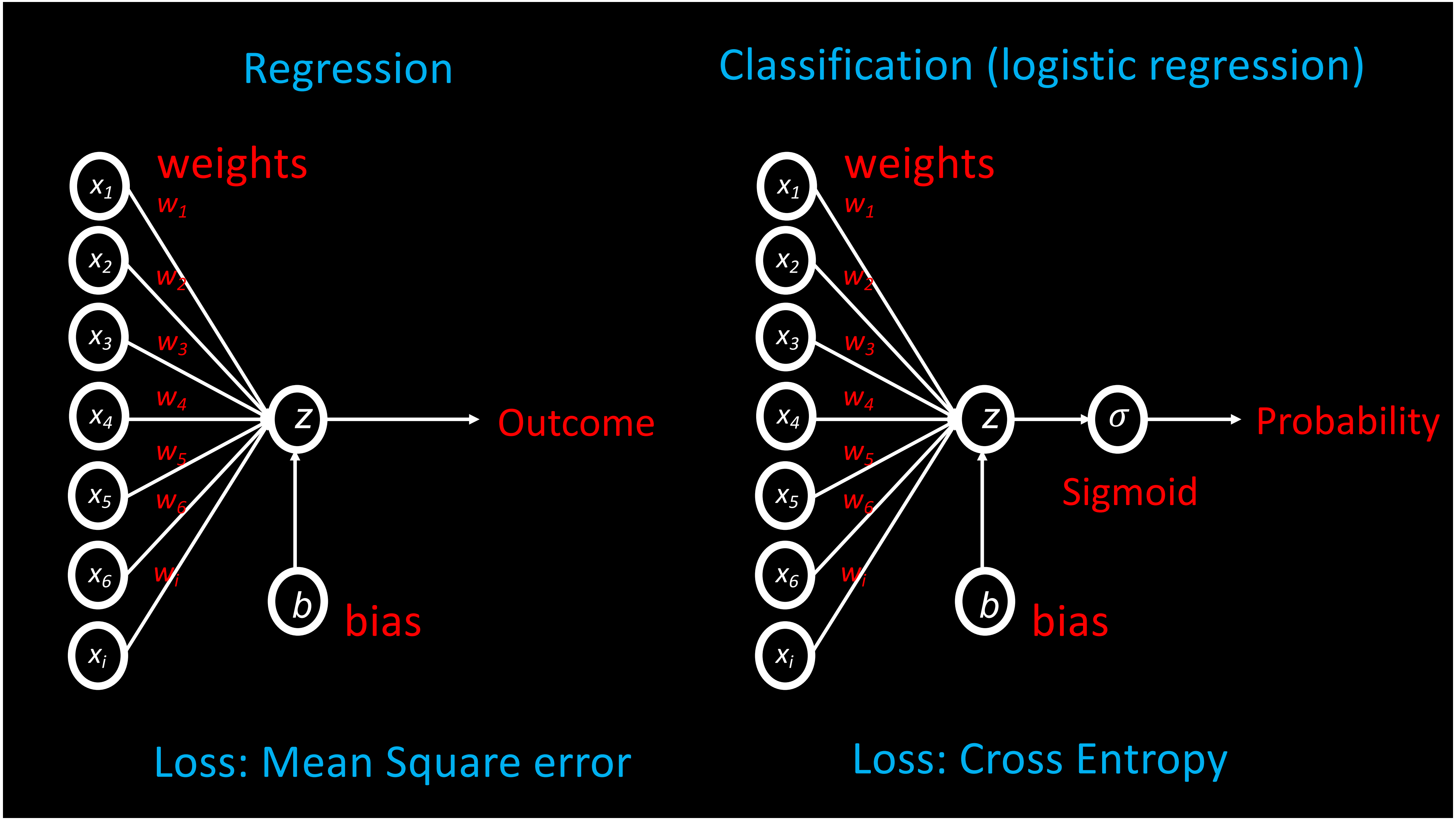
how different they are?

Cross Entropy:

how different they are?

$$H(Q,P) = -\sum_n Q_i \cdot \log(P_i)$$

	Q	***P***			
Good model	0.9	0.9	(0.09)	0.32	*Good*
	0.1	0.1	(0.23)		
Bad model	0.9	0.1	(2.07)	2.08	*Bad*
	0.1	0.9	(0.01)		


Regression
Classification (logistic regression)
weights
x_1 x_2 x_3 x_4 x_5 x_6 x_i
w_1 w_2 w_3 w_4 w_5 w_6 w_i
z
b
bias
Outcome
σ
Sigmoid
Probability
Loss: Mean Square error
Loss: Cross Entropy

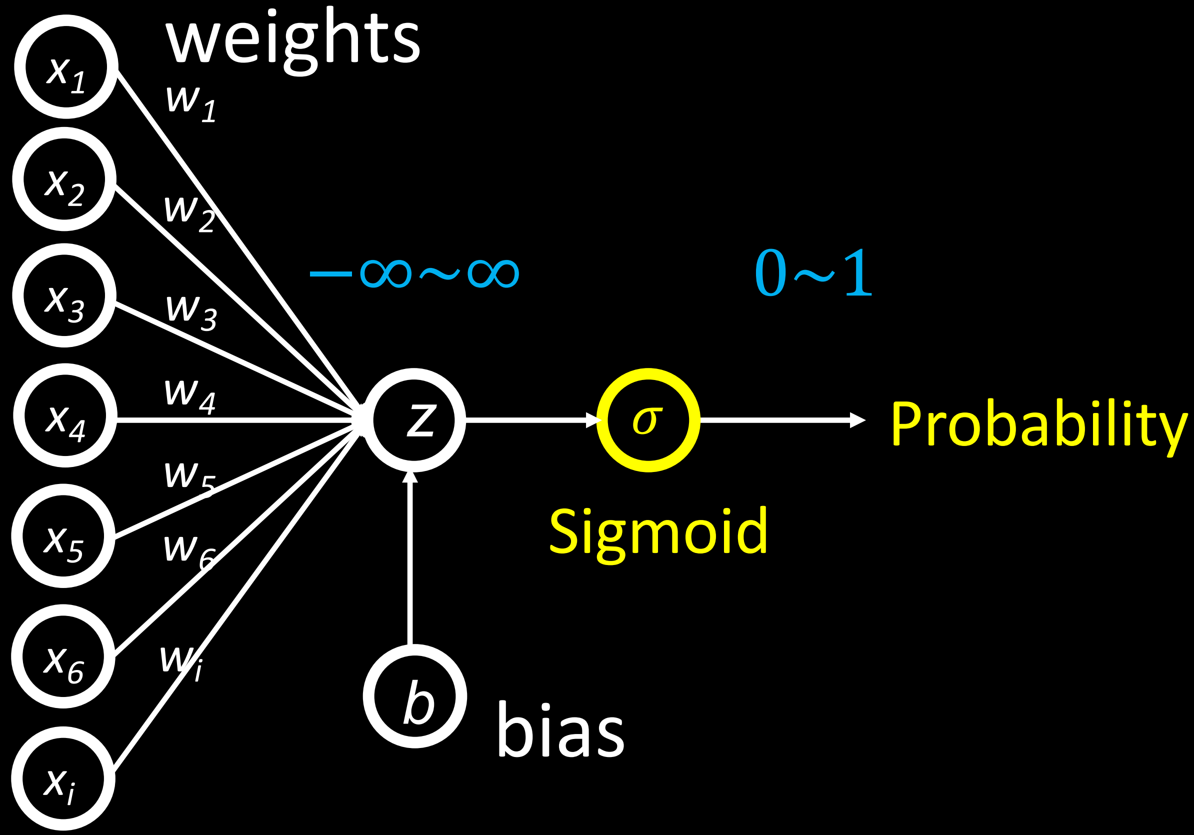
In practice!

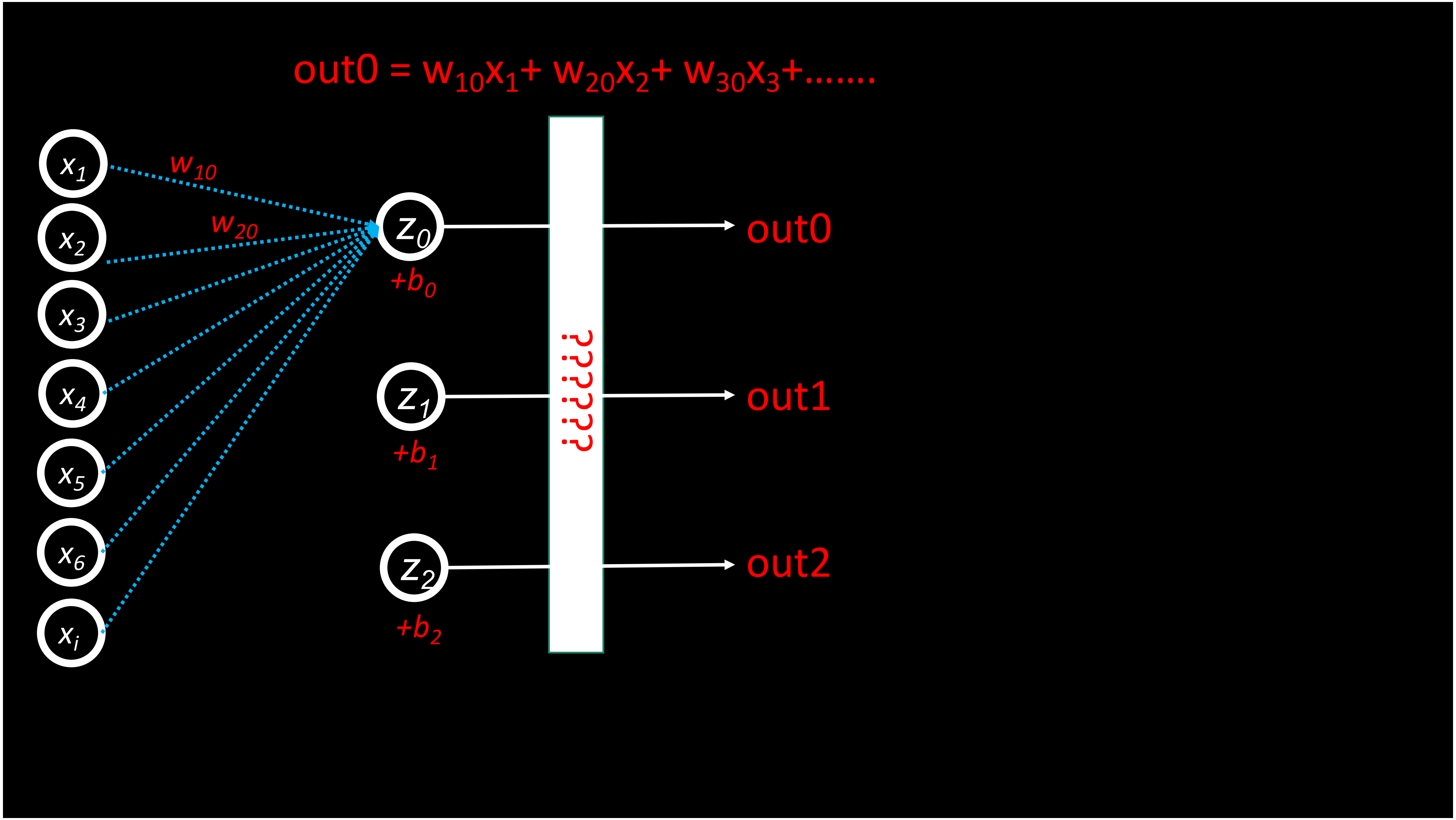
```
model = LinearRegression()
model.fit(weights, dosages)

log_model = LogisticRegression()
log_model.fit(cholesterol, heart_disease)
```

Multi-class Logistic Regression

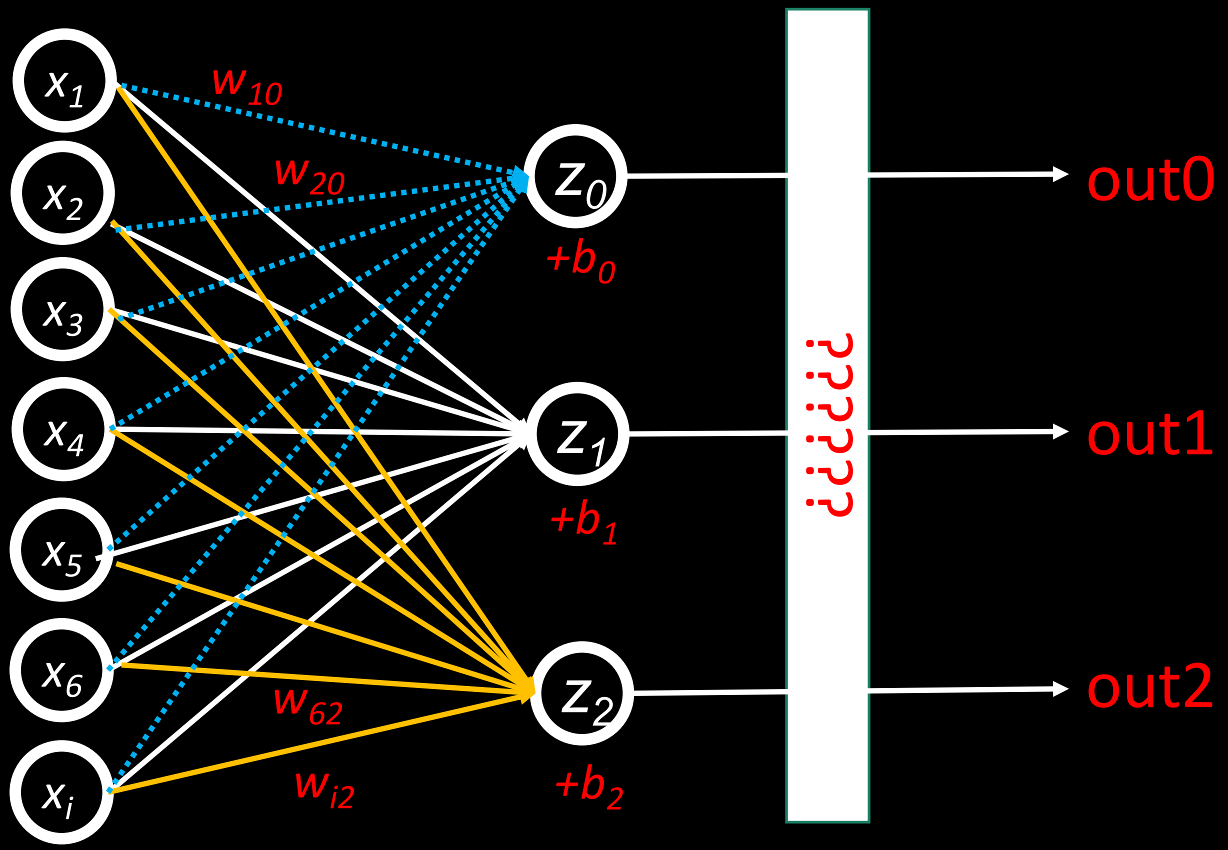
Single Class Logistic Regression

$$p = \frac{1}{1+exp-(\Sigma_i w_i x_i + b)} = \text{Sigmoid}(\Sigma_i w_i x_i + b) = S(WX)$$

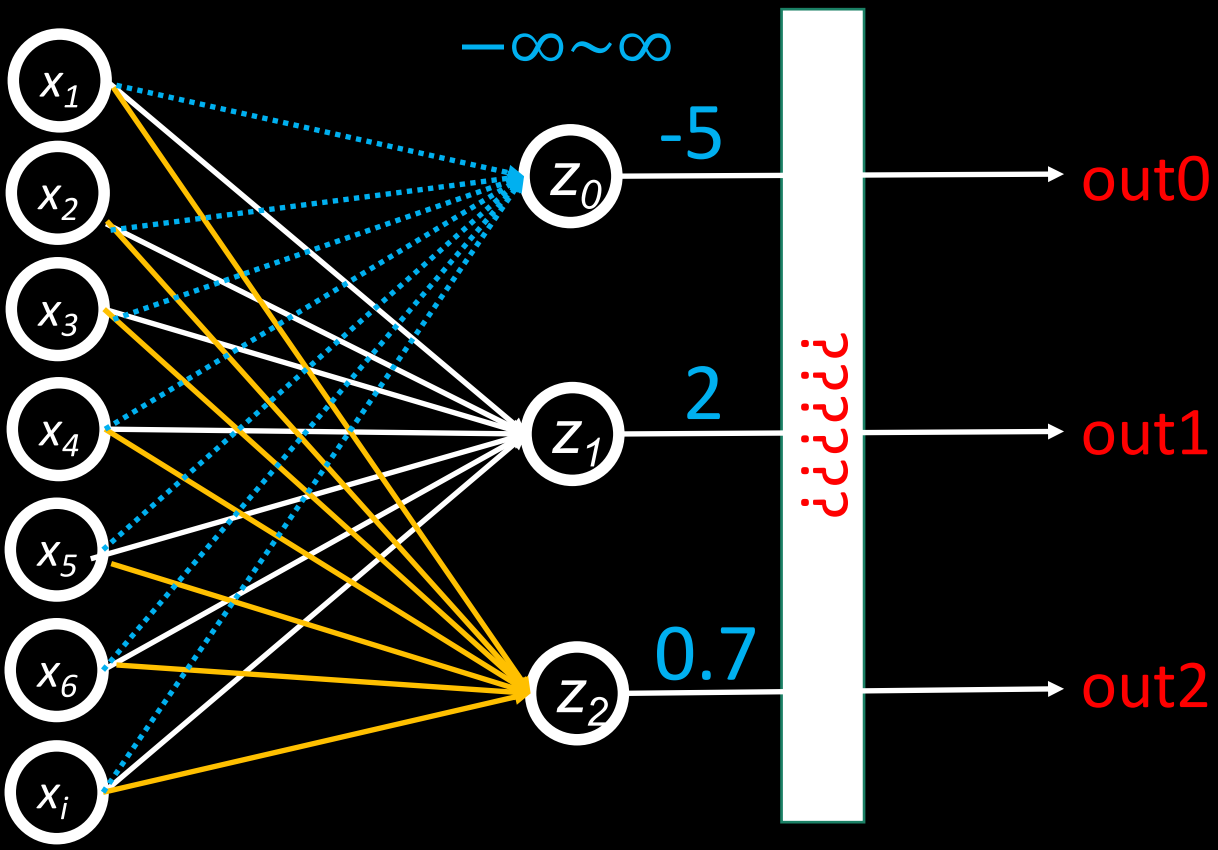




out0 = $w_{10}x_1 + w_{20}x_2 + w_{30}x_3 + ...$
x_1
x_2
x_3
x_4
x_5
x_6
x_i
w_{10}
w_{20}
z_0
$+b_0$
z_1
$+b_1$
z_2
$+b_2$
¿¿¿¿¿¿
out0
out1
out2



$out_j =$
$w_{1j}x_1 + w_{2j}x_2 + w_{3j}x_3 + ...$

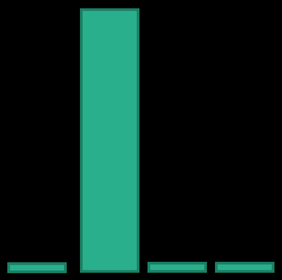


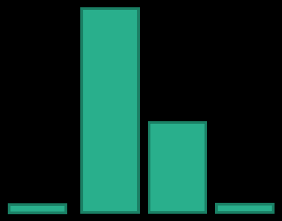
Probability
0~1
&
(out0+out1+out2+.....=1)

$$p(y=N) = Softmax(W_N X) = \frac{exp-(W_N X)}{\sum_n exp-(W_n X)}$$

Softmax ==
Sigmoid for multi-class

Data	Output (Wx)	exp(Wx)	argmax	Softmax exp(Wx) / sum(exp(Wx)))
3	-0.84	0.43	0	0.02
4	2.69	14.73	1	0.68
5	1.76	5.81	0	0.27
6	-0.69	0.50	0	0.02





$p = \frac{1}{1+exp-(\Sigma_i w_i x_i + b)}$ = Sigmoid$(w_i x_i + b)$ = S($\mathrm{W}X$)

V S

$$p(y=0) = SX(W_0X) = \frac{exp-(W_0X)}{\sum_n exp-(W_nX)}$$

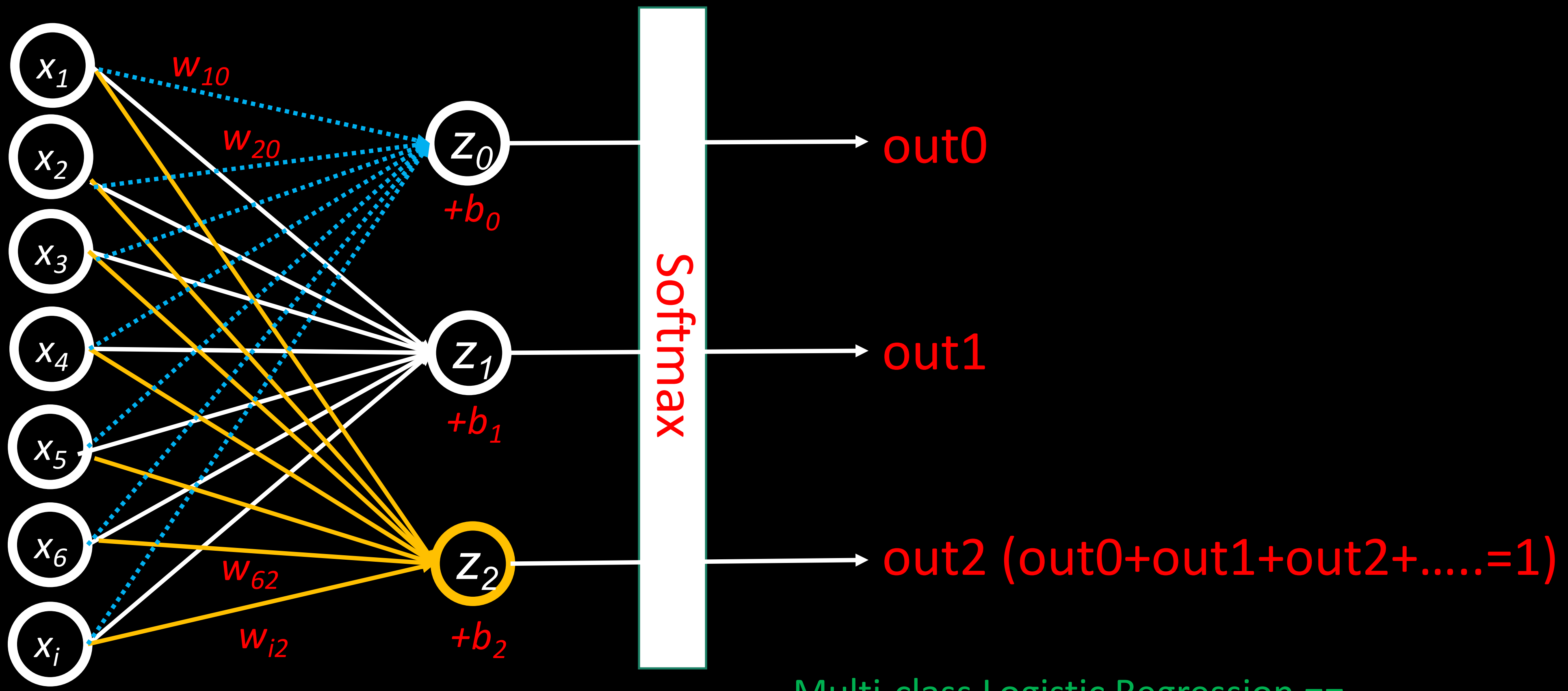
$$p(y=1) = SX(W_1X) = \frac{exp-(W_1X)}{\sum_n exp-(W_nX)}$$

$$\vdots$$

$$p(y=N) = SX(W_NX) = \frac{exp-(W_NX)}{\sum_n exp-(W_nX)}$$

For multi-classes, you will have one *W* for each class

The summation for all the P Should be 1



Multi-class Logistic Regression ==

Linear regression between probability and Softmax(WX)

Binary Cross Entropy

$$-\sum_{n}[\mathrm{y} \cdot \log(p(\hat{y})) \; + \; (1-y) \cdot \log(p(1-\hat{y}))])$$

Multi-Class Cross Entropy

$$-\sum_{n}\mathrm{y} \cdot \log(p(\hat{y}))$$

y	$\hat{y}$		
0	0.6		
1	0.3	=	-1 * log(0.3)
0	0.1		

Image Classification by Logistic Regression

MNIST dataset

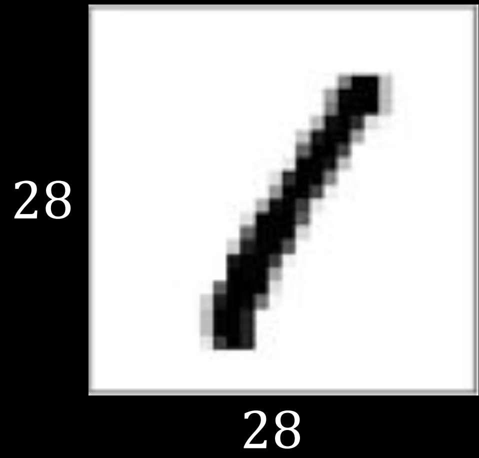
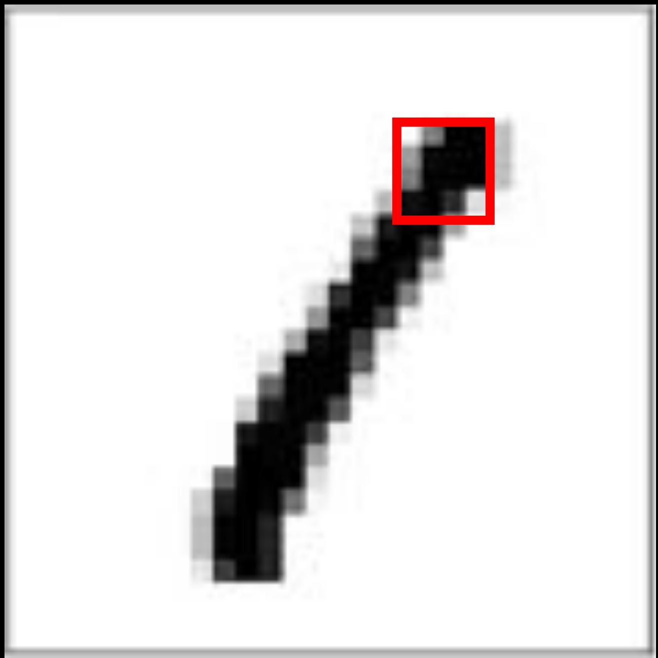
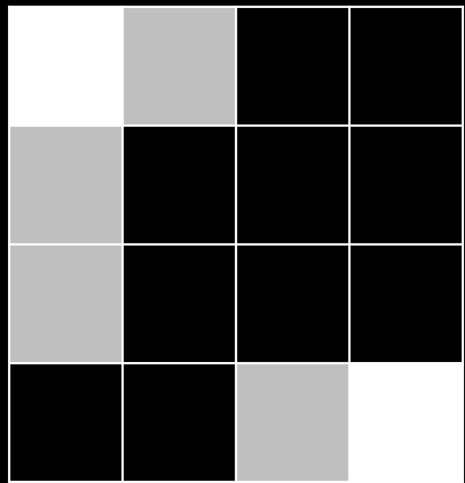


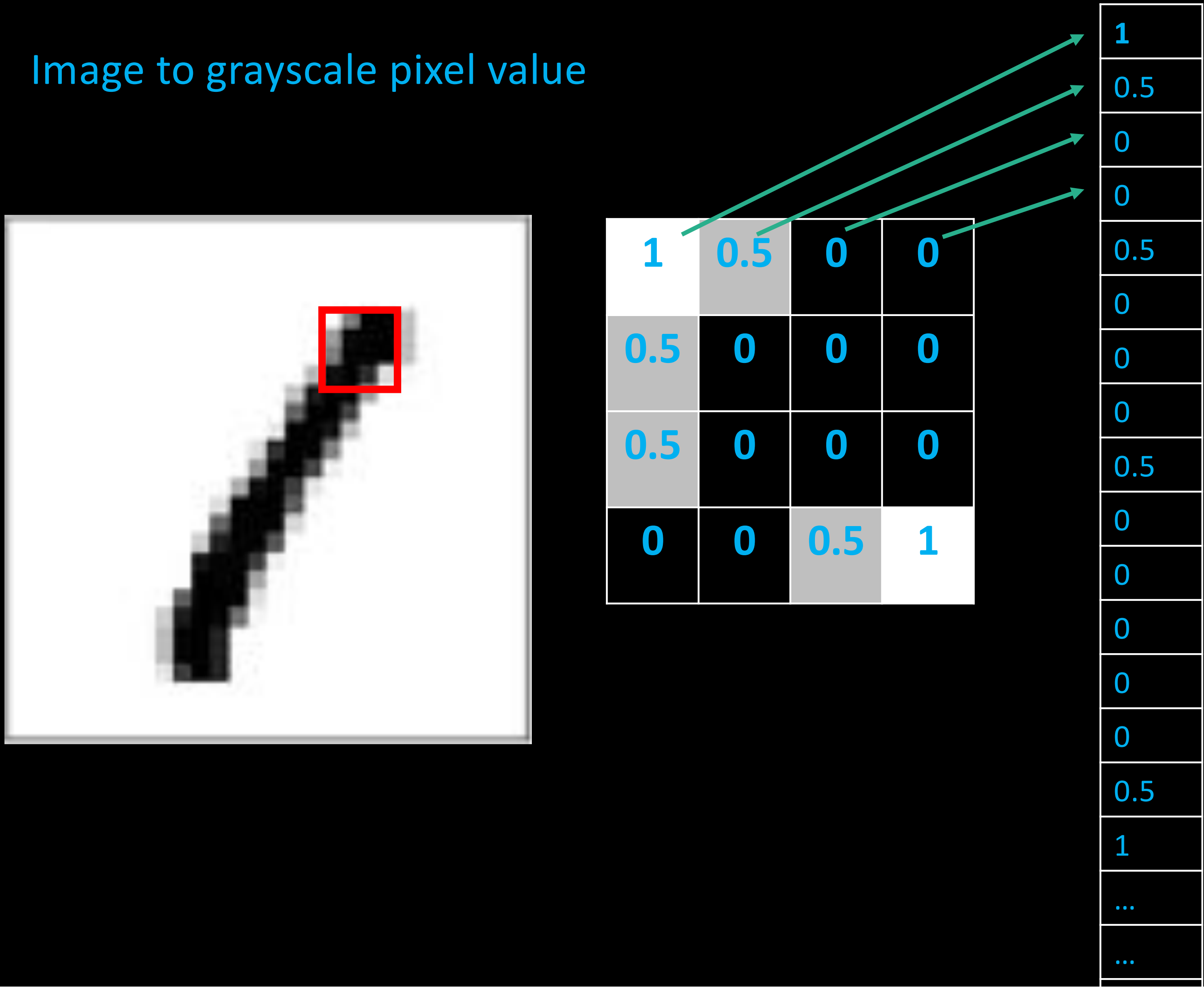
Image to grayscale pixel value





1	0.5	0	0
0.5	0	0	0
0.5	0	0	0
0	0	0.5	1

Image to grayscale pixel value



numpy.ndarray.flatten

torch.array.view(-1)

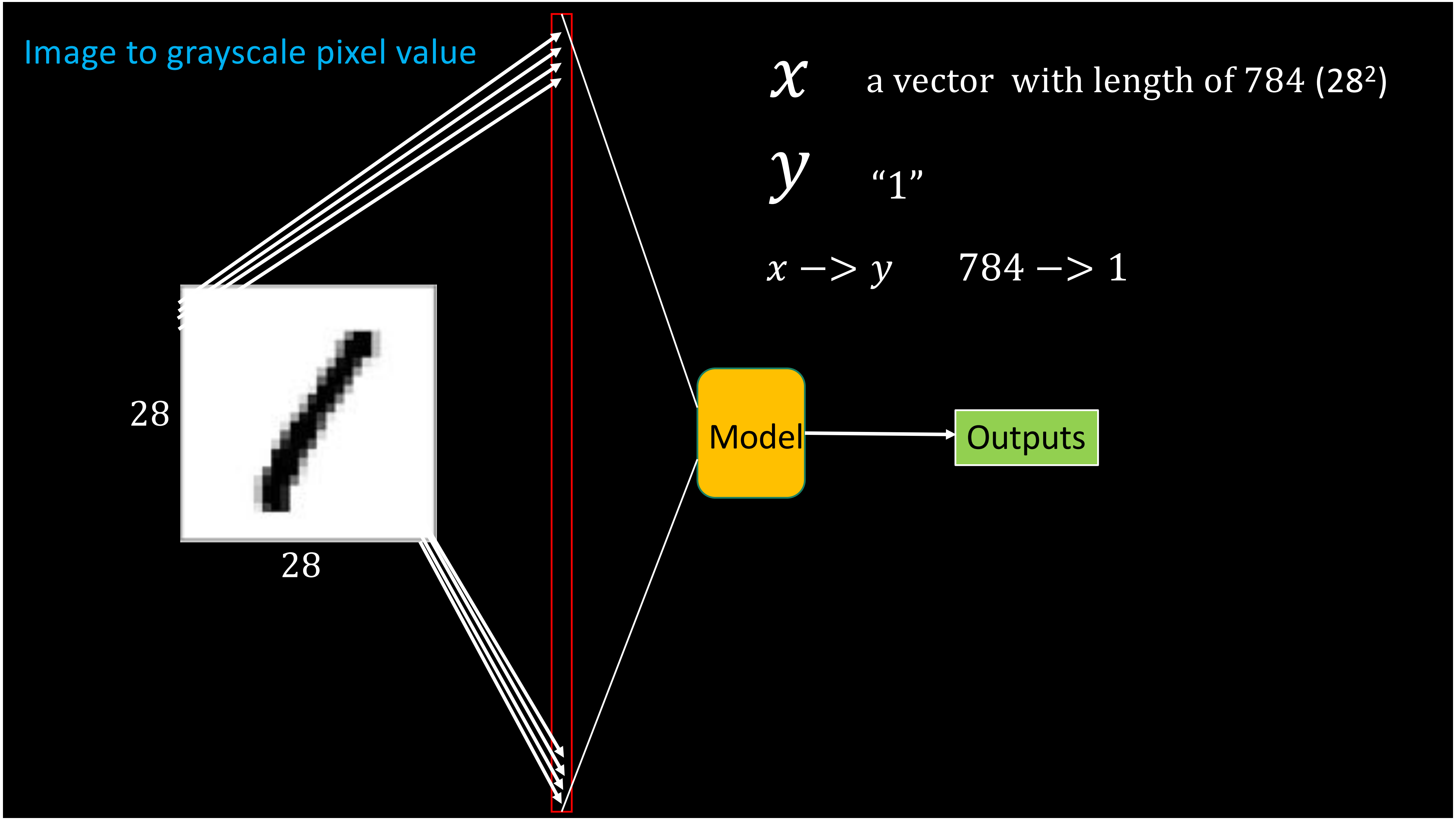


Image to grayscale pixel value
x a vector with length of 784 (28²)
y "1"
x −> y 784 −> 1
28
28
Model
Outputs