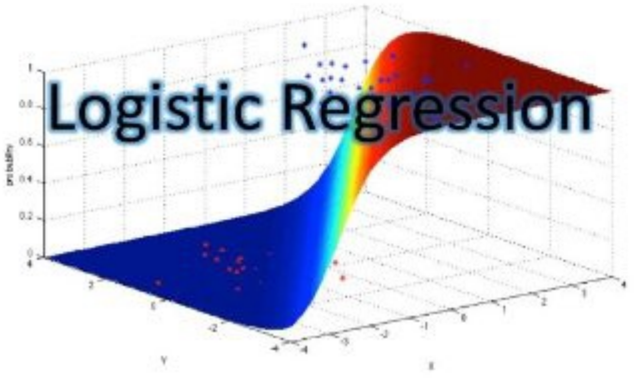


Logistic Regression

~Abhishek Kumar

Logistic Regression

- Supervised
- It is not a regression but a Classfication algorithm
- Example:
  - Spam/Not spam,
  - Tumor: Malignant/Benign

E-mail spam classifier

Spam

Non-spam (ham)

E-mail spam classifier

Spam

Buy, l0ts of money, now, che@p buy buy free mon3y

Non-spam (ham)

Hello grandson, I made cookies. Love, Grandma

E-mail spam classifier

Spam

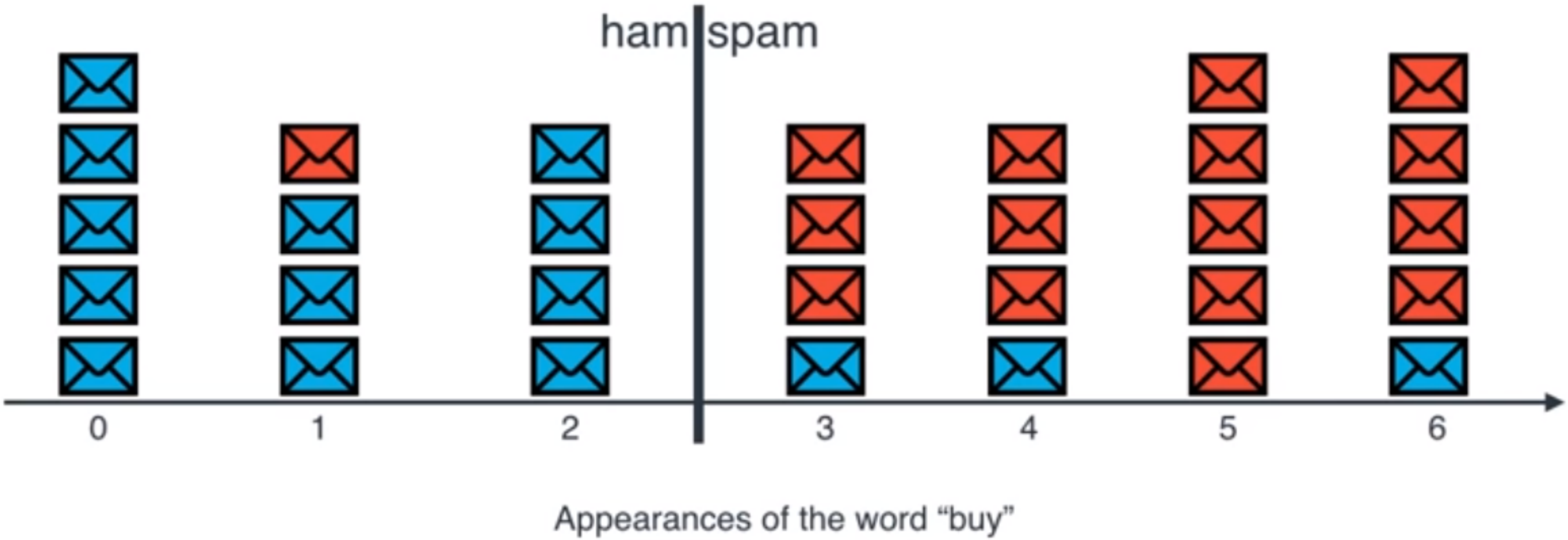
Buy, l0ts of money,
now, che@p buy
buy free mon3y

Non-spam (ham)

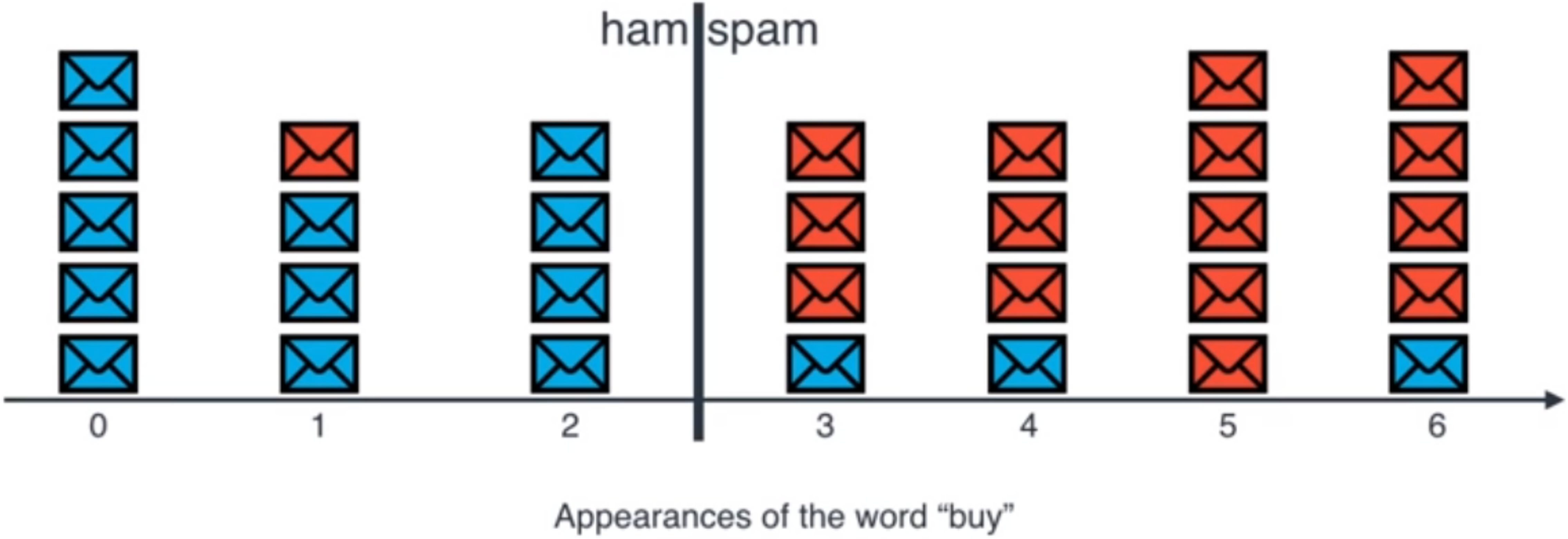
Hello grandson,
I made cookies.
Love, Grandma

'buy'
spelling mistakes


0
1
2
3
4
5
6
Appearances of the word "buy"

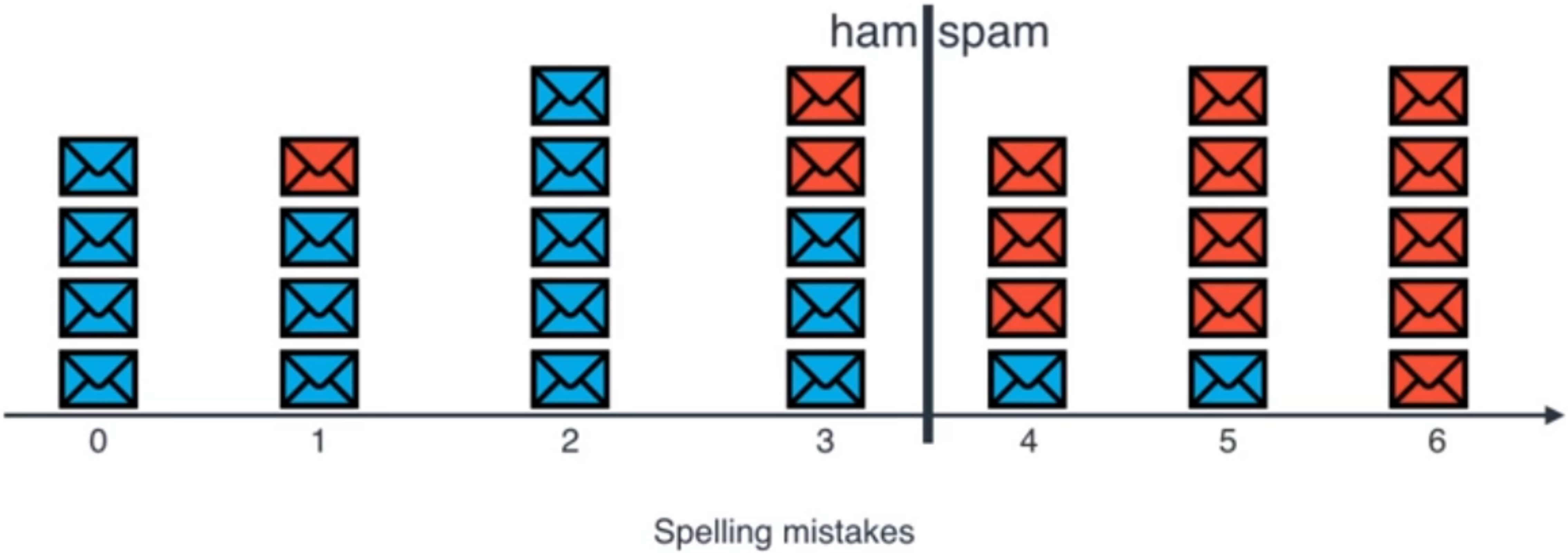

ham
spam
0
1
2
3
4
5
6
Appearances of the word “buy”

Rule 1: If #appearances of the word 'buy' > 2, then spam




0
1
2
3
4
5
6
Spelling mistakes

Rule 2: If #spelling mistakes > 3, then spam



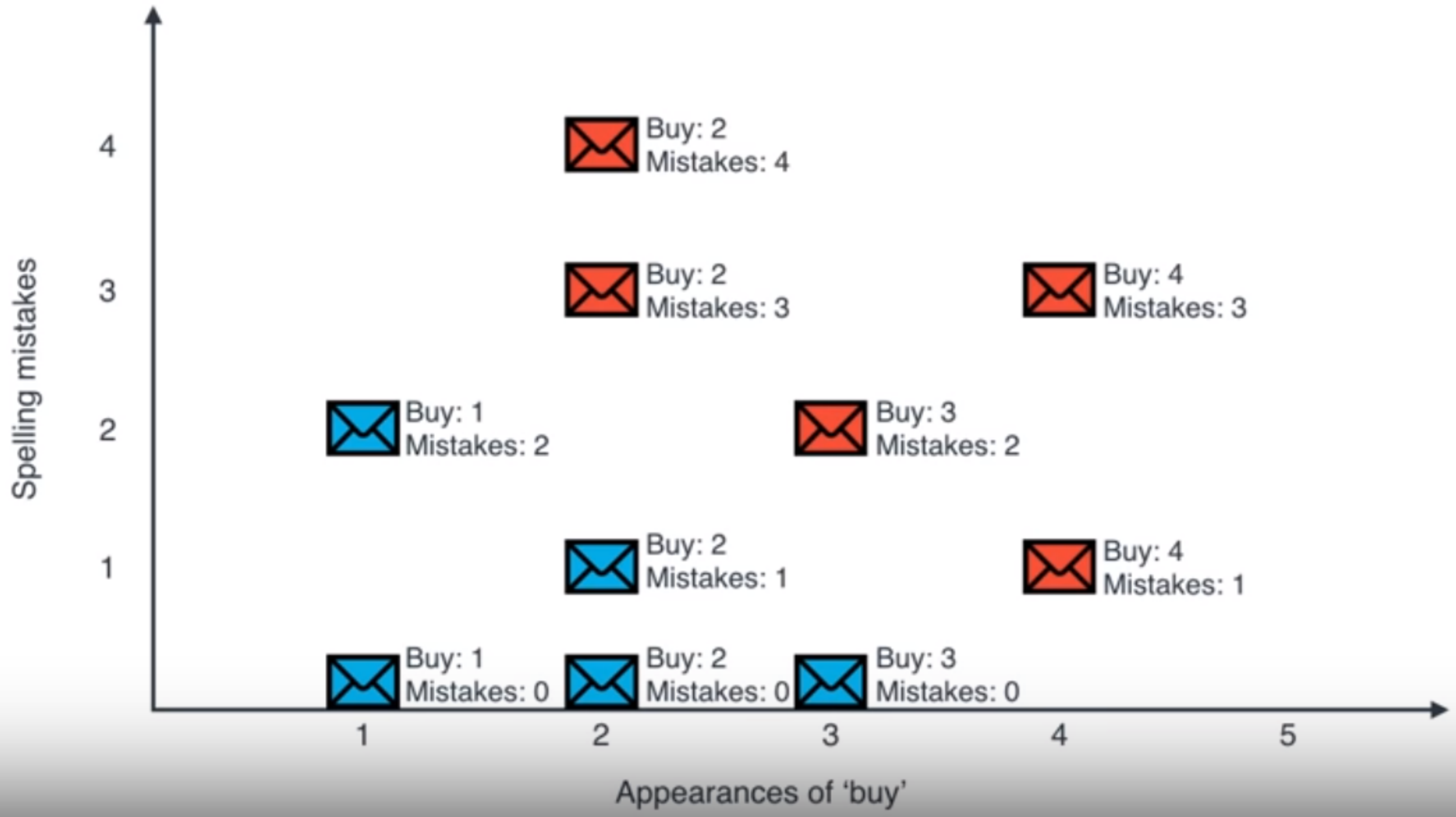
E-mails

- Buy: **4**, Mistakes: **3**
- Buy: **1**, Mistakes: **2**
- Buy: **0**, Mistakes: **2**
- Buy: **2**, Mistakes: **1**
- Buy: **4**, Mistakes: **1**
- Buy: **0**, Mistakes: **3**
- Buy: **2**, Mistakes: **3**
- Buy: **0**, Mistakes: **1**
- Buy: **2**, Mistakes: **4**
- Buy: **3**, Mistakes: **2**

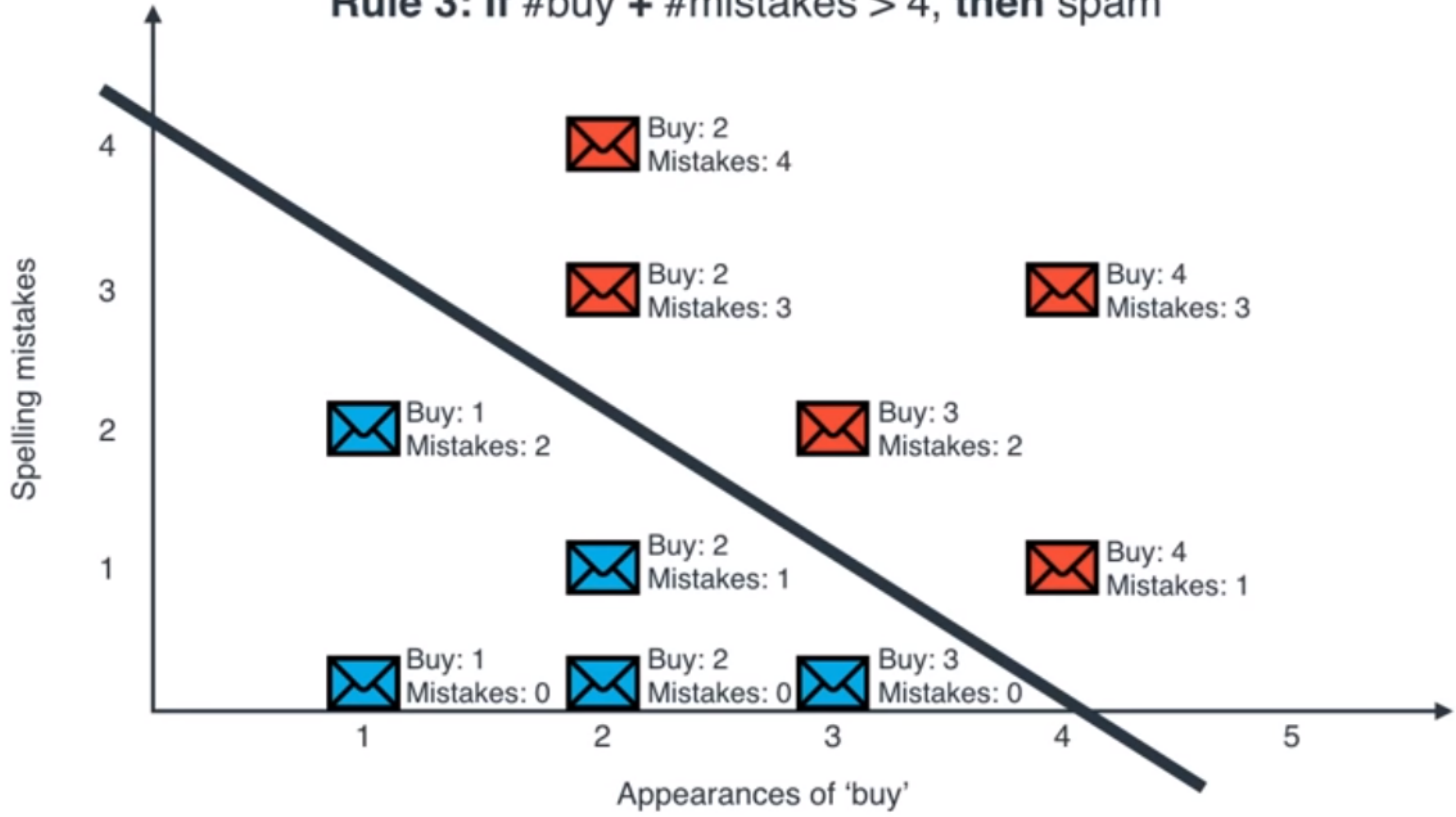
E-mails	Buy + Mistakes
Buy: **4**, Mistakes: **3**	7
Buy: **1**, Mistakes: **2**	3
Buy: **0**, Mistakes: **2**	2
Buy: **2**, Mistakes: **1**	3
Buy: **4**, Mistakes: **1**	5
Buy: **0**, Mistakes: **3**	3
Buy: **2**, Mistakes: **3**	5
Buy: **0**, Mistakes: **1**	1
Buy: **2**, Mistakes: **4**	6
Buy: **3**, Mistakes: **2**	5

E-mails	Buy + Mistakes
Buy: **4**, Mistakes: **3**	7
Buy: **1**, Mistakes: **2**	3
Buy: **0**, Mistakes: **2**	2
Buy: **2**, Mistakes: **1**	3
Buy: **4**, Mistakes: **1**	5
Buy: **0**, Mistakes: **3**	3
Buy: **2**, Mistakes: **3**	5
Buy: **0**, Mistakes: **1**	1
Buy: **2**, Mistakes: **4**	6
Buy: **3**, Mistakes: **2**	5

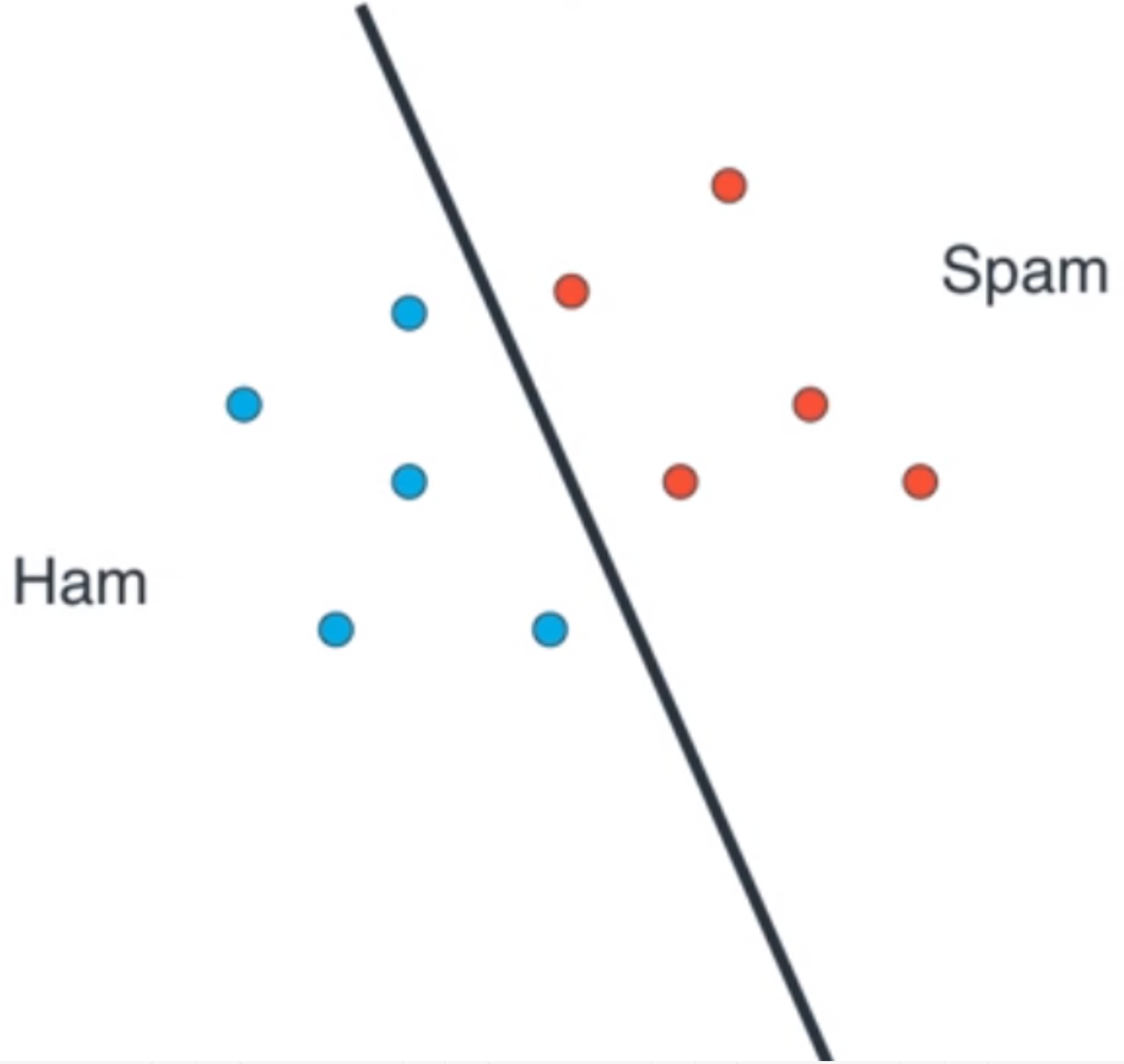
Rule 3: If #buy **+** #mistakes > 4, **then** spam


Spelling mistakes
4
3
2
1
Buy: 2
Mistakes: 4
Buy: 2
Mistakes: 3
Buy: 4
Mistakes: 3
Buy: 1
Mistakes: 2
Buy: 3
Mistakes: 2
Buy: 2
Mistakes: 1
Buy: 4
Mistakes: 1
Buy: 1
Mistakes: 0
Buy: 2
Mistakes: 0
Buy: 3
Mistakes: 0
1
2
3
4
5
Appearances of 'buy'

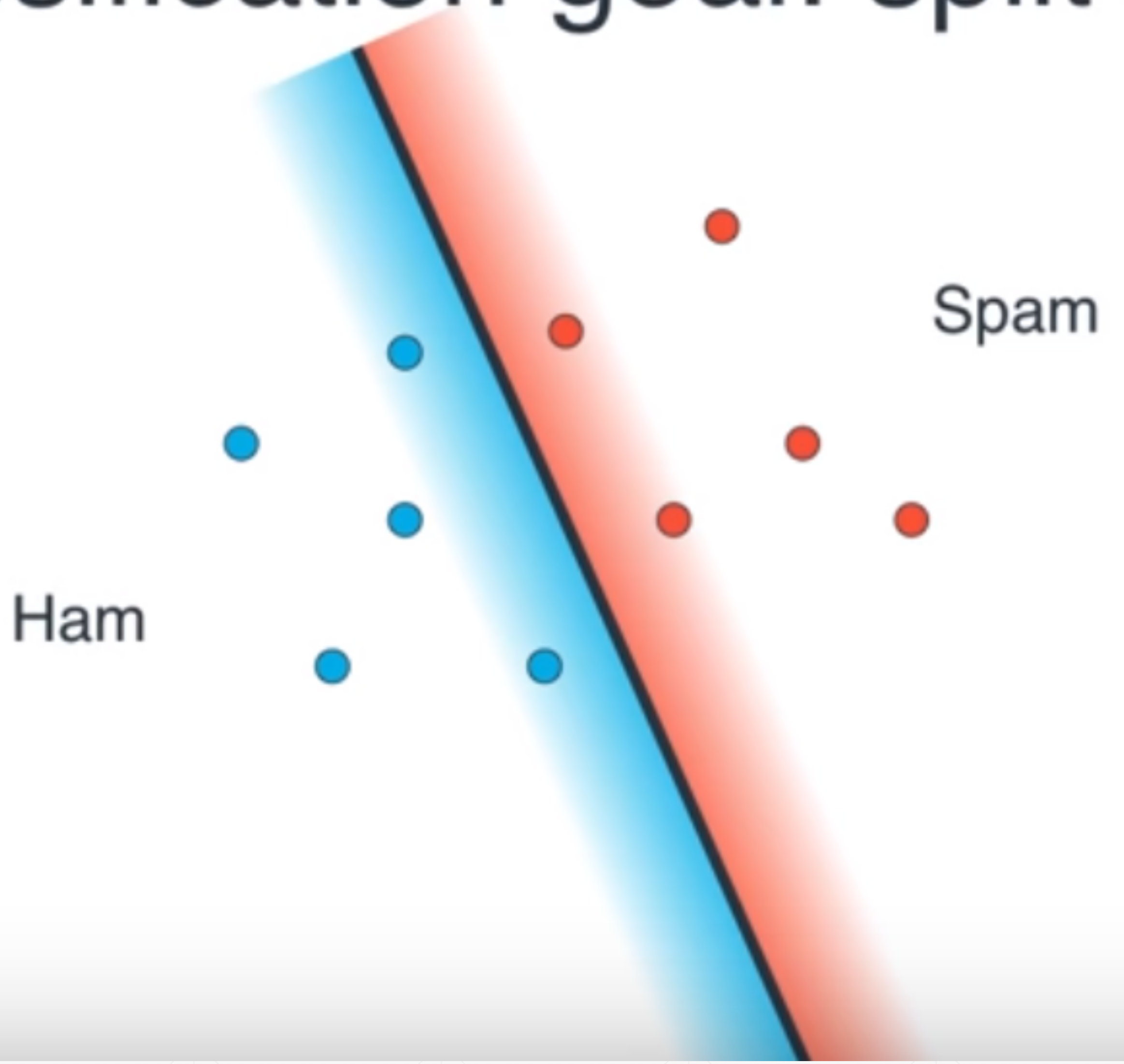
Rule 3: If #buy **+** #mistakes > 4, **then** spam

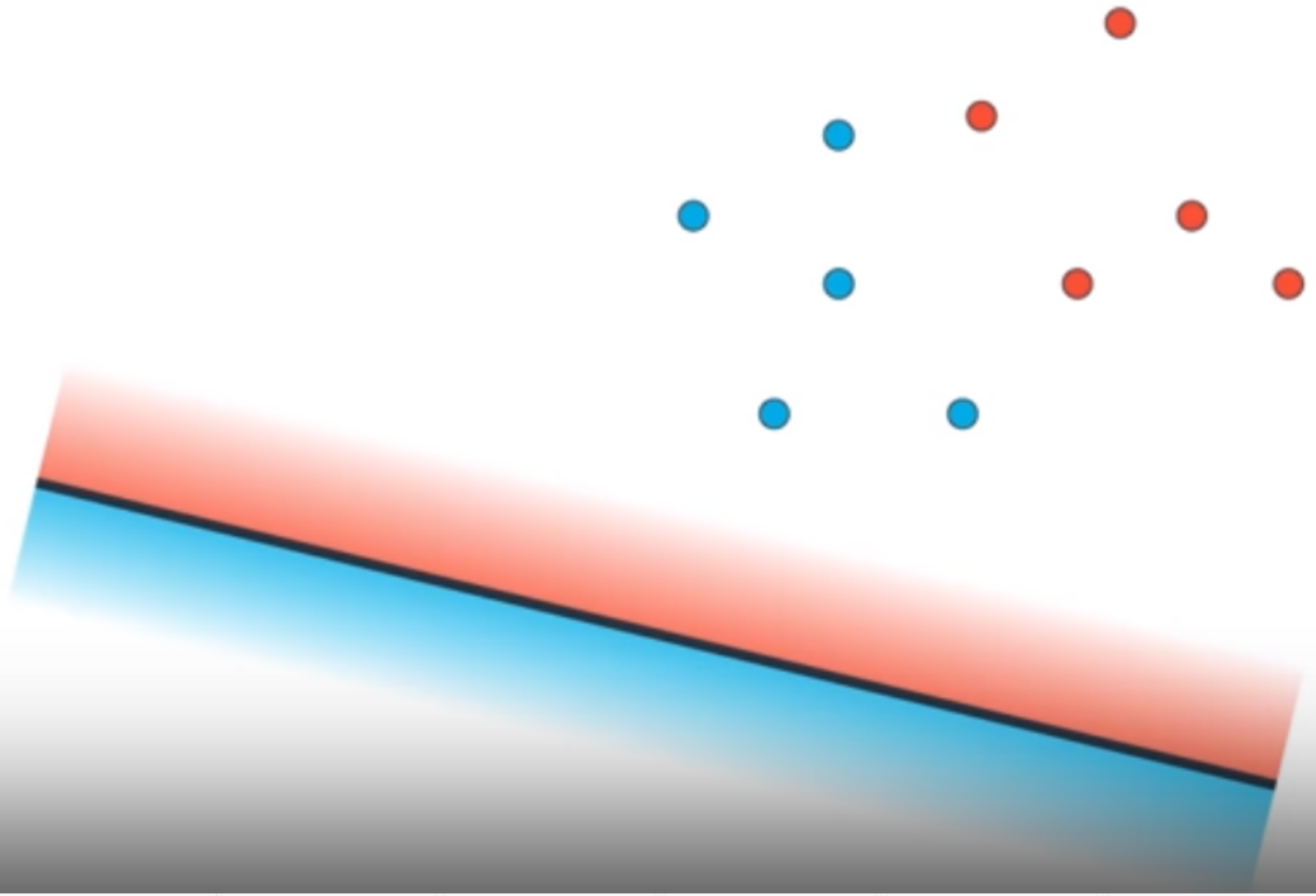


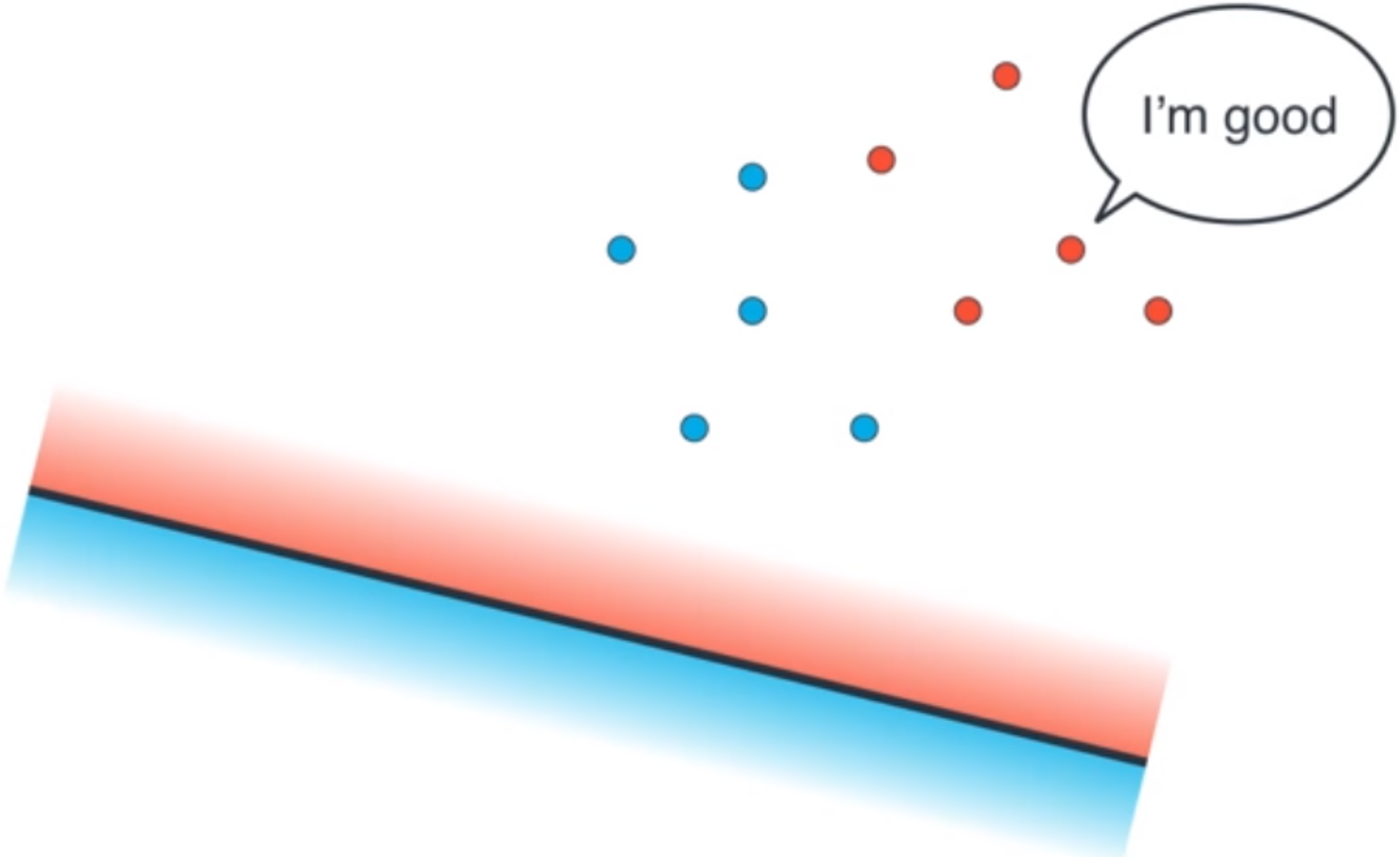
Classification goal: split data

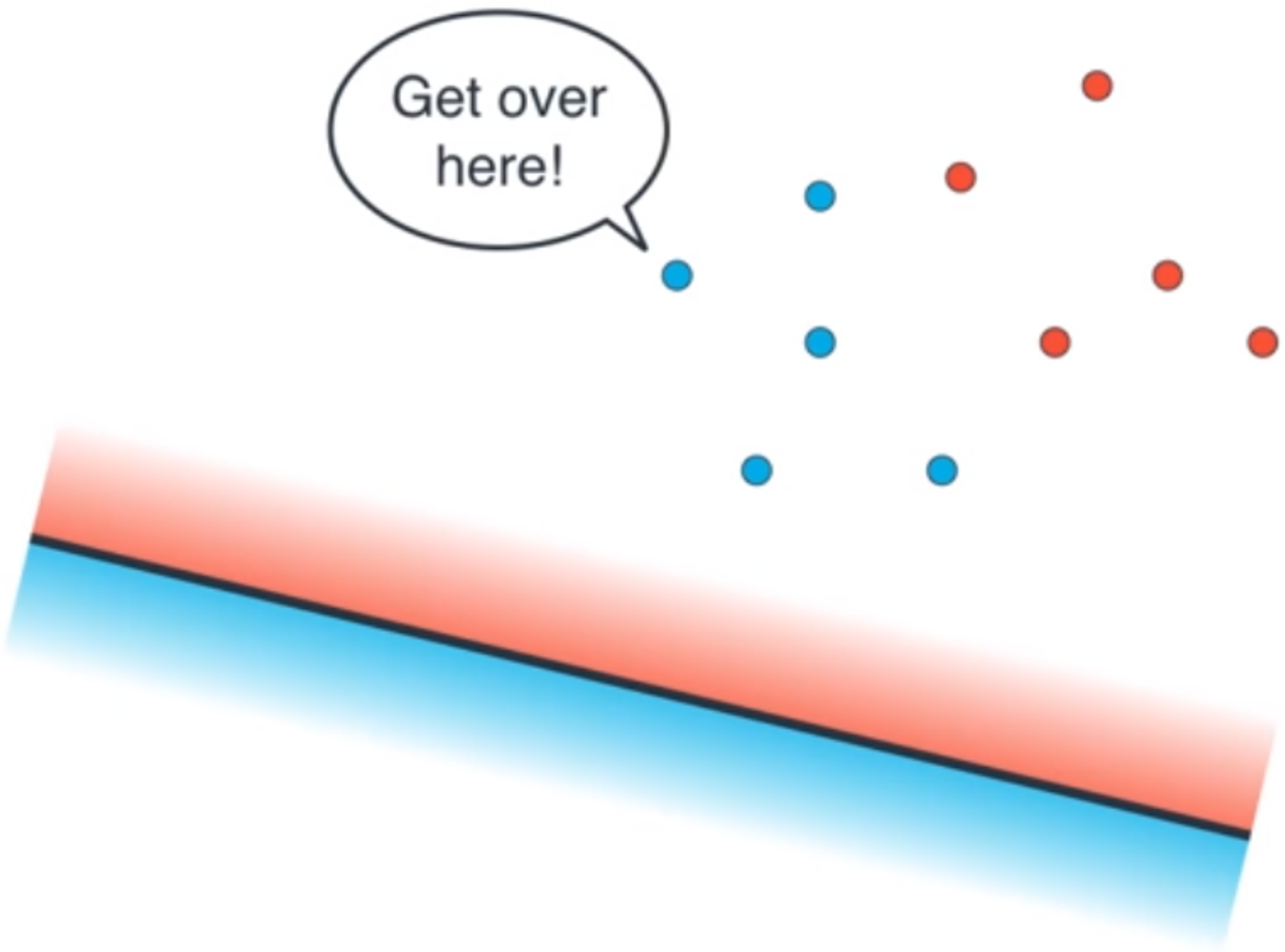


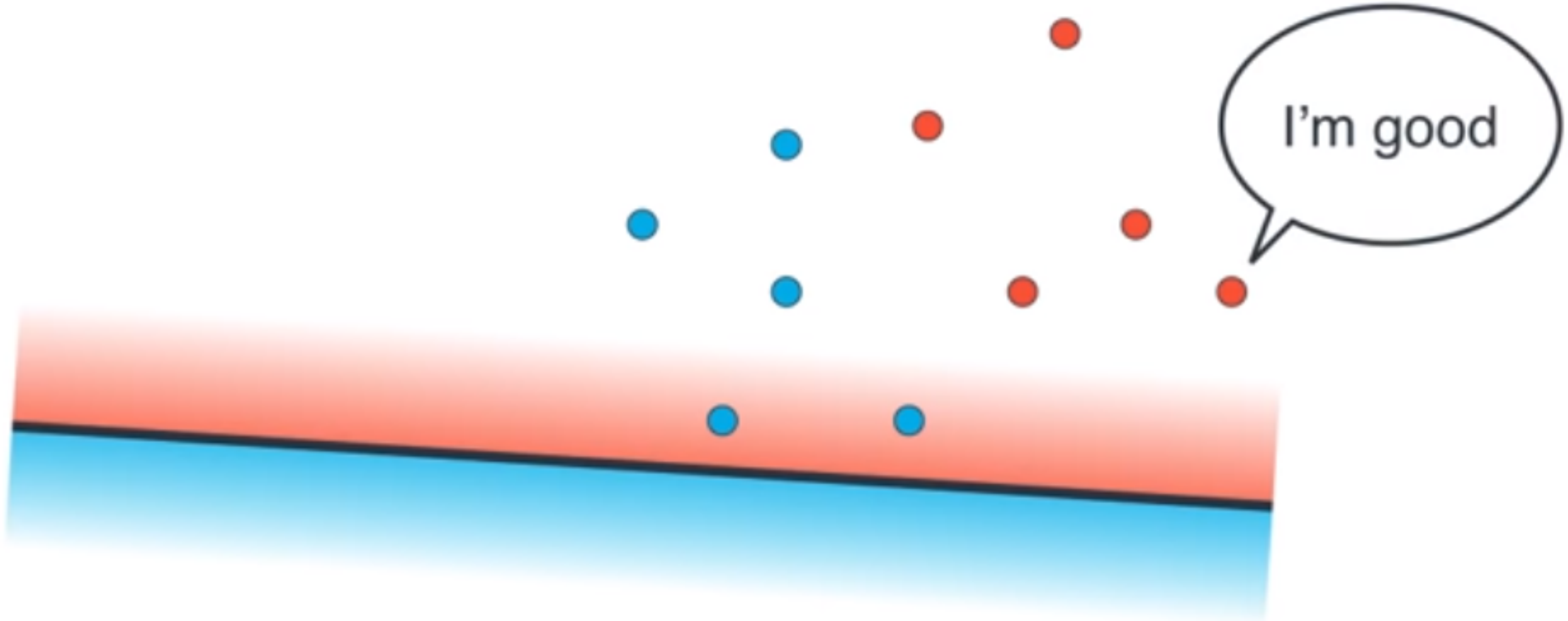
Classification goal: split data

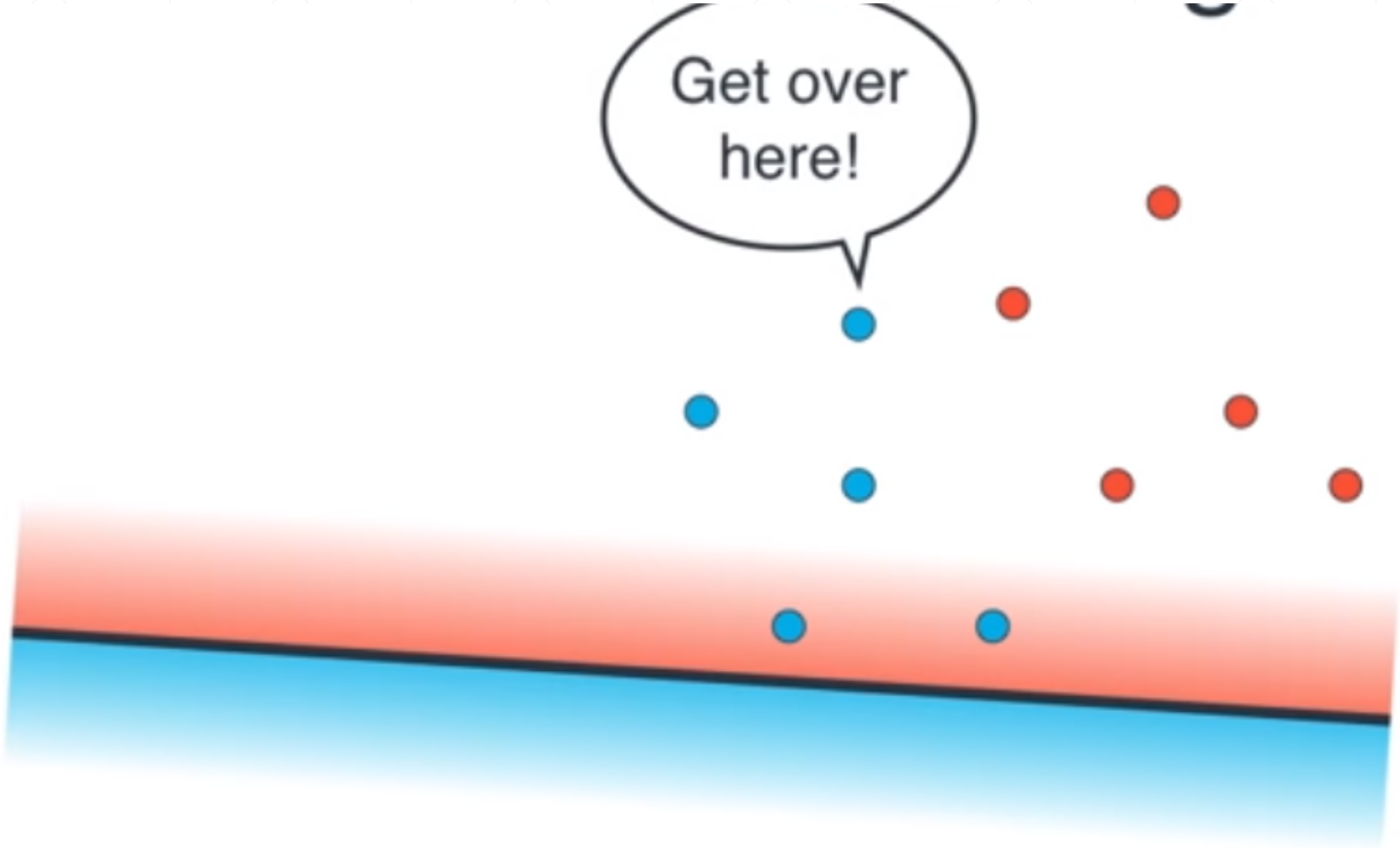


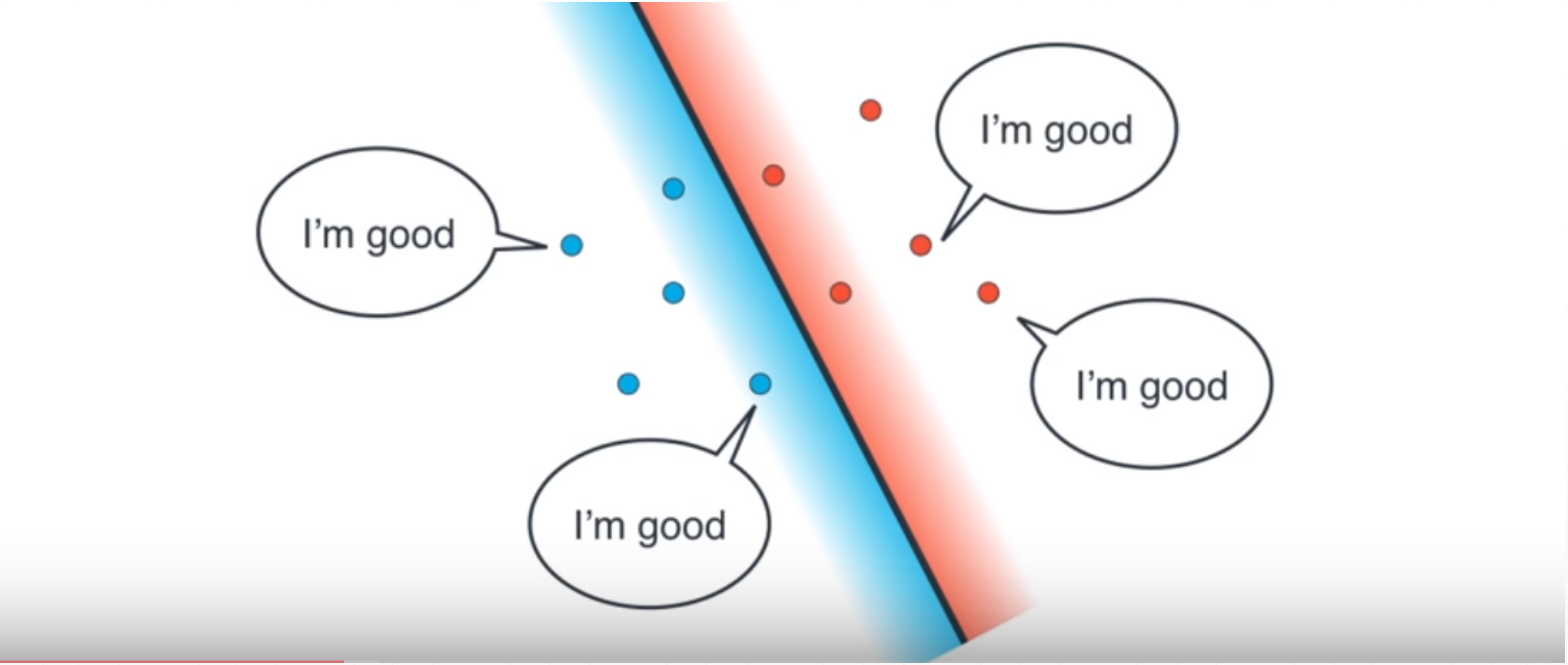



I’m good

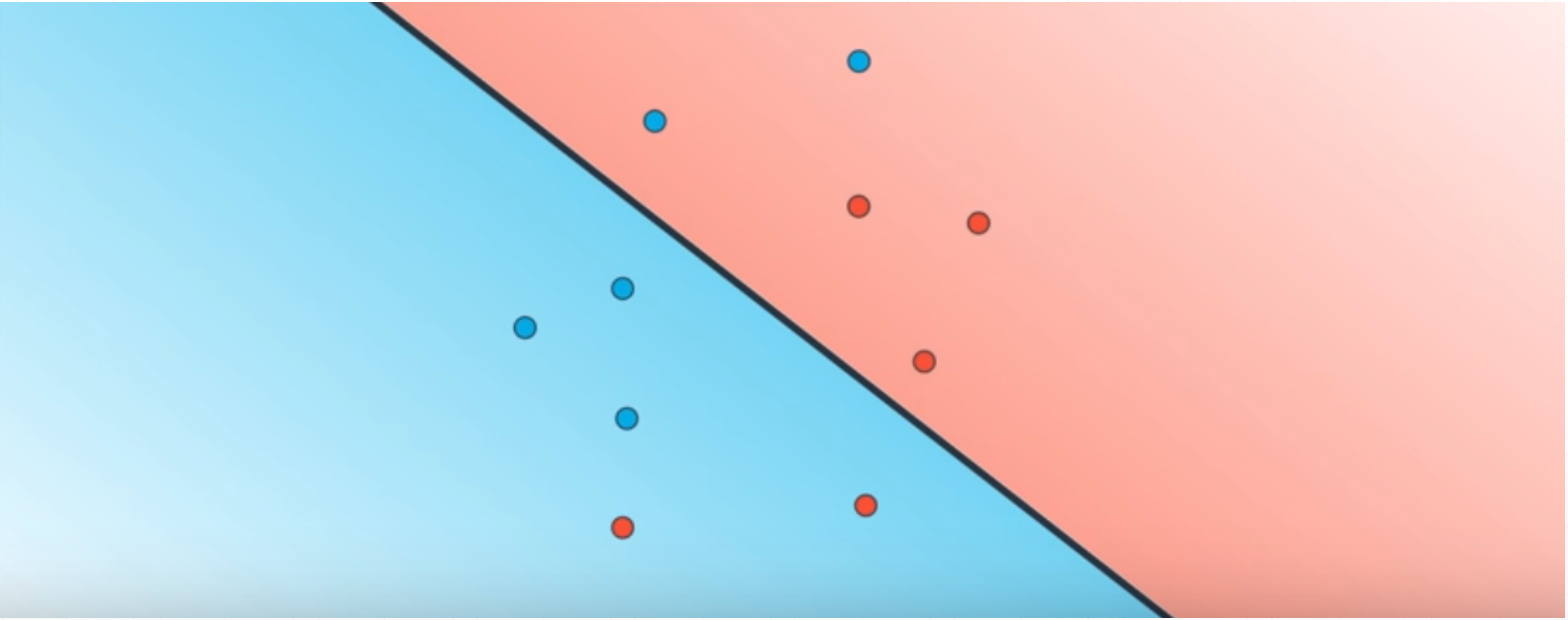

Get over
here!

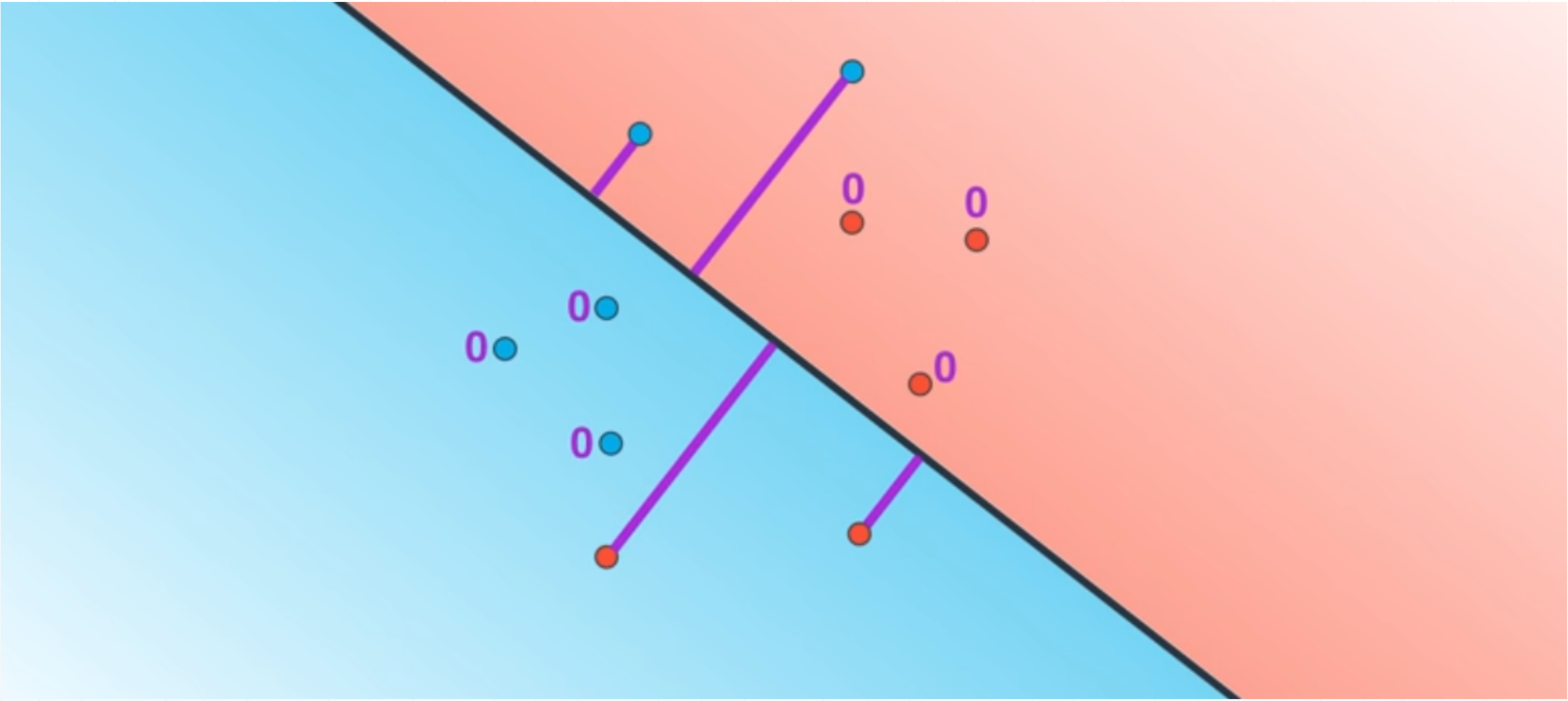

I’m good

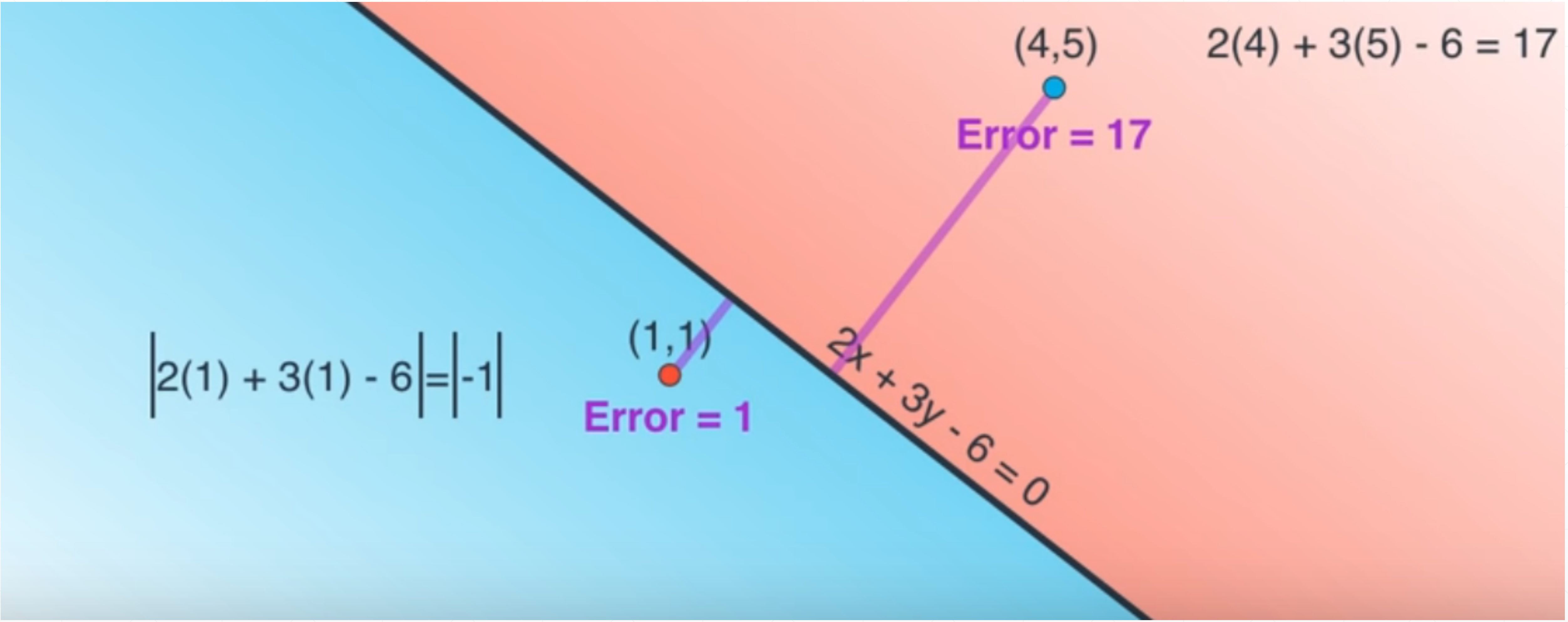

Get over here!



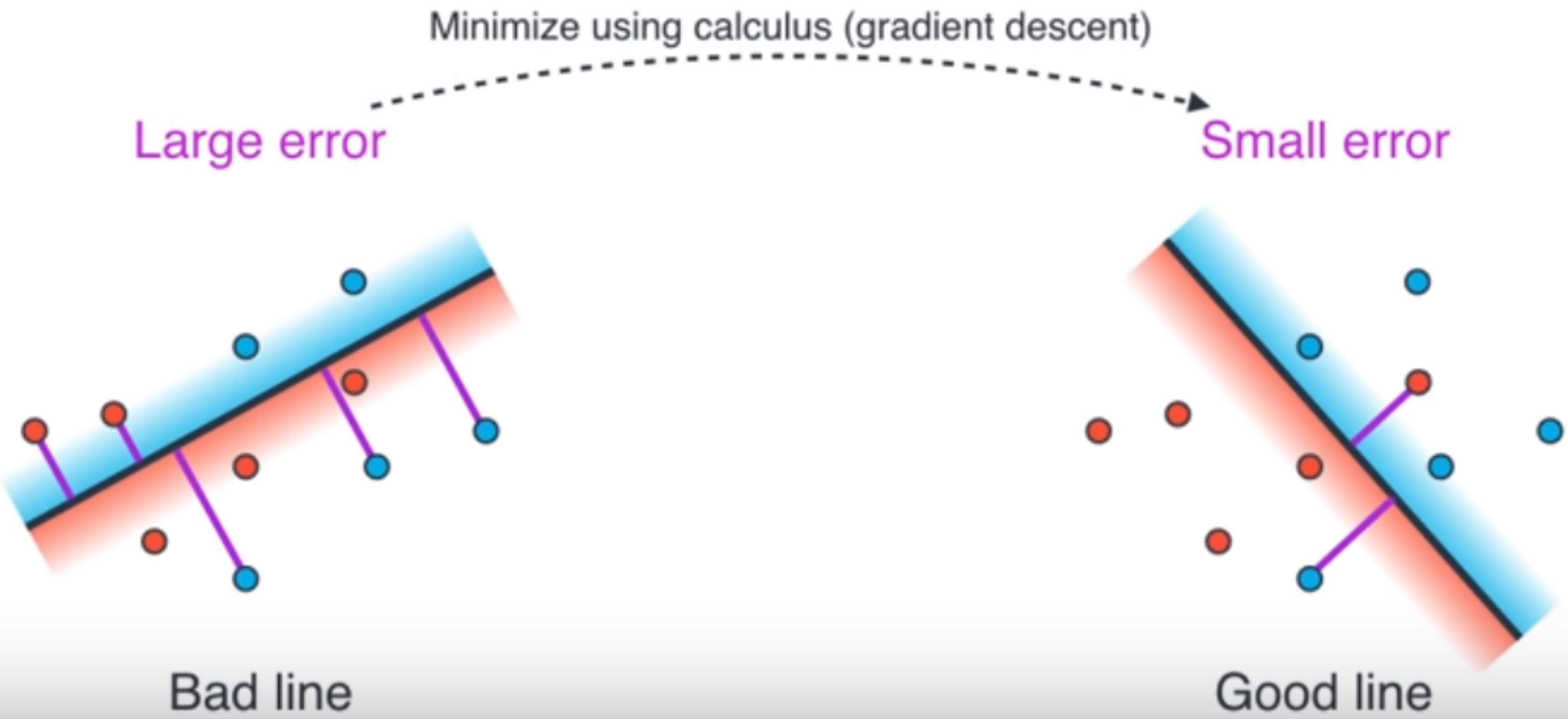
I'm good
I'm good
I'm good
I'm good



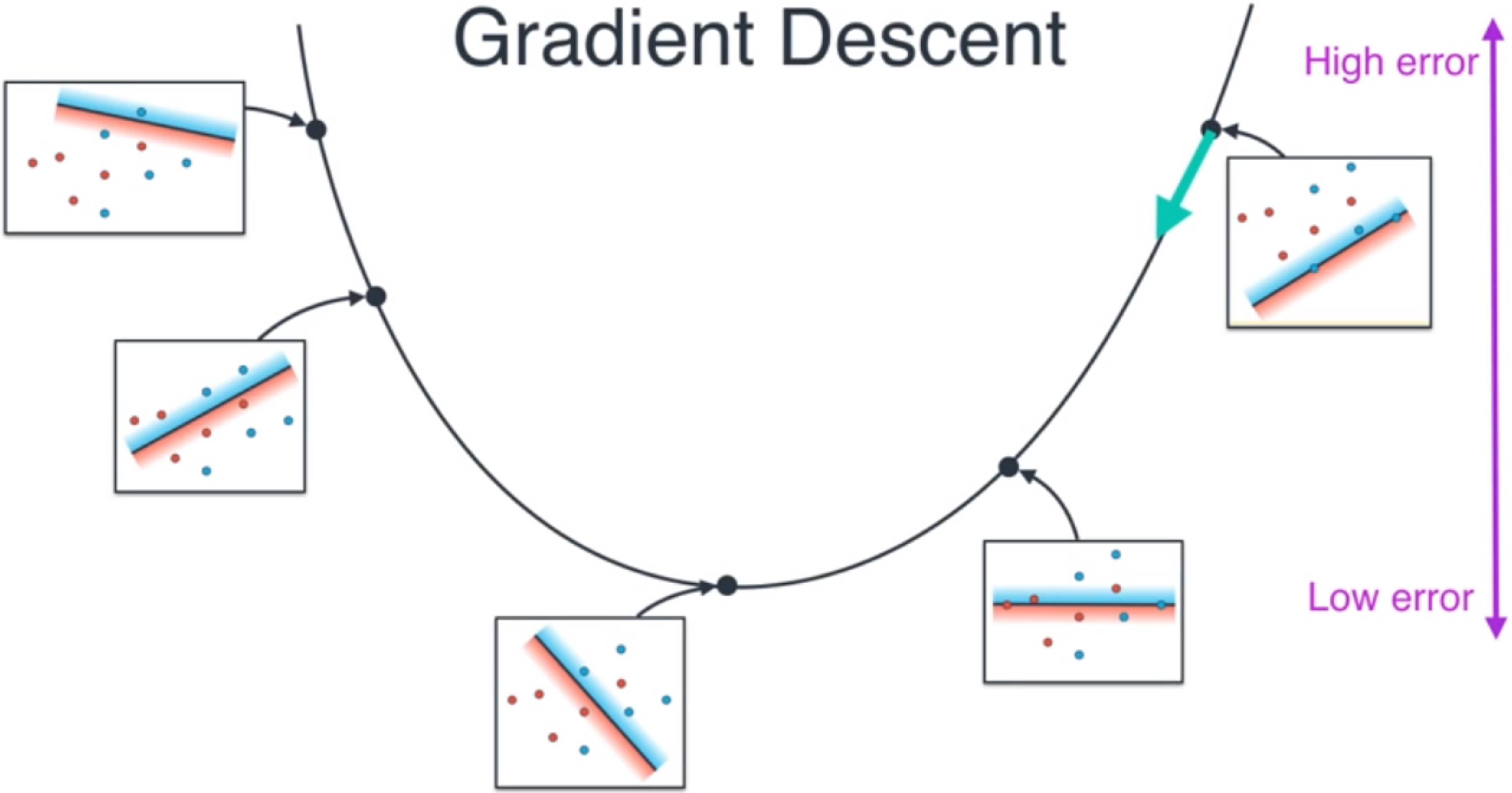

0
0
0
0
0
0

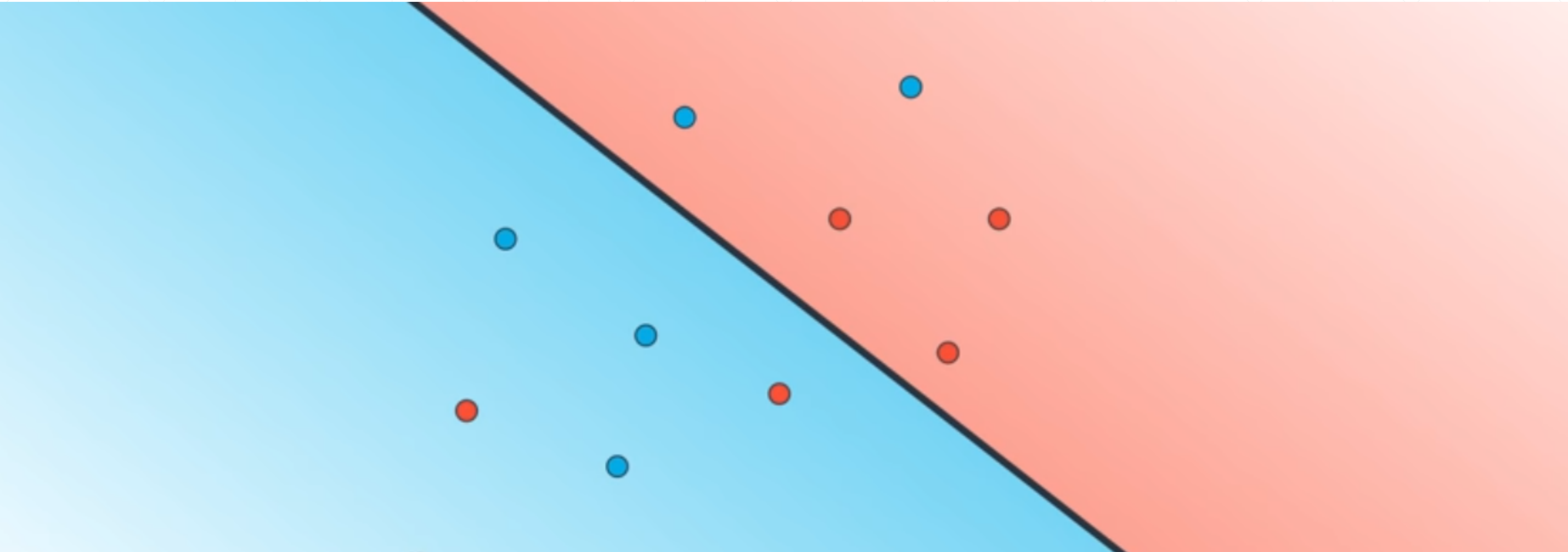

(4,5)
2(4) + 3(5) - 6 = 17
Error = 17
(1,1)
|2(1) + 3(1) - 6|=|-1|
Error = 1
2x + 3y - 6 = 0

Gradient Descent

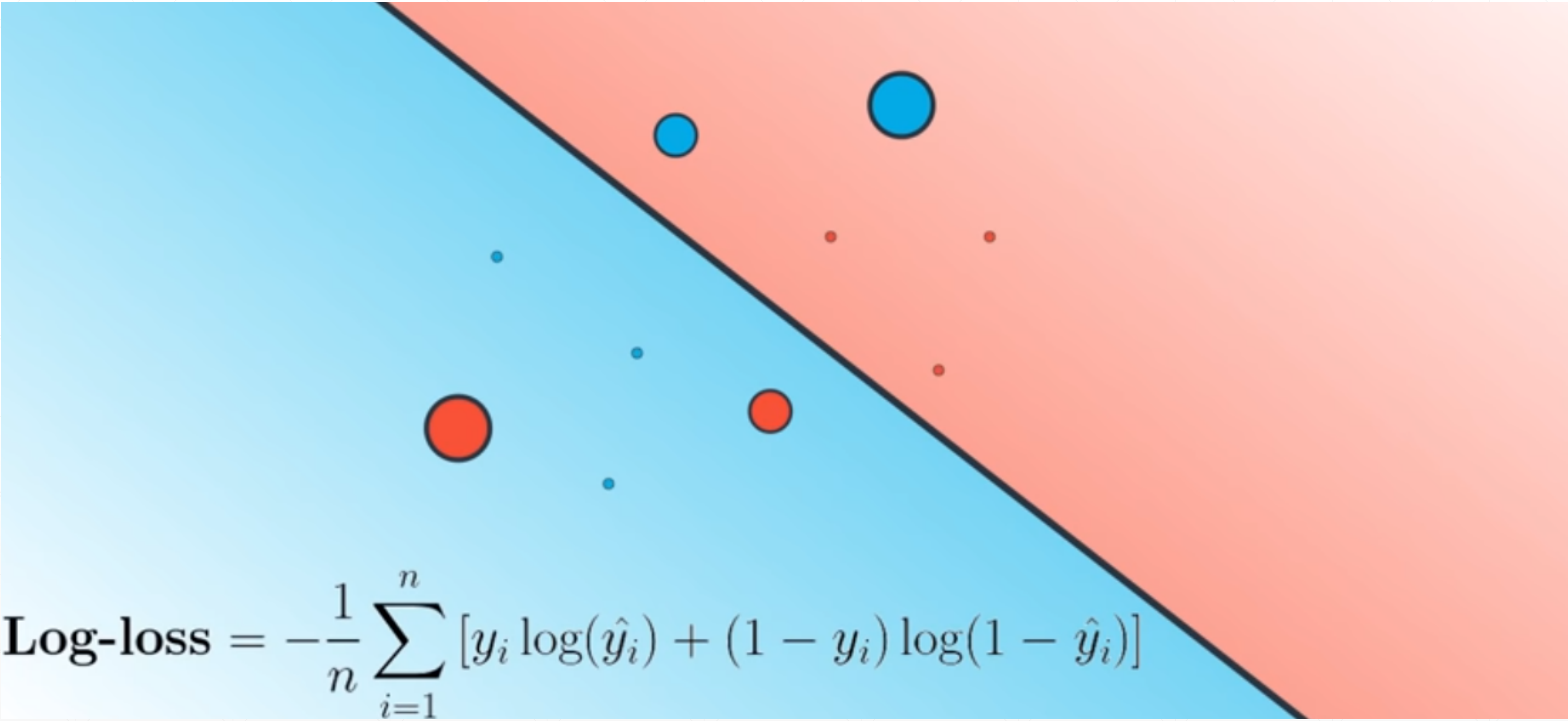


Gradient Descent

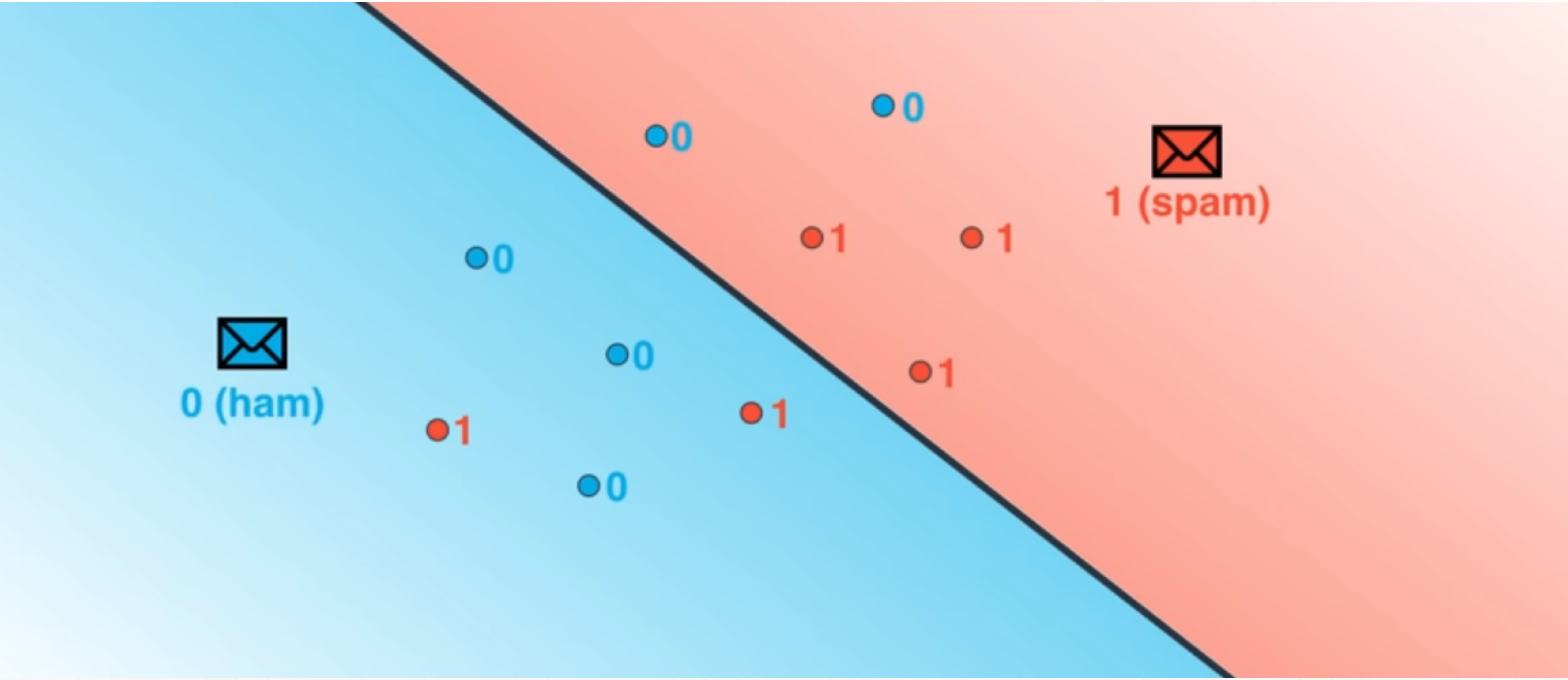




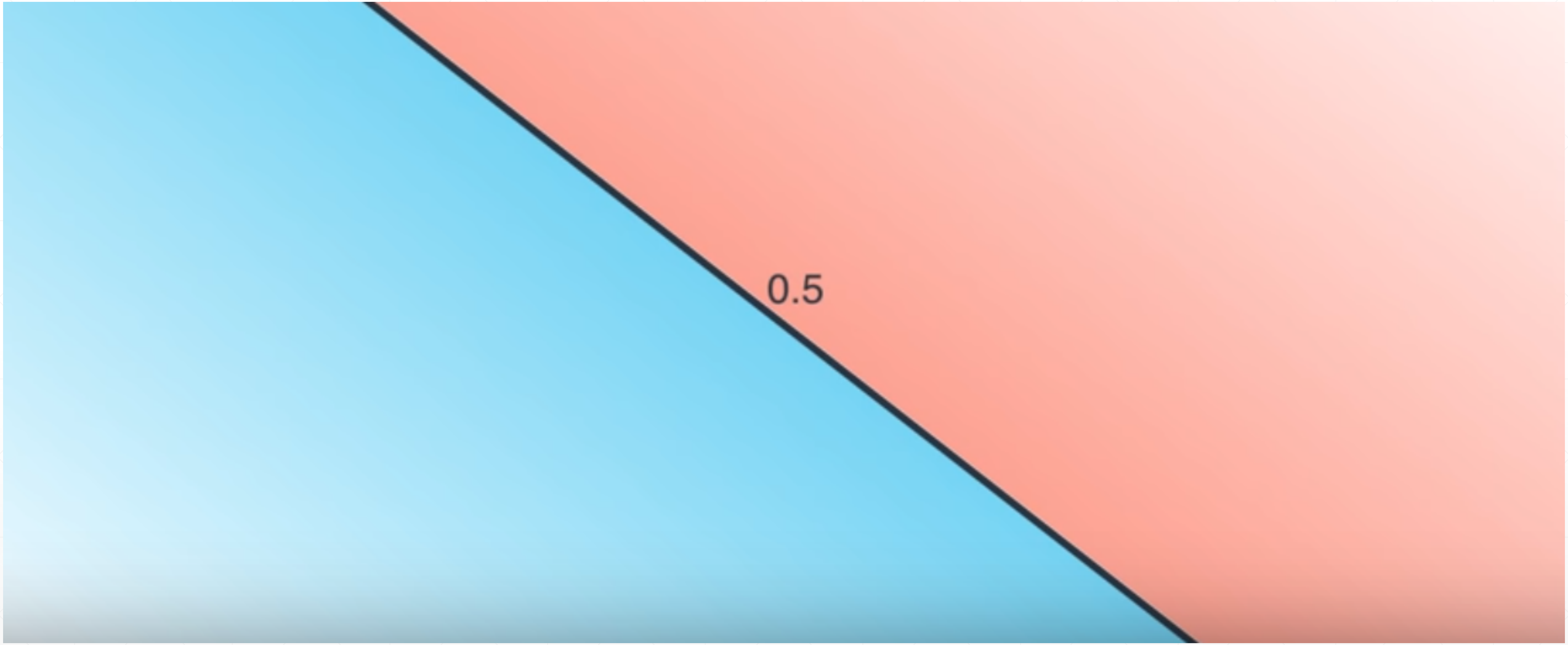
$$\textbf{Log-loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

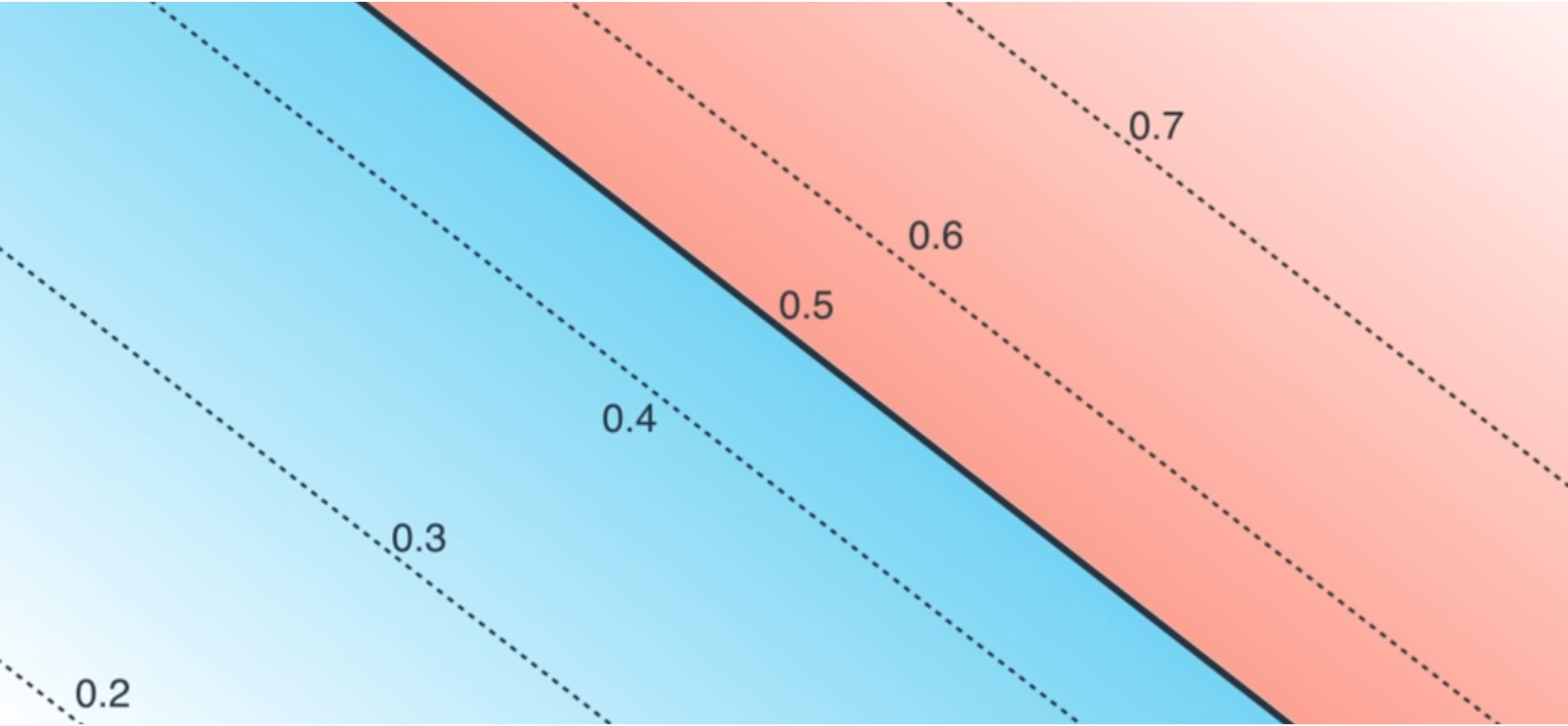


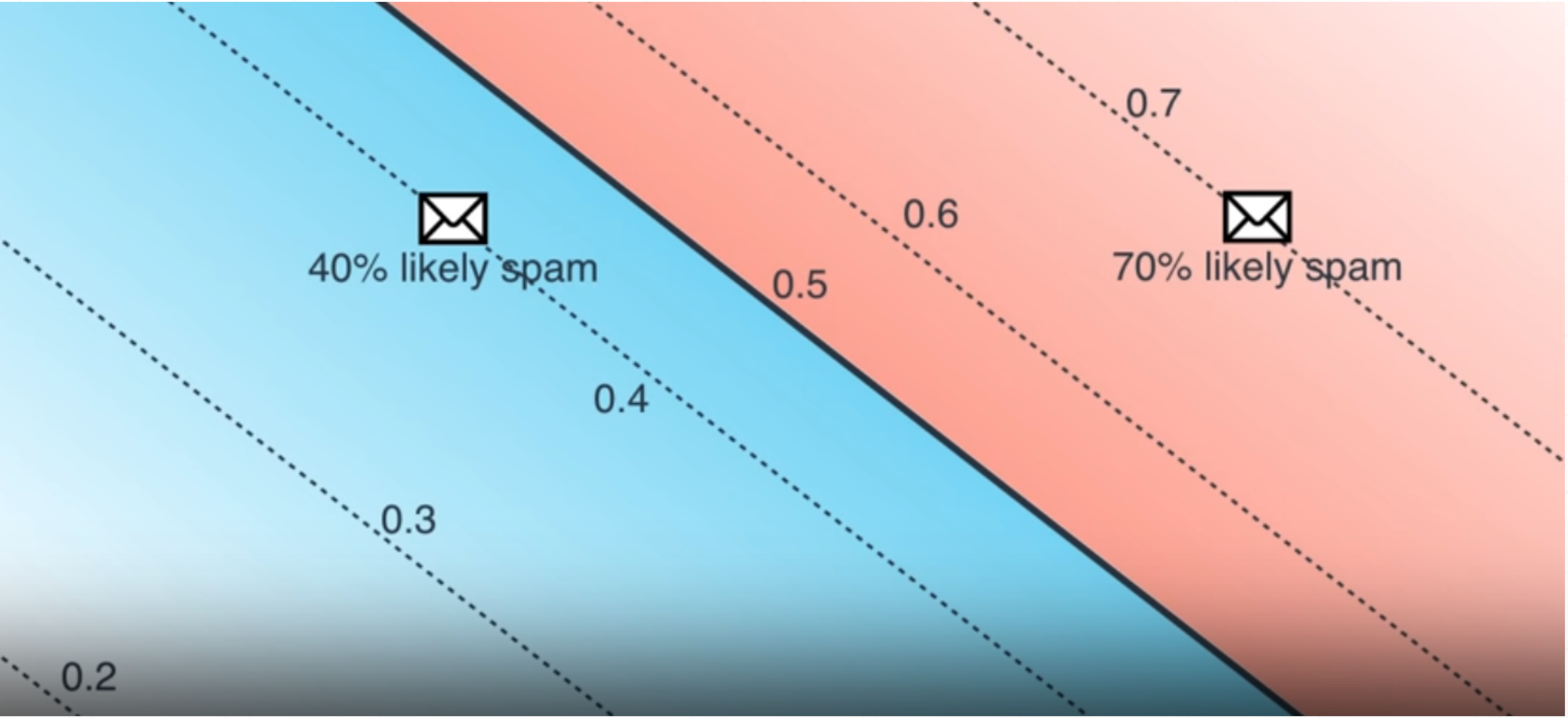
$$\textbf{Log-loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$



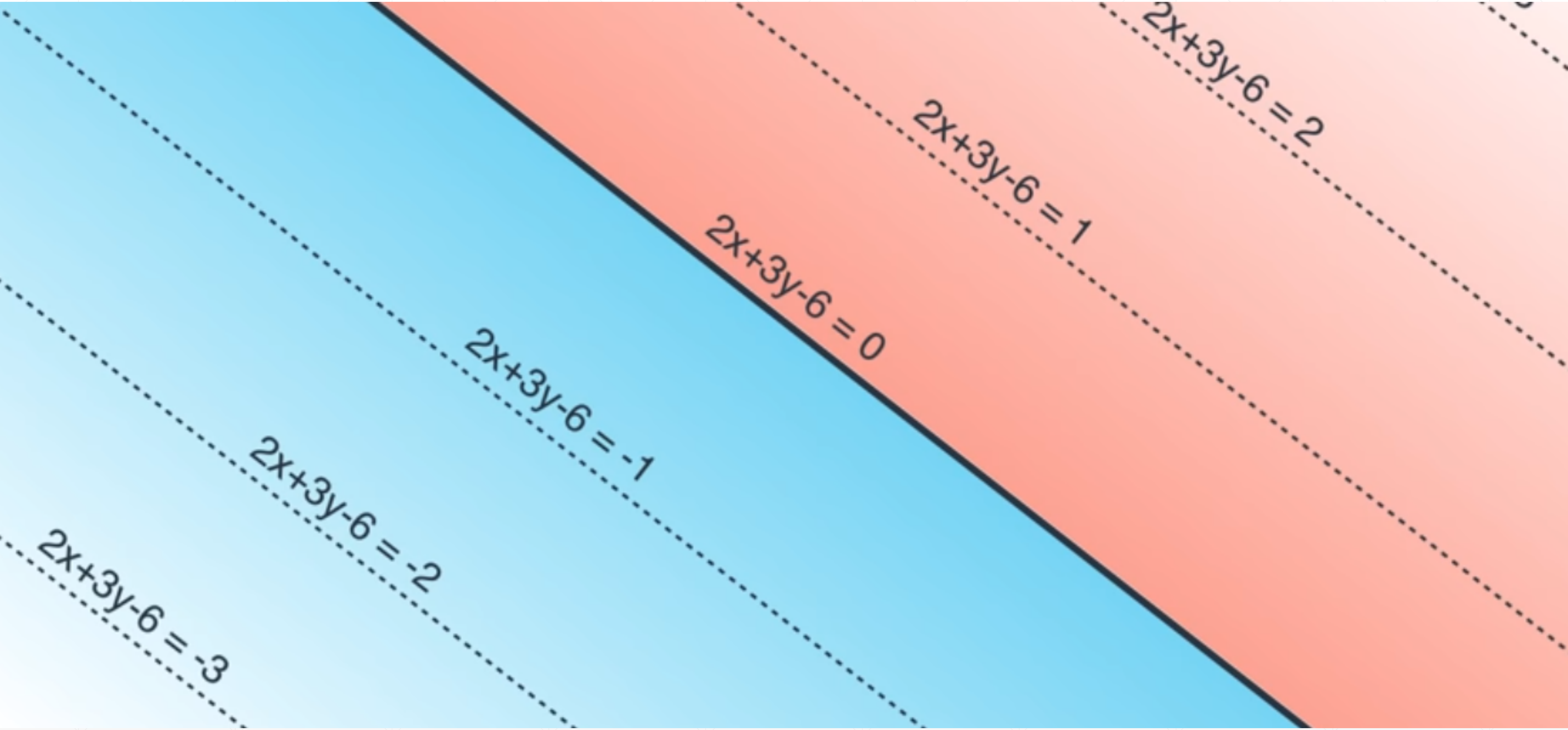
0
0
1 (spam)
1
1
0
0 (ham)
0
1
1
1
0


0.5


0.7
0.6
0.5
0.4
0.3
0.2

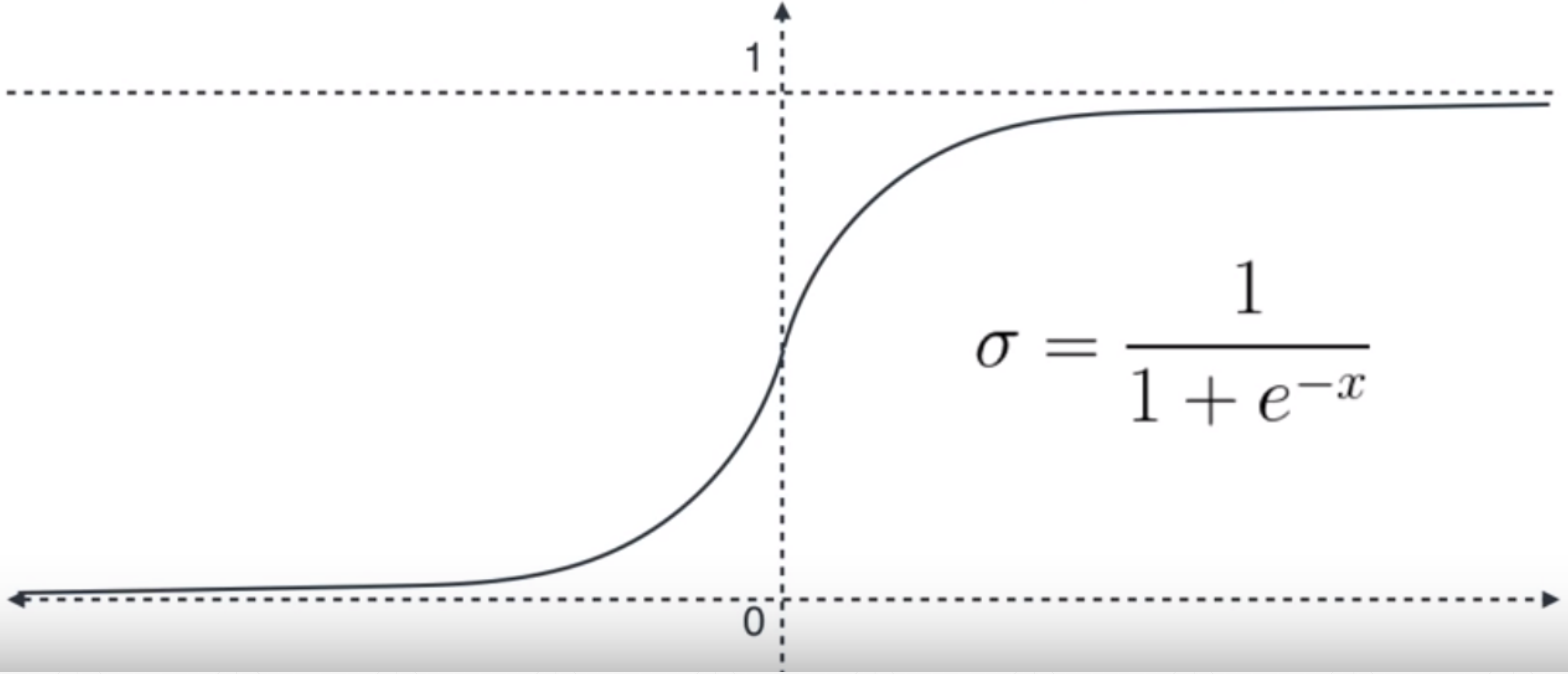


0.7
0.6
0.5
0.4
0.3
0.2
40% likely spam
70% likely spam

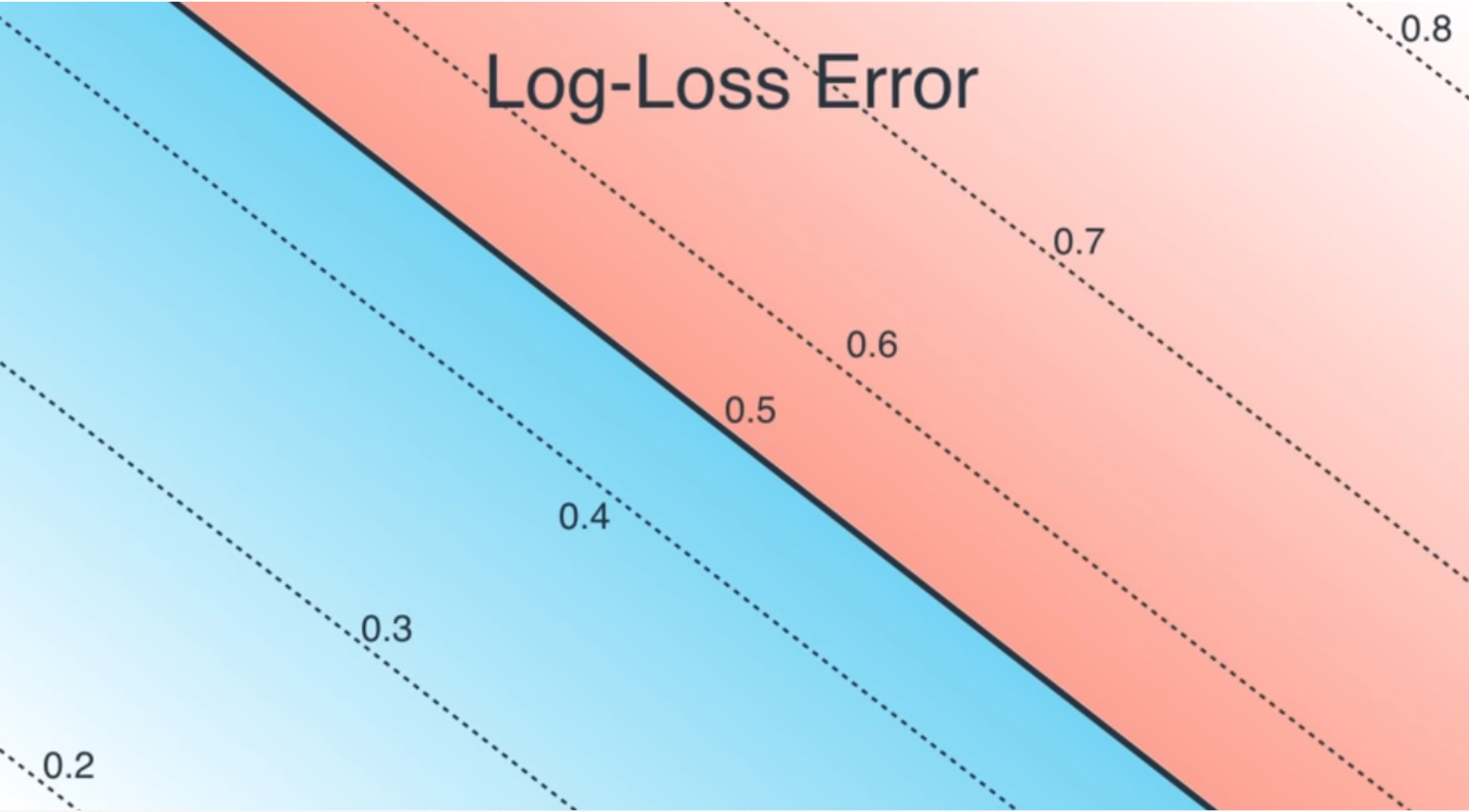


2x+3y-6 = 2
2x+3y-6 = 1
2x+3y-6 = 0
2x+3y-6 = -1
2x+3y-6 = -2
2x+3y-6 = -3

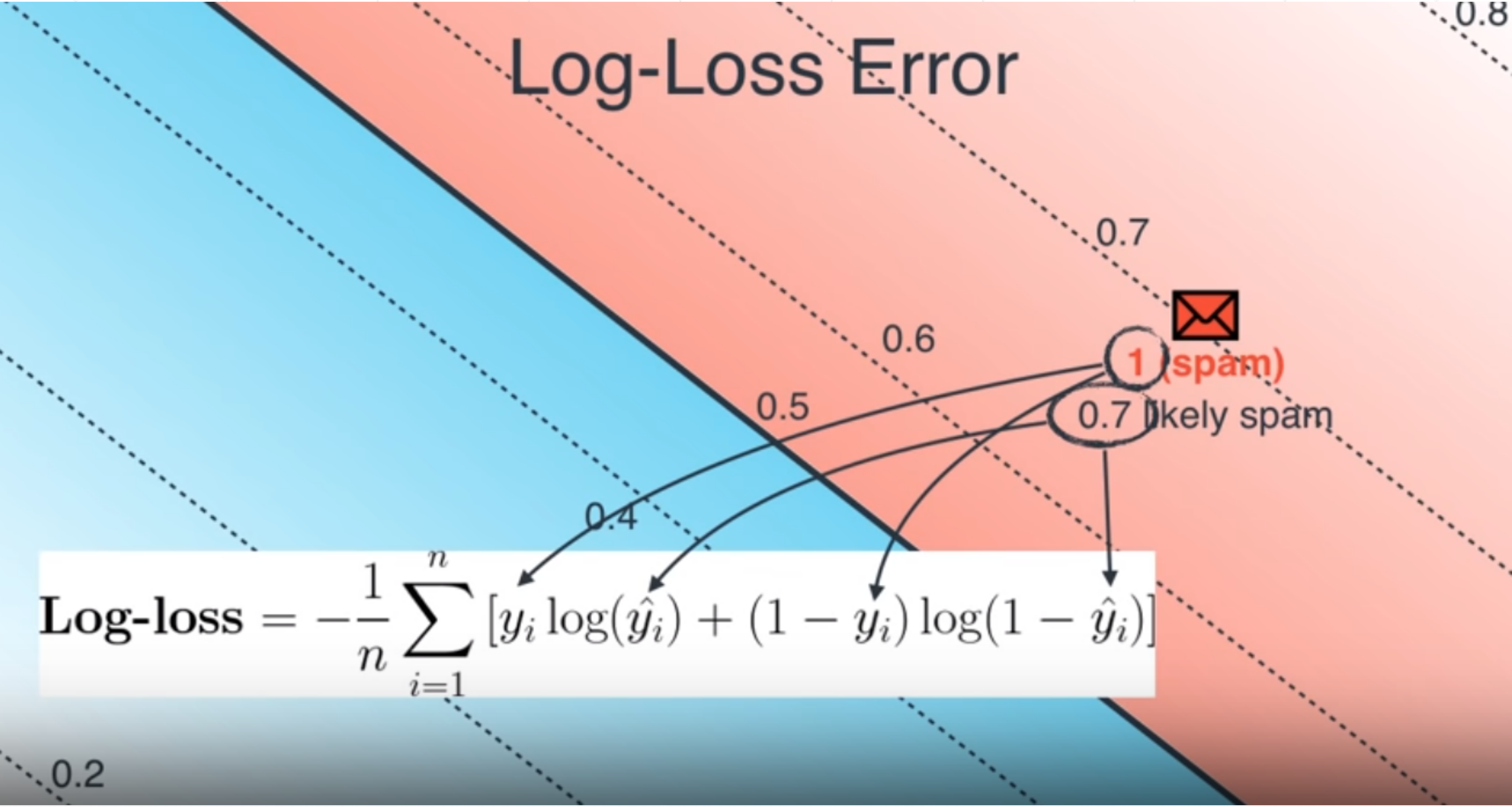
Activation function (sigmoid)



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

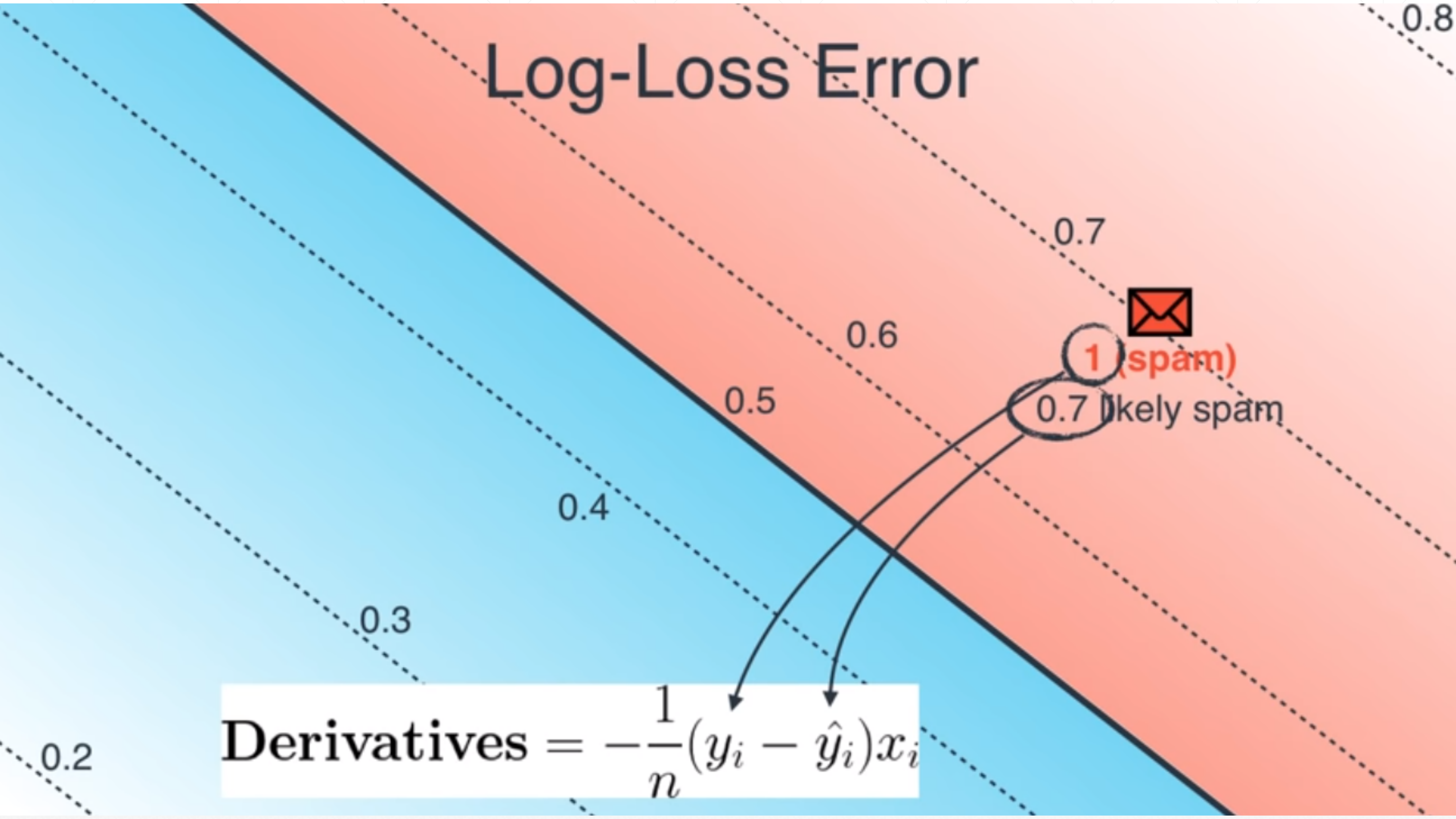

Log-Loss Error
0.8
0.7
0.6
0.5
0.4
0.3
0.2

Log-Loss Error



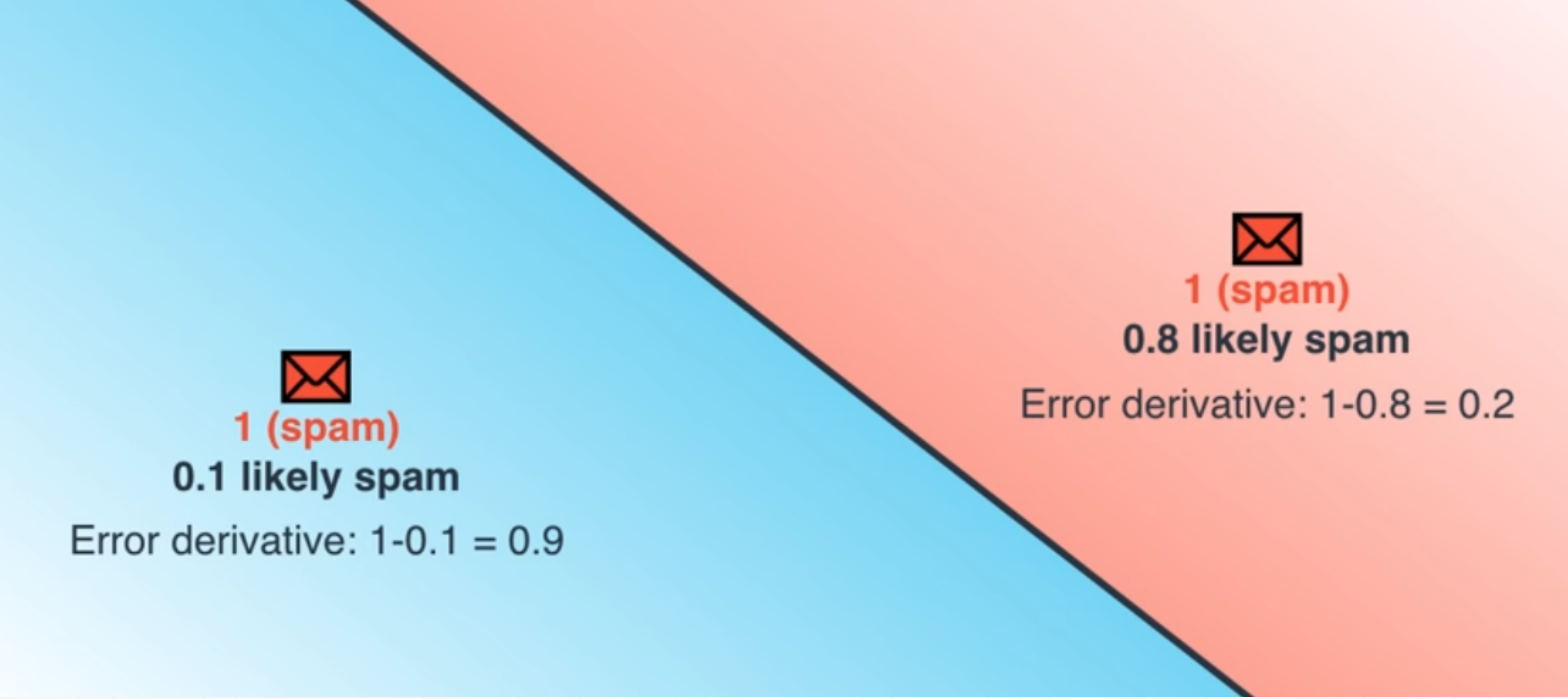
$$\mathbf{Log\text{-}loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Log-Loss Error

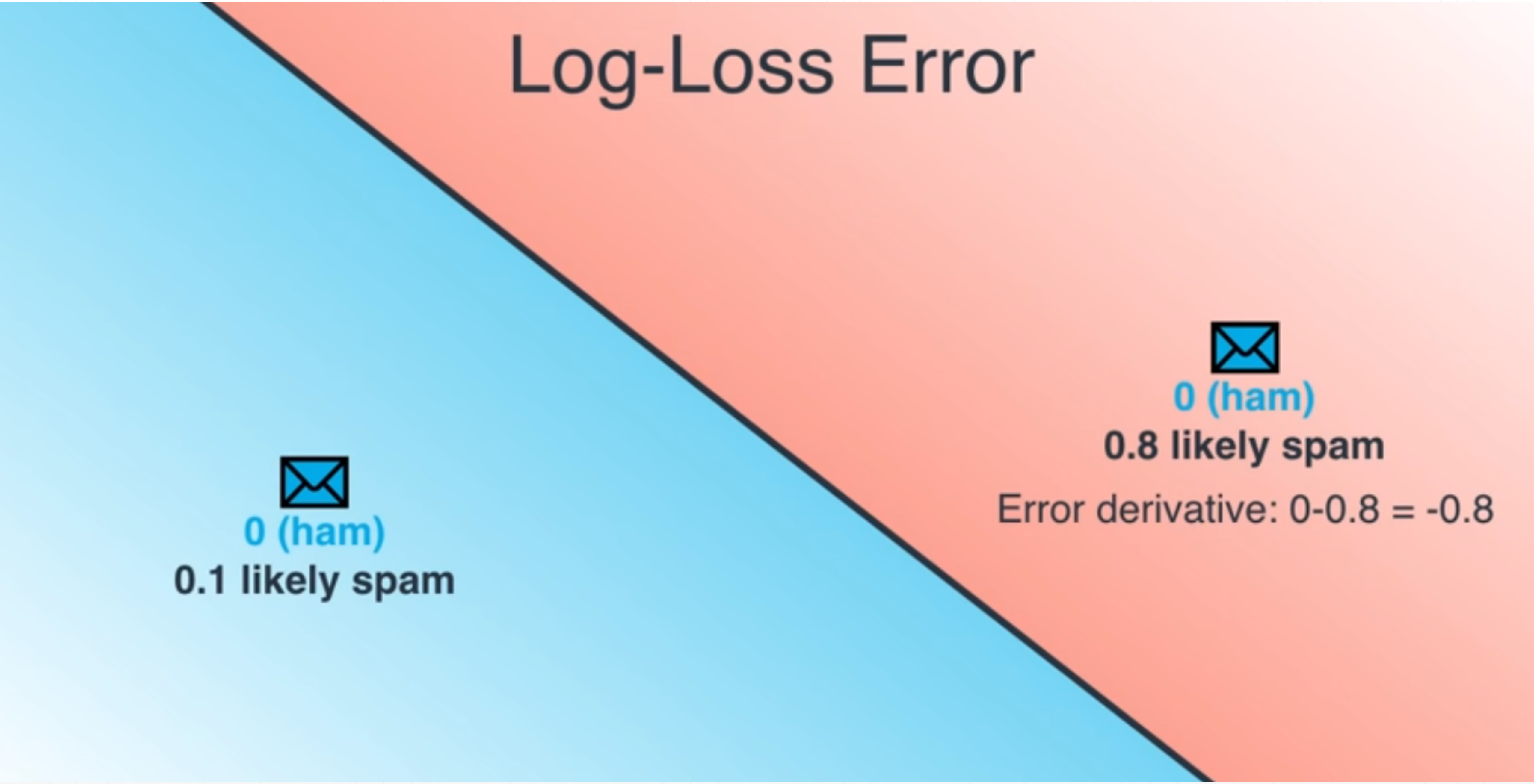


$$\textbf{Derivatives} = -\frac{1}{n}(y_i - \hat{y}_i)x_i$$

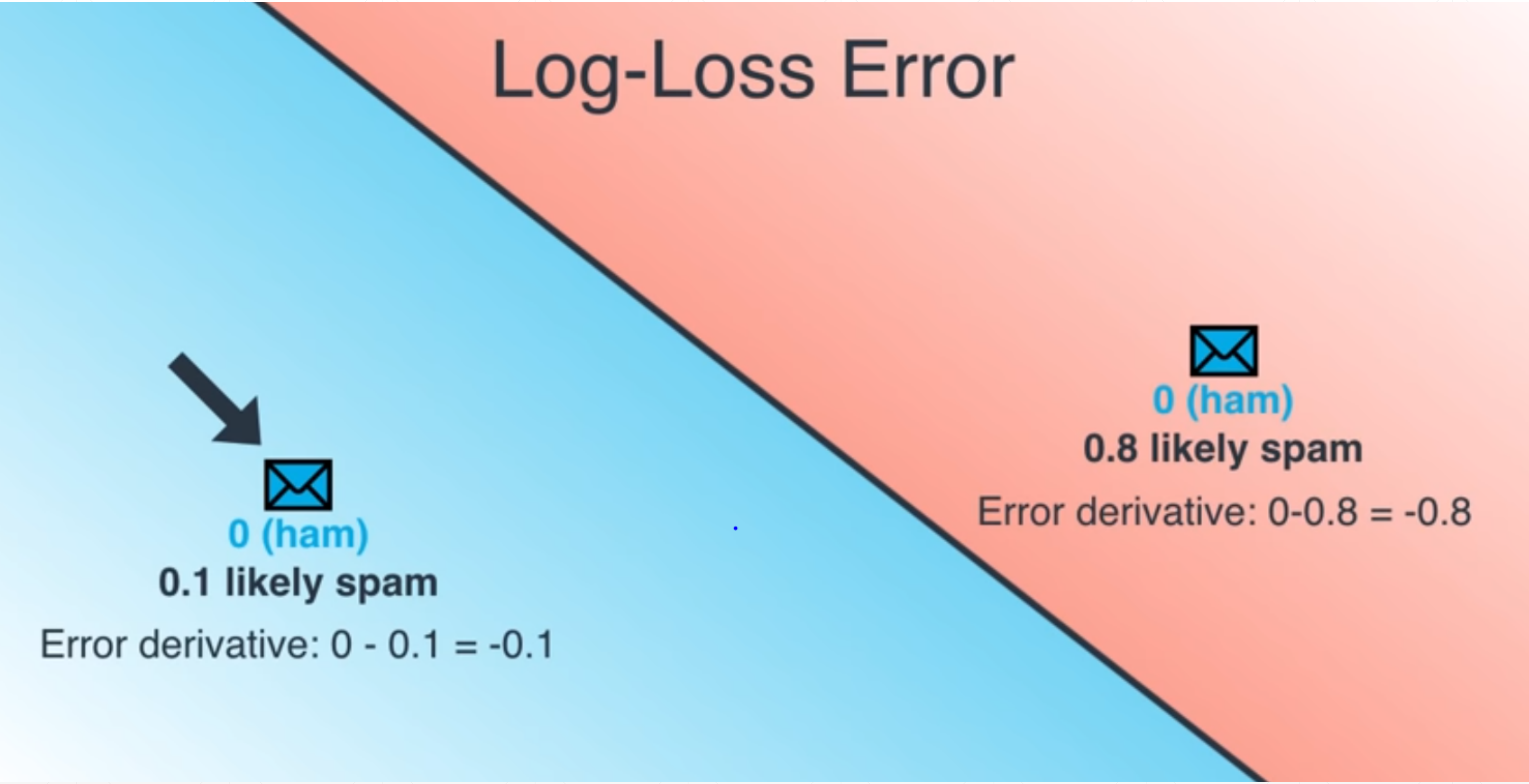
Log-Loss Error



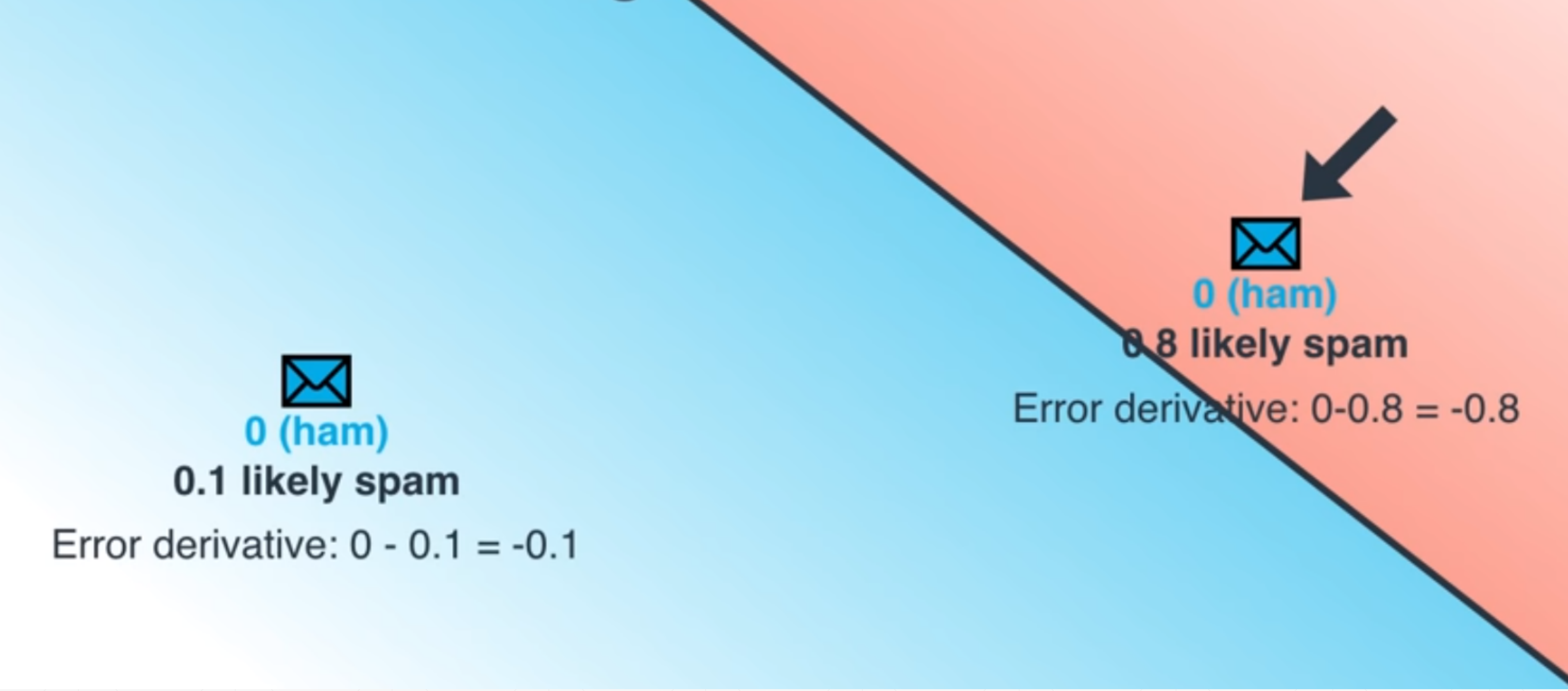
Log-Loss Error


Log-Loss Error
0 (ham)
0.1 likely spam
0 (ham)
0.8 likely spam
Error derivative: 0-0.8 = -0.8

Log-Loss Error

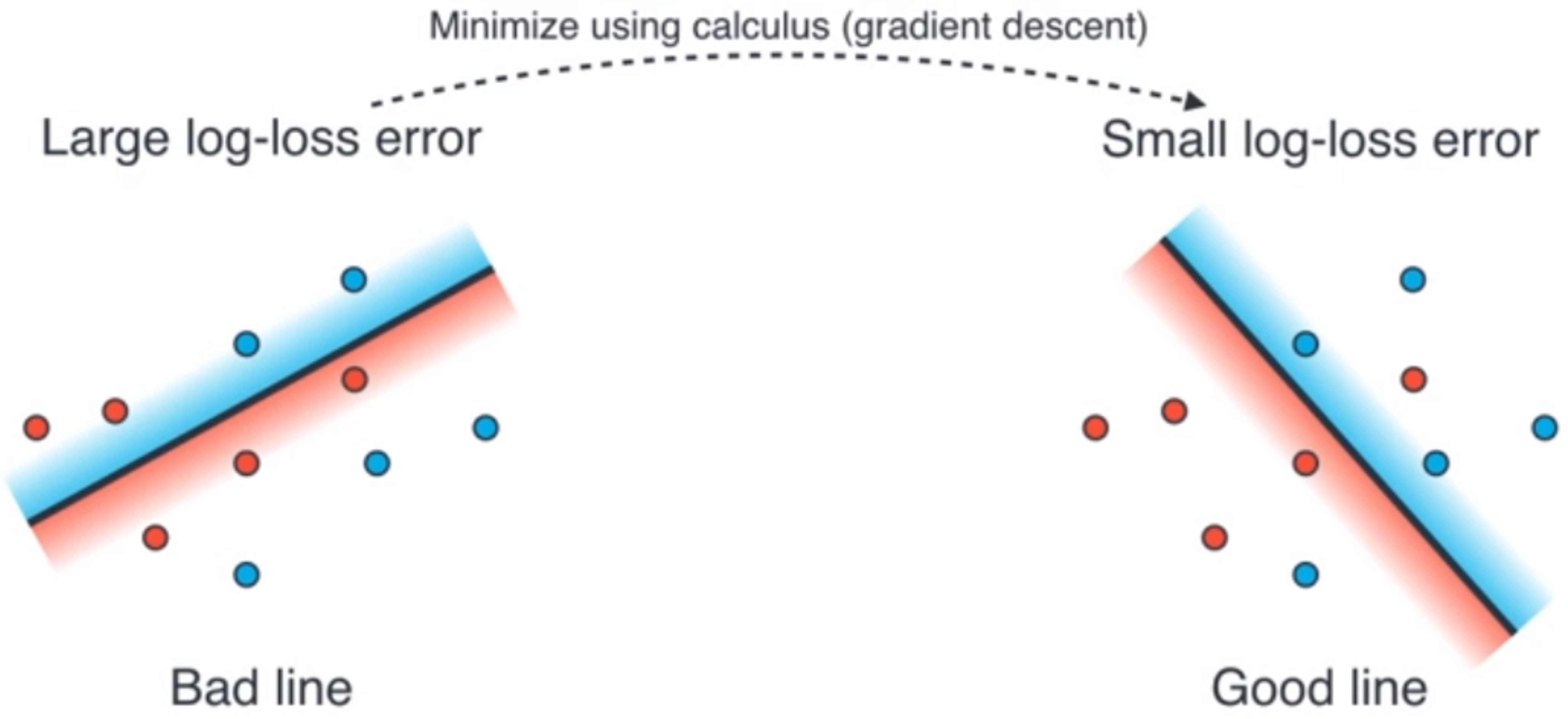


Log-Loss Error

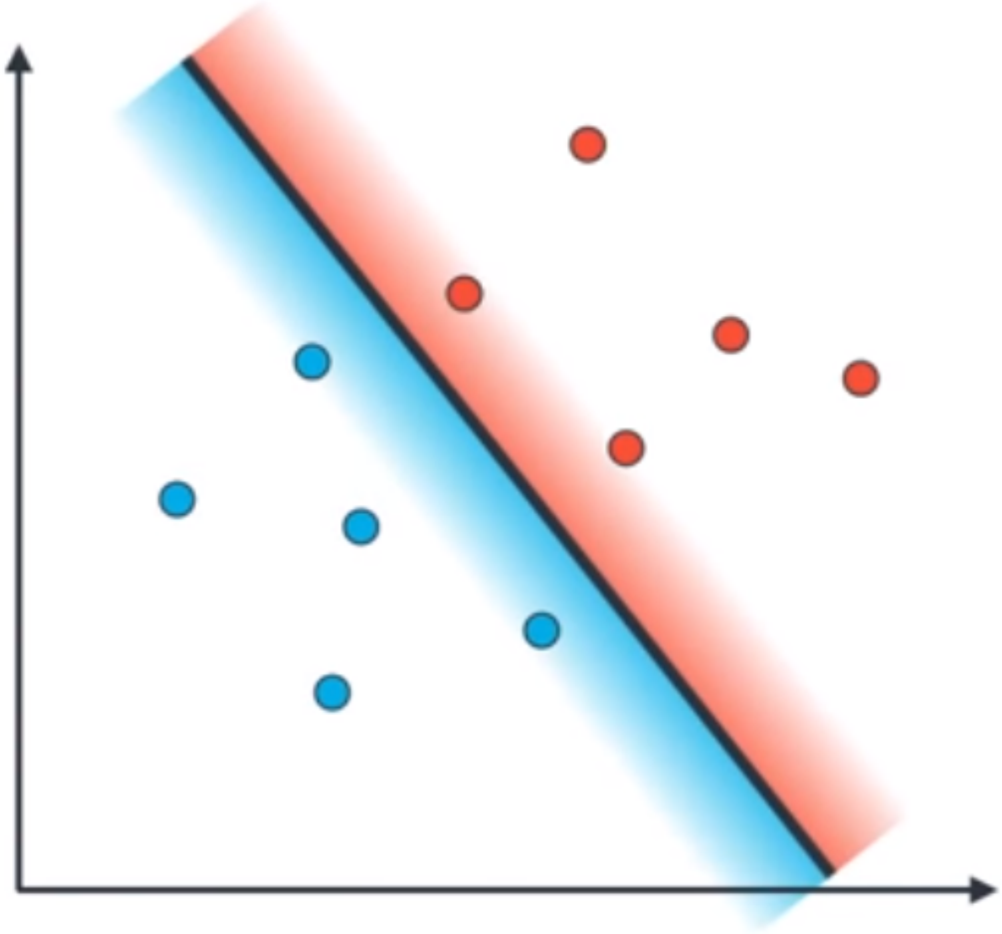


Log-loss error

Logistic regression algorithm



Logistic regression algorithm



Step 1: Start with a random line of equation $ax + by + c = 0$

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

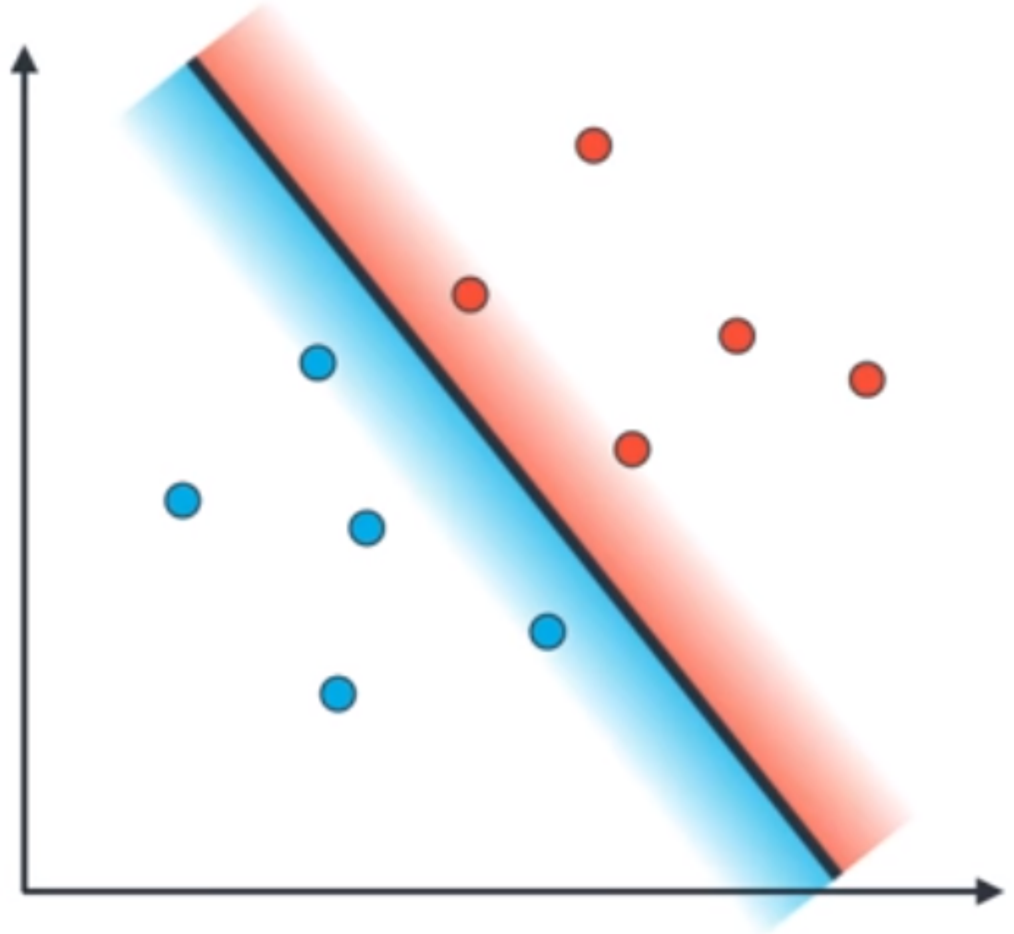
Step 3: Pick a small number. 0.01 (learning rate)

Step 4: (repeat 1000 times)

- Pick random point (p,q)
- If point is correctly classified:
  - Move line away from point
- If point is incorrectly classified
  - Move line towards point

Step 5: Enjoy your fitted line!

Logistic regression algorithm



Step 1: Start with a random line of equation $ax + by + c = 0$

Step 2: Pick a large number. 1000 (number of repetitions, or epochs)

Step 3: Pick a small number. 0.01 (learning rate)

Step 4: (repeat 1000 times)

- Pick random point (p,q)
  - Add $0.01(y - \hat{y})p$ to a
  - Add $0.01(y - \hat{y})q$ to b
  - Add $0.01(y - \hat{y})$ to c

Step 5: Enjoy your fitted line!

Math in the machine

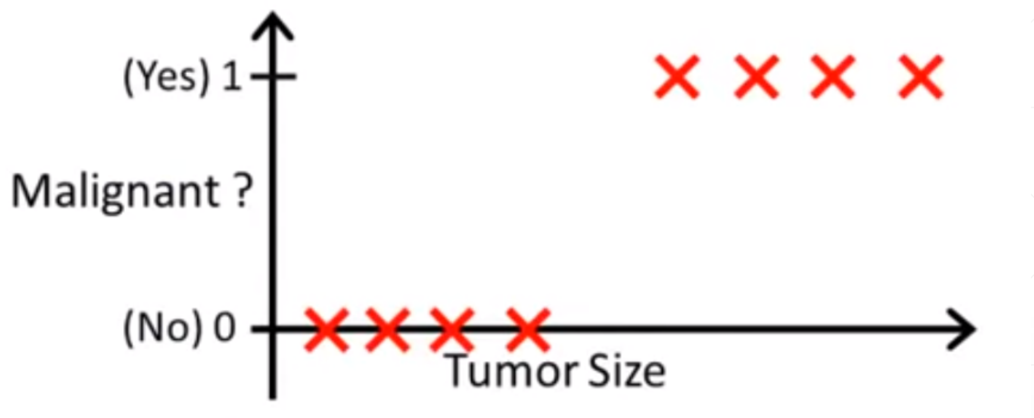
In case of Linear Regression

- Threshold classifier output

h(x) at 0.5

if h(x) >= 0.5, predicts “y=1”

if h(x) < 0.5, predicts “y=0”



- Classification y = 0 or 1

h(x) can be >1 or <0

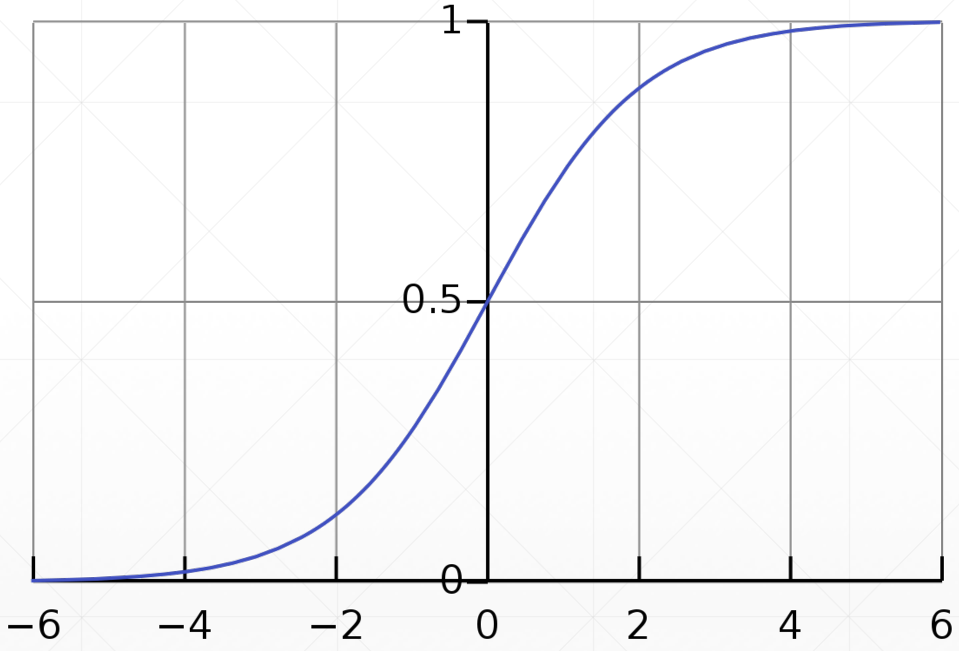
- Logistic Regression 0 <= h(x) <= 1

Logistic Regression

- Want 0 <= h(x) <= 1
- h(x) = g(mx+c)

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

- Sigmoid/ Logistic function
- Fit the parsmetrs to our data



Interpretation of our model

$h(x)$ = estimated probability that $y = 1$, input x

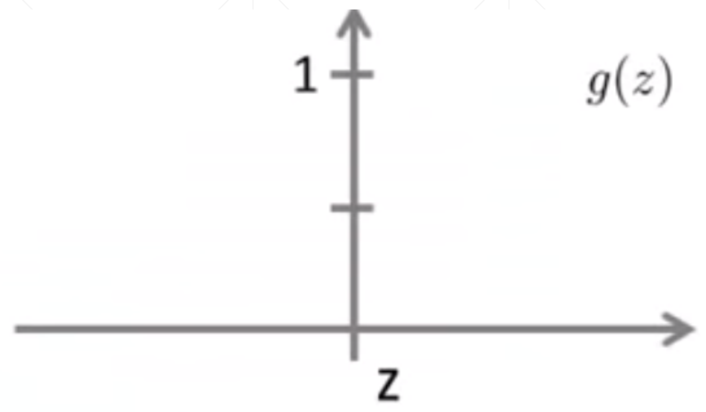
Tumor example

$h(x) = 0.7$ ---- Pateint has 70% chance of tumor being malignant

$P(y = 0) + P(y = 1) = 1$

Logistic regression

$$h_\theta(x) = g(\theta^T x)$$
$$g(z) = \frac{1}{1+e^{-z}}$$



We can suppose:

"y = 1" if h(x) >= 0.5

g(z) >= 0.5 when z >= 0

h(x) = g(mx + c) >= 0.5 so mx+c >= 0

"y=0" if h(x) < 0.5

g(z) < 0.5 when z < 0

h(x) = g(mx + c) < 0.5 so mx+c < 0

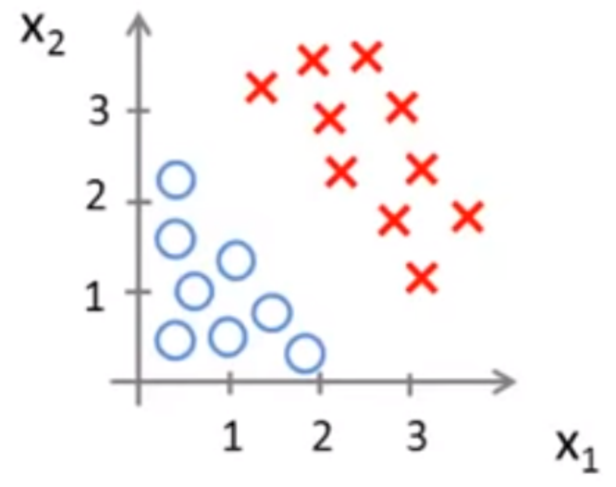
Decision Boundary

$h(x) = g\ (c + m_1x_1 + m_2x_2)$

$c = -3,\ m_1 = 1,\ m_2 = 1$

Predict "y =1" if $-3 + x_1 + x_2 >= 0$

$x_1 + x_2 >= 3$



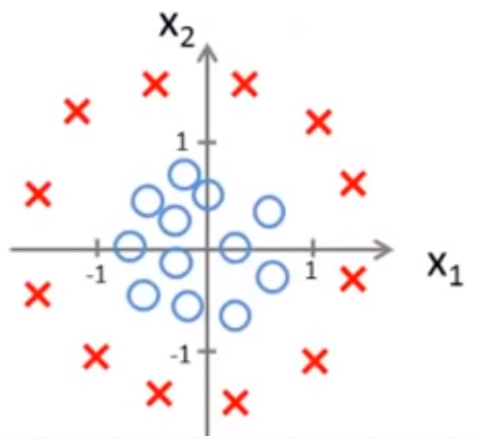
Seprates the region where y = 1 and y = 0

Decision boundary is the property of the parameters of the hypothesis not the property of dataset

Another example

$h(x) = g\ (c + m_1x_1 + m_2x_2 + m_3x_1^2 + m_4x_2^2)$

$c=0\ ,\ m_1 = 0,\ m_2 = 0,\ m_3 = 1,\ m_4 = 1$



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

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

Even we can do it for higher polynomial

Cost function

Linear regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Non – convex function

Come up with a different function that is convex

Logistic regression cost function

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

Gradient Descent

Same formula as Linear regression

We need to choose alpha

Advanced optimization algorithms

Conjugate Descent
BFGS
L-BFGS

Advantage:
- Often faster than gradient descent
- No need to manually pick alpha

Disadvantage:
- More complex

Multi – class classification

Discussion

Thank you!