Korea Aerospace University

2AI4203: Machine Learning

Lec03: Classification with Linear Models

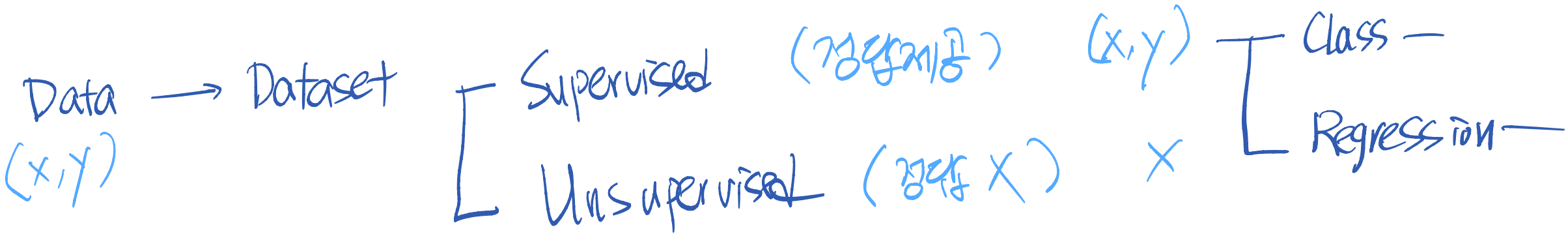
Jiyoon Shin

KAU College of AI Convergence

Sep. 16, 2026

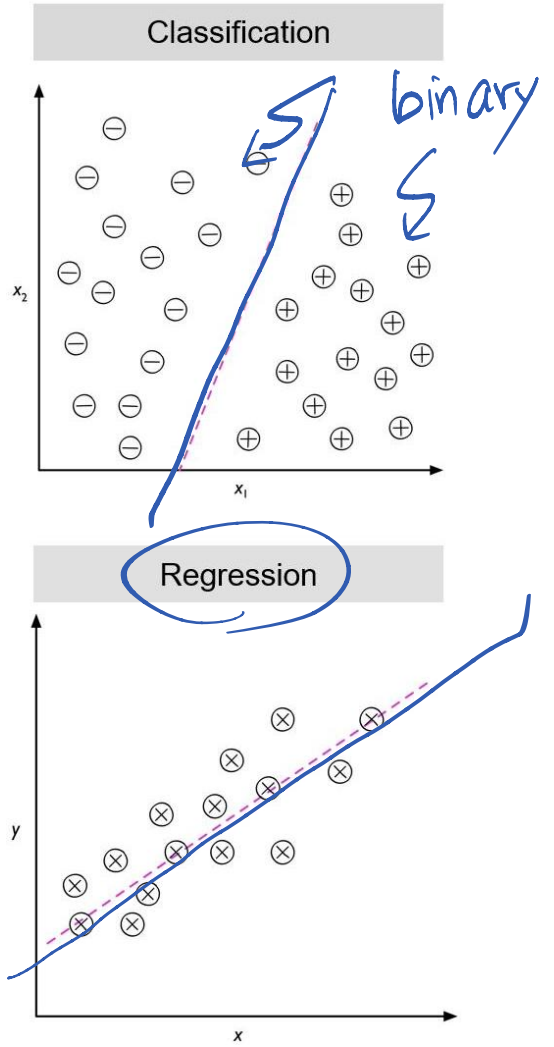
Recap:

Learning from Data & Linear Regression



Classification vs. Regression

- Classification
  - Predict a category (class)
  - Binary: Cat / Dog, Spam / Not spam
  - Multiclass: Cat / Dog / Bird / …
  - $y \in \{\text{class}_1, \text{class}_2, \ldots\}$
- Regression
  - Predict a continuous value
  - House features → Price
  - Exam scores → GPA
  - $y \in \mathbb{R}$



Linear Model

$$\hat{y} = wx + b$$

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$

- $\hat{y}$: prediction
- w / $w_1, \ldots, w_d$: slope / feature weights
- x / $x_1, \ldots, x_d$: input / input features
- b: bias

Loss Function

- Measures how well the model prediction matches the target
  - Small loss → better prediction
  - Large loss → worse prediction



mean square error

- Regression example (MSE):

$$L(w,b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

square

$wx+b$

error

- Goal: find parameters that minimize the loss

$$w^*, b^* = \arg\min_{w,b} L(w,b)$$

$\min L(w,b)$ = L의 최솟값

$\arg\min_{w,b} L(w,b)$ = L을 최소로 만드는 w랑 b

Gradient Descent

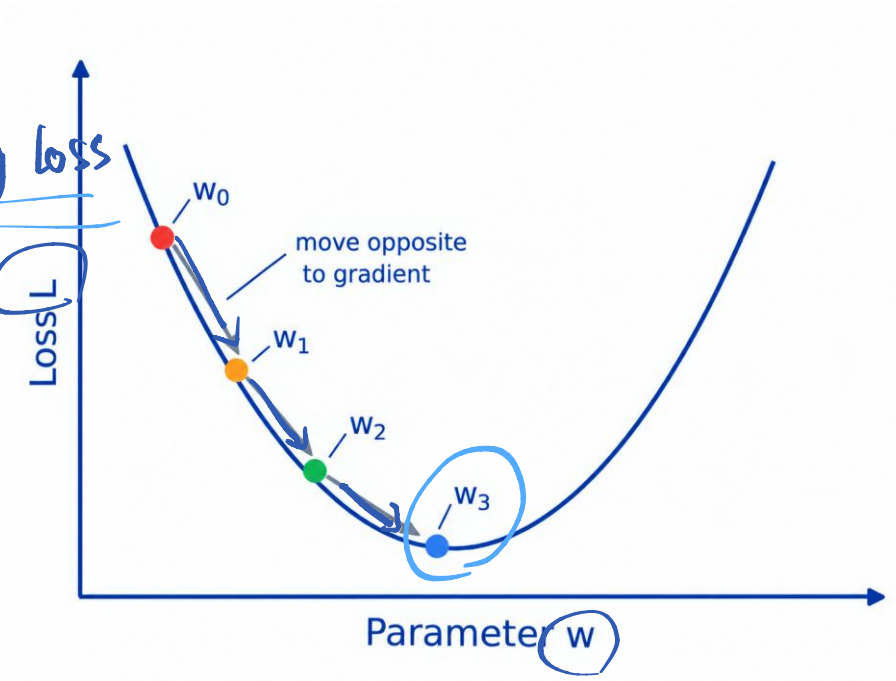
$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

$$b \leftarrow b - \eta \frac{\partial L}{\partial b}$$

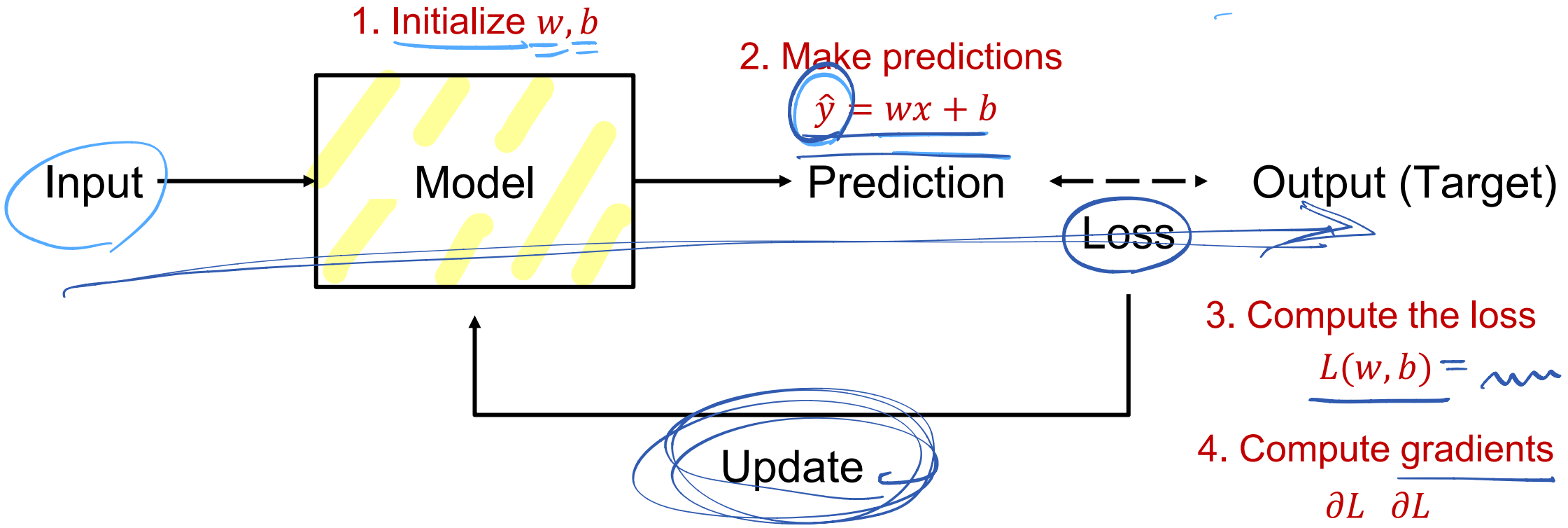
- Gradient: points towards increasing loss

  → move in the opposite direction

- Learning rate η: controls the step size
- Repeat: update parameters iteratively to reduce the loss



Machine Learning Workflow



learning paradigm

supervised

unsupervised

Task

classification

regression

Model

linear model

linear regression

binary class → logistic model regression

multiclass class → softmax regression

multinomial logistic regress...

Korea Aerospace University

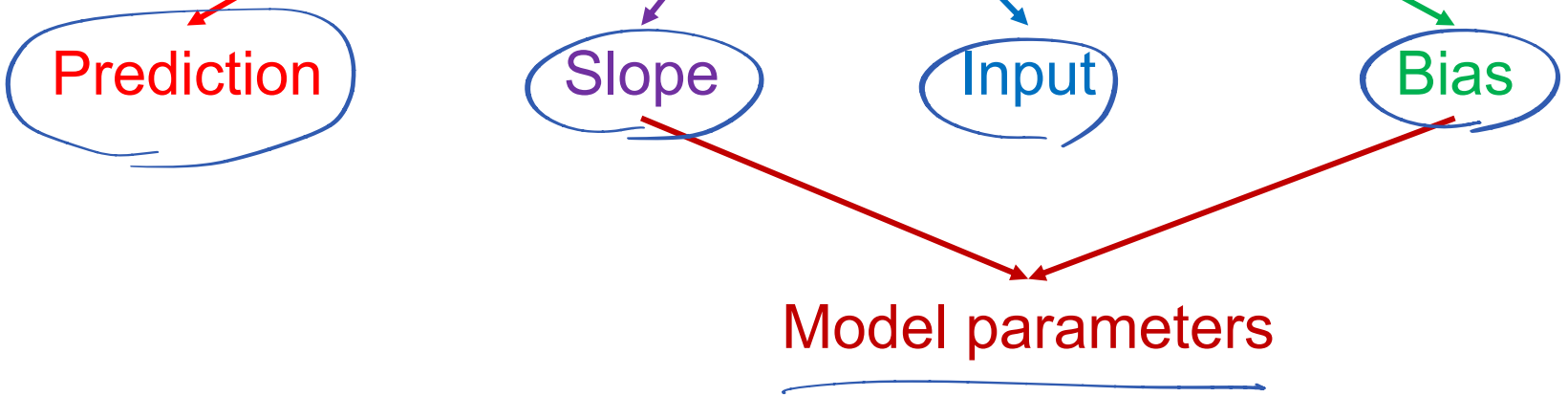
Part 1. Binary Classification

Recall: Linear Model

linear regression 기반주

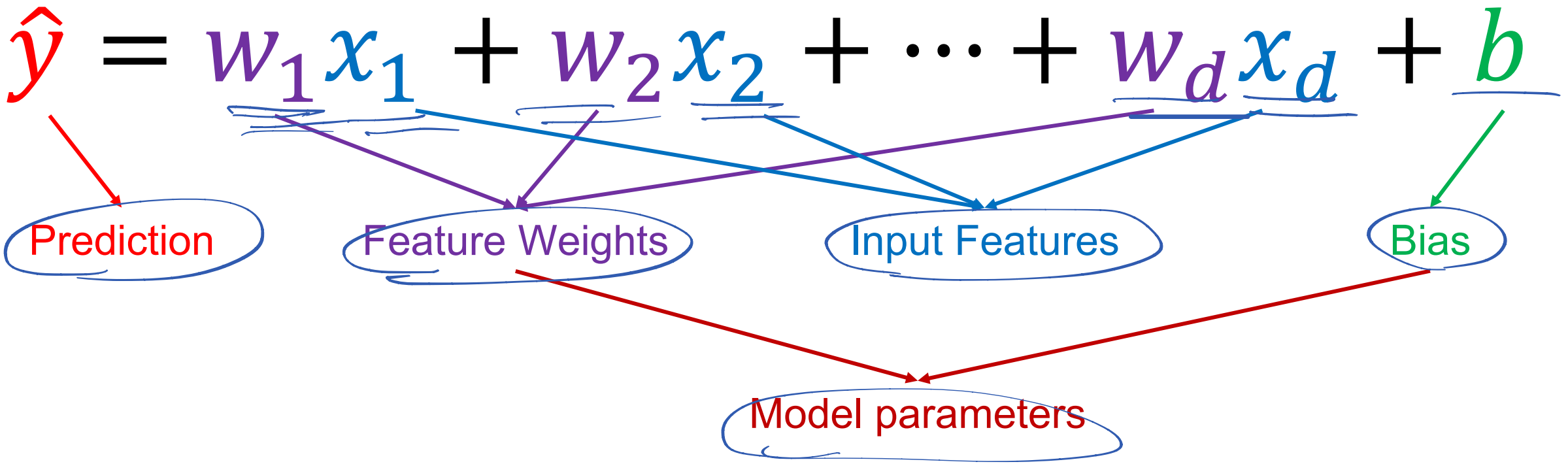
선형

$$\hat{y} = wx + b$$



Recall: Linear Model (Multiple Features)

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$



- Each feature contributes to the prediction through its weight

Vector Notation

- $\mathbf{x} \in \mathbb{R}^d$: feature vector
- $\mathbf{w} \in \mathbb{R}^d$: weight vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

$$\mathbf{w}^T\mathbf{x} = \begin{bmatrix} w_1 & w_2 & \cdots & w_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = w_1x_1 + w_2x_2 + \cdots + w_dx_d$$

$$\hat{y} = \mathbf{w}^T\mathbf{x} + b$$

$\hat{y} = \mathbf{W} \cdot \mathbf{X} + b$ 벡터끼리

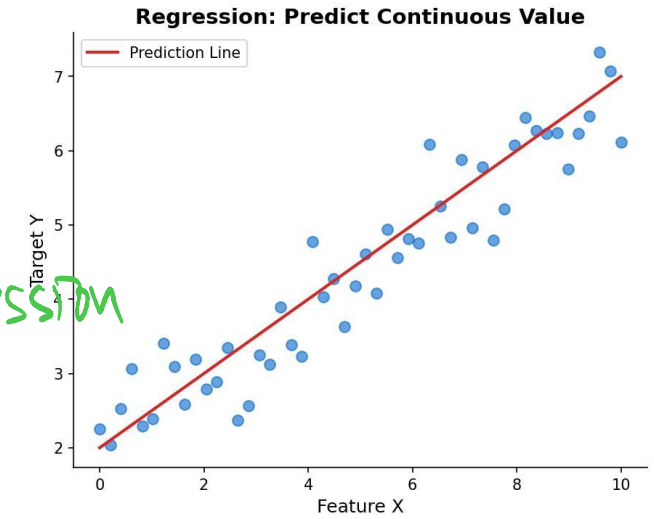
From Regression to Classification

- Regression

$$\hat{y} = \mathbf{w}^T\mathbf{x} + b$$

  - The model output can be used directly as the prediction
  - Example
    - $\hat{y} = 4.20 \longrightarrow$ house price
    - $\hat{y} = 3.7 \longrightarrow$ GPA

linear model → linear regression



- Classification

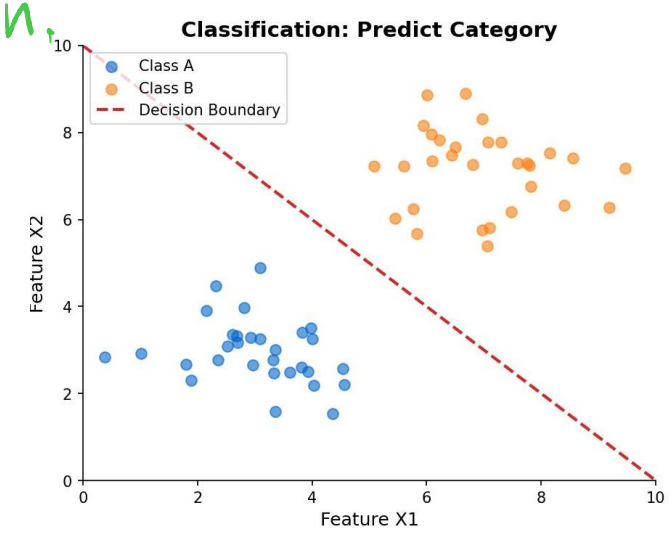
? model → binary classification.

$$y \in \{0, 1\}$$

  - The target is a discrete class label
  - But the linear model can output any real number

$$\mathbf{w}^T\mathbf{x} + b \in (-\infty, +\infty)$$

  - How can we use a linear model for classification?



Linear Score

$$\hat{y} = w^T x + b$$

$$z = \mathbf{w}^T \mathbf{x} + b$$

- For classification, we use the linear model output as a score
- The score can take any real value

$$z \in (-\infty, +\infty)$$

- The sign of z indicates which class the model favors

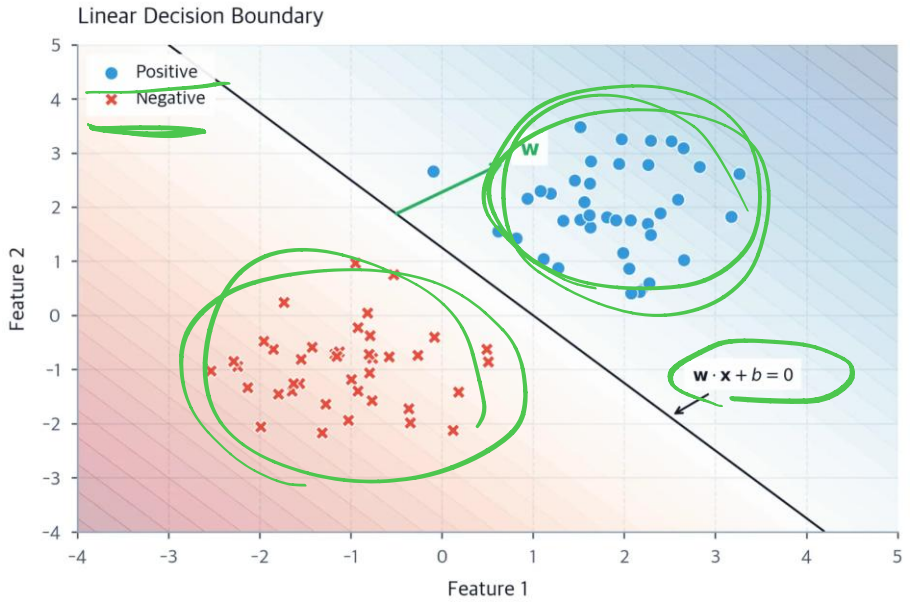
Binary Decision

$$\hat{y} = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

$$z = \mathbf{w}^T \mathbf{x} + b$$

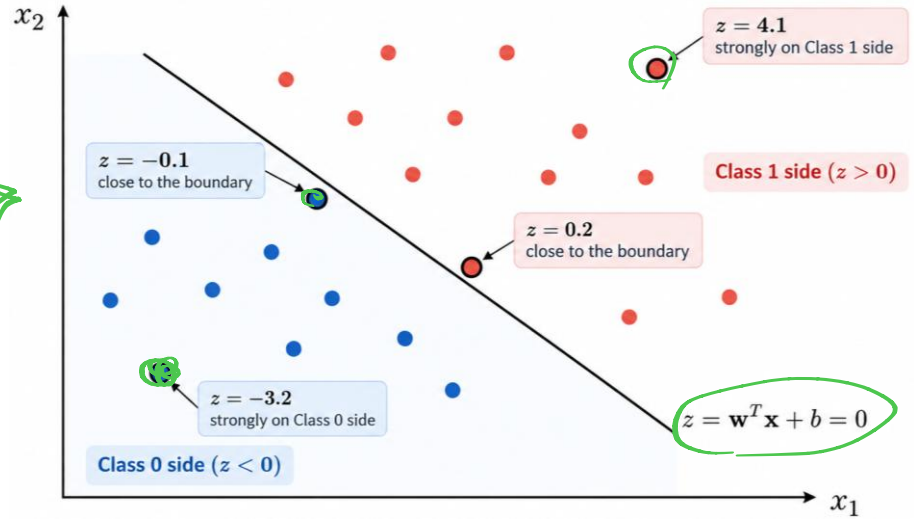
score

- $z > 0 \longrightarrow$ Class 1
- $z < 0 \longrightarrow$ Class 0
- $z = 0 \longrightarrow$ Decision boundary



Score to Probability

- The linear score tells us which class is favored, but it is not a probability
- Example
  - $z = -3.2$: strongly favors Class 0
  - $z = -0.1$: slightly favors Class 0
  - $z = 0.2$: slightly favors Class 1
  - $z = 4.1$: strongly favors Class 1



- Sign of z $\longrightarrow$ which side of the boundary
- Magnitude of z ($|z|$) $\longrightarrow$ how strongly the model favors that side

Score to Probability

- A linear score can take any real value

$$z \in (-\infty, +\infty)$$

$$\hat{y} = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

- A probability must lie between 0 and 1

$$p \in [0, 1]$$

- We need a function that maps z to a valid probability

$$z \in \mathbb{R} \rightarrow ? \rightarrow p \in [0, 1]$$

Korea Aerospace University

Logistic Regression

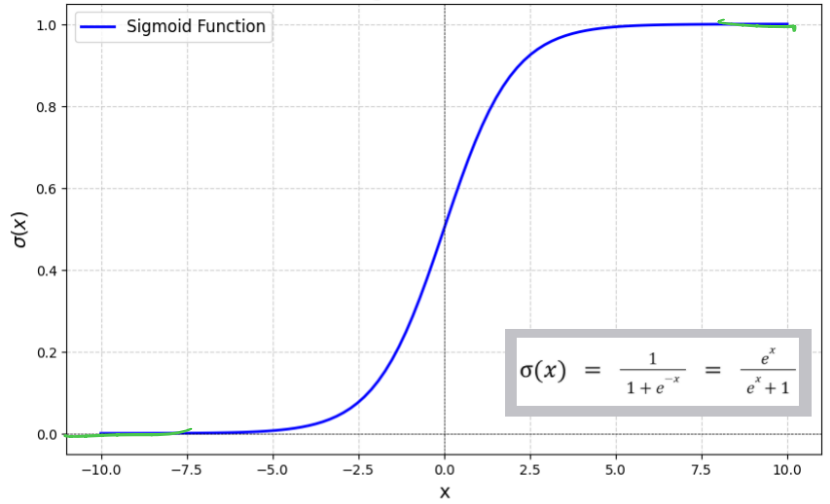
Sigmoid Function

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

- Maps any real-valued score z to a value between 0 and 1

$$-\infty < z < +\infty \longrightarrow 0 < \sigma(z) < 1$$

- Also called the logistic function
- Produces an S-shaped curve



Sigmoid Function

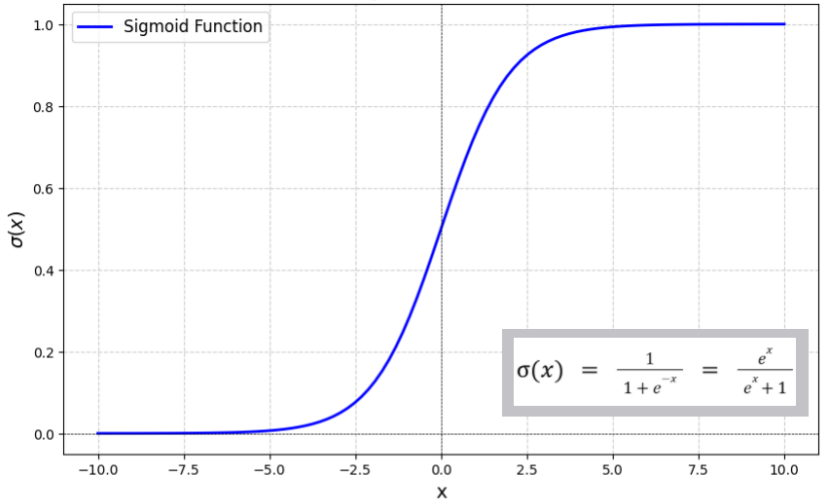
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- Properties
  - $z \to -\infty$ $\quad \sigma(z) \to 0$
  - $z = 0$ $\quad \sigma(z) = 0.5$
  - $z \to +\infty$ $\quad \sigma(z) \to 1$

단조함수

  - Monotonic: larger z gives larger $\sigma(z)$
  - Bounded: $0 < \sigma(z) < 1$
  - Differentiable: useful for gradient-based learning

미분가능!!



Logistic Model

$$z = \mathbf{w}^T\mathbf{x} + b$$

linear model

$$\downarrow$$

$$p = \sigma(z)$$

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

$$\downarrow$$

$$p = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}}$$

0.7

$$\begin{bmatrix} p & \text{class 1} \\ 1-p & \text{class 0} \end{bmatrix}$$

- First, compute a linear score
- Then, transform the score using the sigmoid function
- The output is interpreted as the probability of Class 1

$$P(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$$

Probability

$$P(y = 1|\mathbf{x}) = p$$

$$P(y = 0|\mathbf{x}) = 1 - p$$

- p is the predicted probability of Class 1
- $1 - p$ is the predicted probability of Class 0

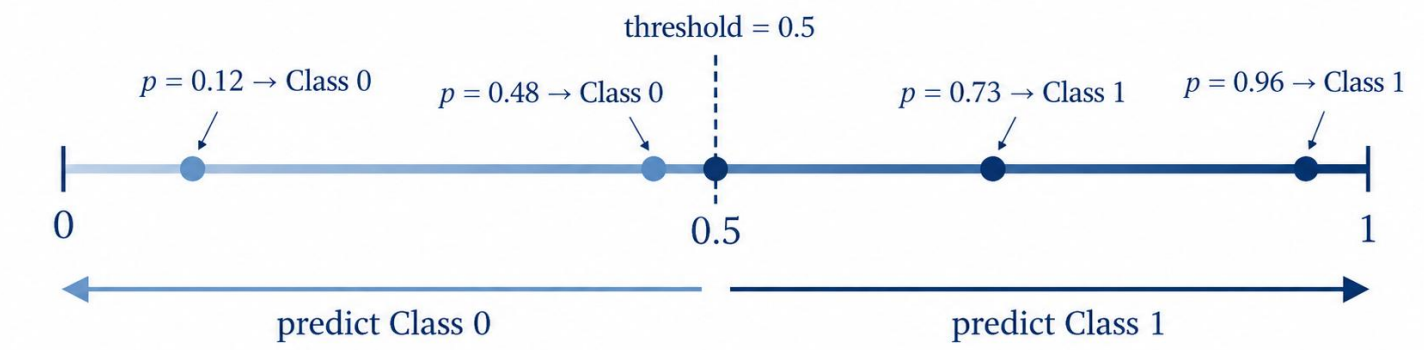
- Higher p means stronger preference for Class 1

Threshold

$$p = P(y = 1|\mathbf{x})$$

$$\hat{y} = \begin{cases} 1, & p \geq 0.5 \\ 0, & p < 0.5 \end{cases}$$

- Example
  - $p = 0.12 \longrightarrow$ Class 0
  - $p = 0.48 \longrightarrow$ Class 0
  - $p = 0.73 \longrightarrow$ Class 1
  - $p = 0.96 \longrightarrow$ Class 1



- 0.5 is a common default threshold for binary logistic regression
- The threshold can be adjusted depending on the application

Binary Decision (revisited)

$z = w^T x + b$

$$\hat{y} = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$z = 0 \rightarrow$ decision bou –

$$p = 0.5$$

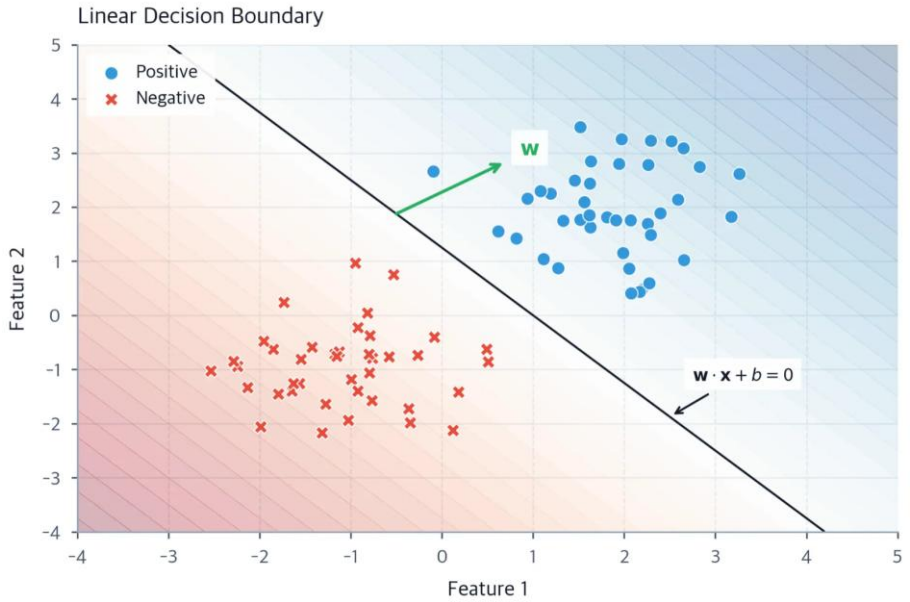
$$\sigma(z) = 0.5 \iff z = 0$$

$$z = \mathbf{w}^T \mathbf{x} + b$$

$$\mathbf{w}^T \mathbf{x} + b = 0$$

$$\frac{1}{1+e^{-z}} = 0.5$$

- $p < 0.5 \longrightarrow$ Class 0
- $p > 0.5 \longrightarrow$ Class 1
- $p = 0.5 \longrightarrow$ decision boundary



Example

- Given

$$z = 0.8x - 4$$

where x is study time in hours

  - Case 1

$$x = 3$$

$$z = 0.8(3) - 4 = -1.6$$

$$p = \sigma(-1.6) \approx 0.17$$

$$\hat{y} = 0$$

  - Case 2

$$x = 5$$

$$z = 0$$

$$p = 0.5$$

Example

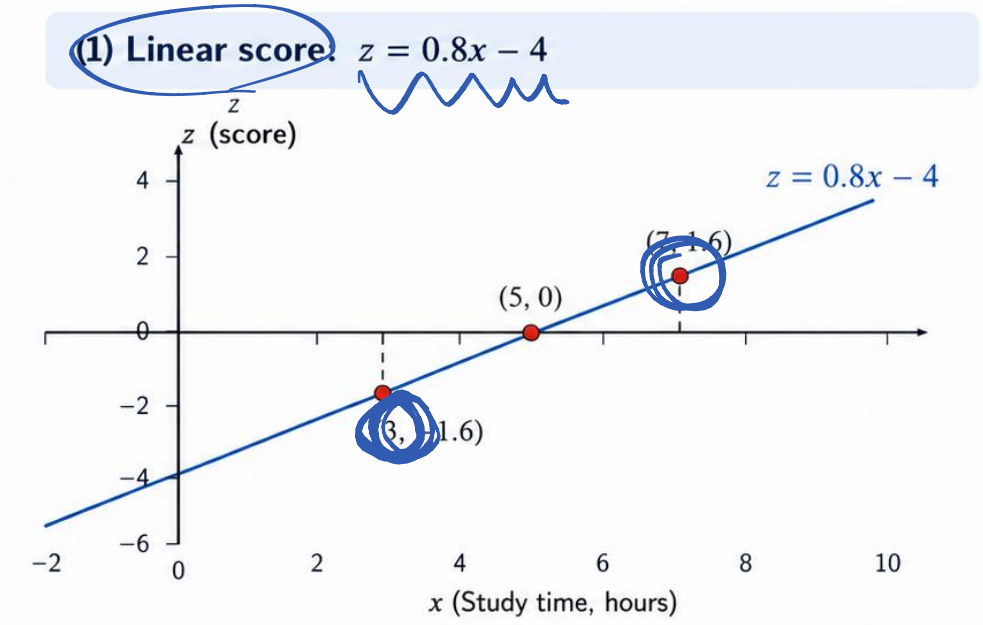
- Case 3

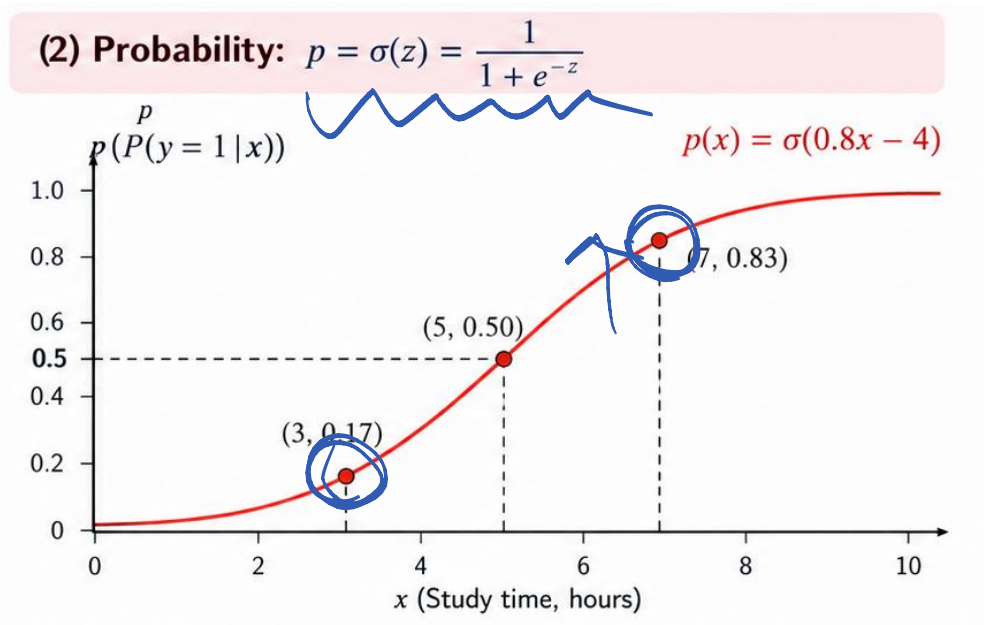
$$x = 7$$

① $$z = 0.8(7) - 4 = 1.6$$

② $$p = \sigma(1.6) \approx 0.83$$

$$\hat{y} = 1$$





Score vs. Probability

Linear Score z	Probability p
$(-\infty, +\infty)$	$(0, 1)$
$z = \mathbf{w}^T\mathbf{x} + b$	$p = \sigma(z)$
Unbounded	Bounded
Not a probability	Probabilistic interpretation

$$
\begin{cases}
z < 0 \Leftrightarrow p < 0.5 \\
z = 0 \Leftrightarrow p = 0.5 \\
z > 0 \Leftrightarrow p > 0.5
\end{cases}
$$

Logistic Regression

$$x \longrightarrow z = w^T x + b \longrightarrow p = \sigma(z) \longrightarrow \hat{y}$$

$$\hat{y} = \begin{cases} 1, & p \geq 0.5 \\ 0, & p < 0.5 \end{cases}$$

- Linear function ⟶ score
- Sigmoid ⟶ probability
- Threshold ⟶ class prediction

- How do we learn $\mathbf{w}$ and b?

$\hat{y} = w^T x + b \longrightarrow L = \frac{1}{n}\sum (y - \hat{y})^2$

5.0 4.1

Learning Logistic Regression

$z = w^T x + b \longrightarrow p = \sigma(z) \xrightarrow{\text{thres}} \hat{y} \begin{matrix} 0 \\ < \\ 1 \end{matrix}$ $y \begin{matrix} 0 \\ < \\ 1 \end{matrix}$

Loss Function

- A loss function measures how well the model prediction matches the target
  - Small loss ⟶ good prediction
  - Large loss ⟶ poor prediction
  - Training aims to find parameters that minimize the loss
- Regression

$$\hat{y} \in \mathbb{R}$$

MSE: $(y - \hat{y})^2$

- Classification

$$p = P(y = 1|\mathbf{x})$$

Need a loss for probabilistic predictions

Loss Function

$$p = P(y = 1|\mathbf{x})$$

- Assume $y = 1$

	Prediction	Interpretation	Desired loss
1	$p = 0.99$	Correct & confident	Very small
2	$p = 0.70$	Correct	Small
3	$p = 0.50$	Uncertain	Medium
4	$p = 0.10$	Wrong & confident	Large

- If $y = 0$: the opposite behavior is desired
  - Small $p \longrightarrow$ good
  - Large $p \longrightarrow$ poor

Binary Cross-Entropy (BCE)

$$L = -[y \log p + (1 - y) \log(1 - p)]$$

- $y \in \{0, 1\}$: true label
- $p = P(y = 1|\mathbf{x})$: predicted probability of Class 1

- BCE reflects both correctness and confidence
  - Correct & confident prediction → small loss
  - Wrong & confident prediction → large loss

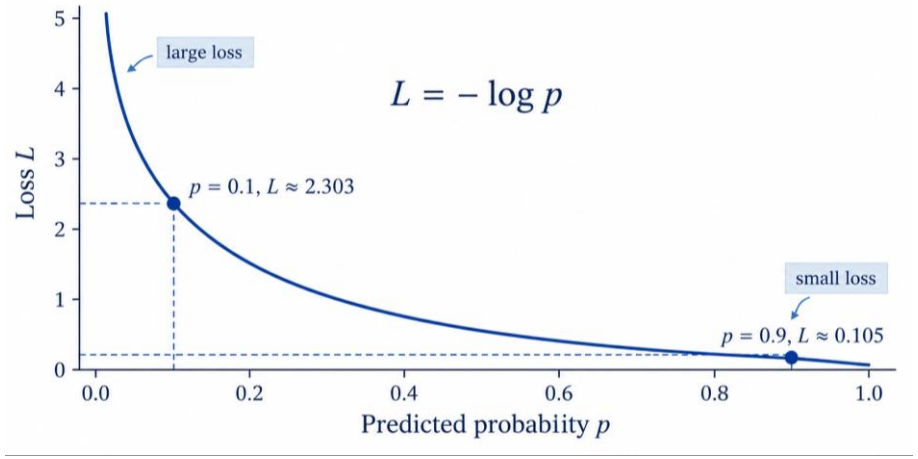
Binary Cross-Entropy

- When $y = 1$

$$L = -[y \log p + (1-y)\log(1-p)]$$
$$= -[1 \log p + 0 \log(1-p)] = -\log p$$

- $p \to 1$ $\quad L \to 0$
- $p \to 0$ $\quad L \to \infty$

- Example
  - $p = 0.9 \implies L = -\log 0.9 \approx 0.105$
  - $p = 0.1 \implies L = -\log 0.1 \approx 2.303$



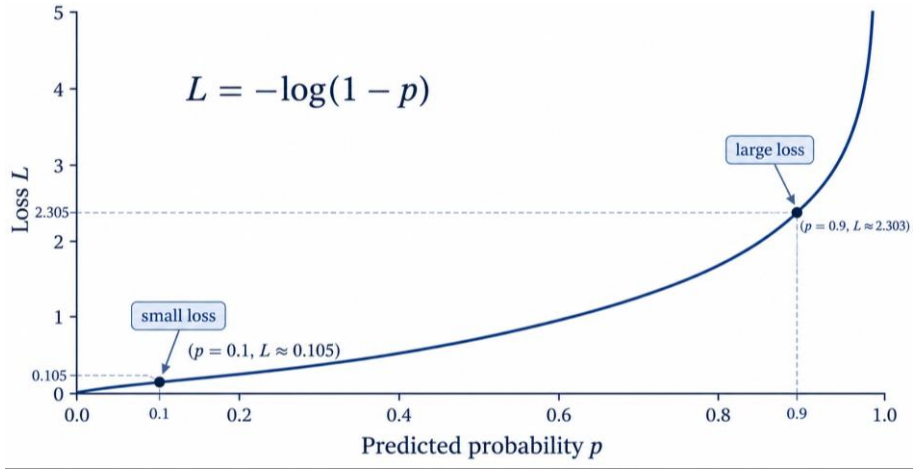
Binary Cross-Entropy

- When $y = 0$

$$L = -[y \log p + (1-y)\log(1-p)]$$
$$= -[0 \log p + 1 \log(1-p)] = -\log(1-p)$$

- $p \to 0$ $\quad L \to 0$
- $p \to 1$ $\quad L \to \infty$

- Example
  - $p = 0.1 \implies L = -\log 0.9 \approx 0.105$
  - $p = 0.9 \implies L = -\log 0.1 \approx 2.303$



Dataset Loss

- For n training examples:

$$(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$$

- For each sample

$$p_i = P(y_i = 1|\mathbf{x_i}) = \sigma(\mathbf{w}^T\mathbf{x_i} + b)$$

- Average loss

$$L(\mathbf{w}, b) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log p_i + (1 - y_i)\log(1 - p_i)]$$

Logistic Regression Training

Binary Classification task

$$\mathbf{x} \longrightarrow z = \mathbf{w}^T\mathbf{x} + b \longrightarrow p = \sigma(z)$$

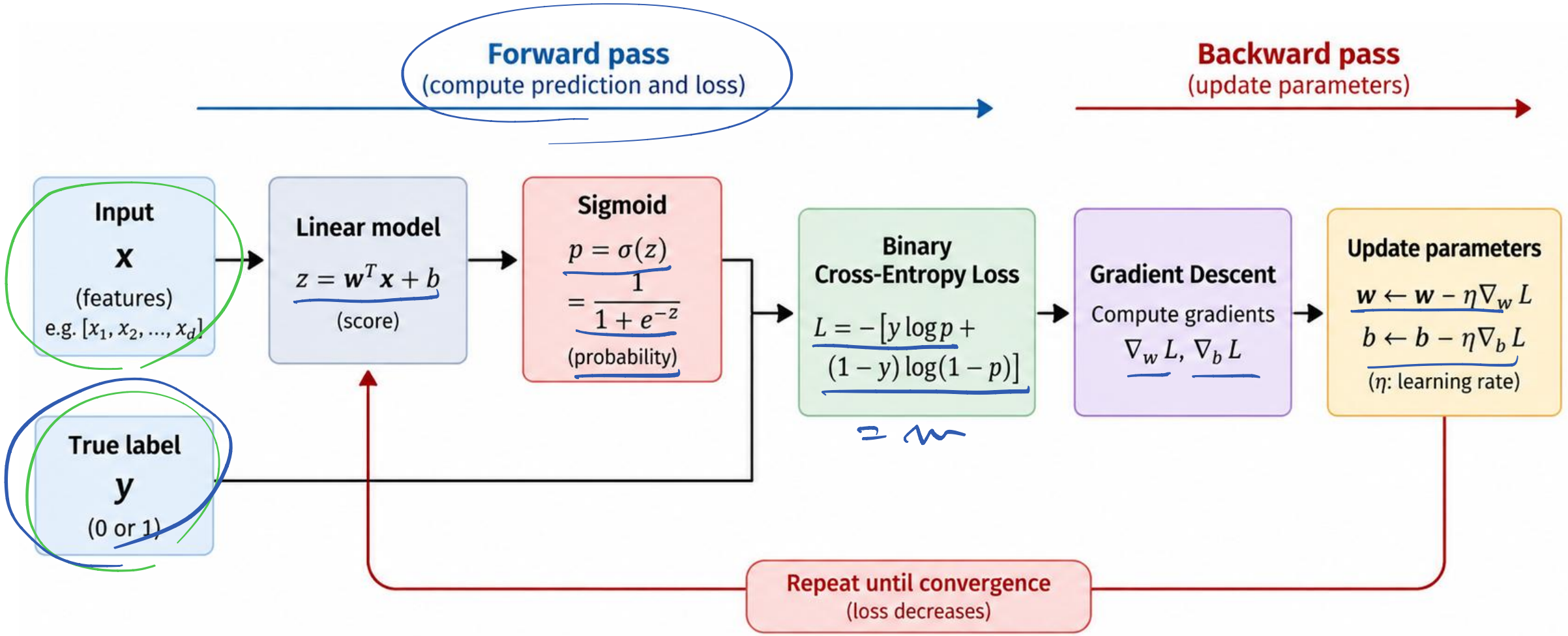
$$(p, y) \longrightarrow L_{BCE} \longrightarrow \mathbf{w}, b \text{ update}$$

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} L$$

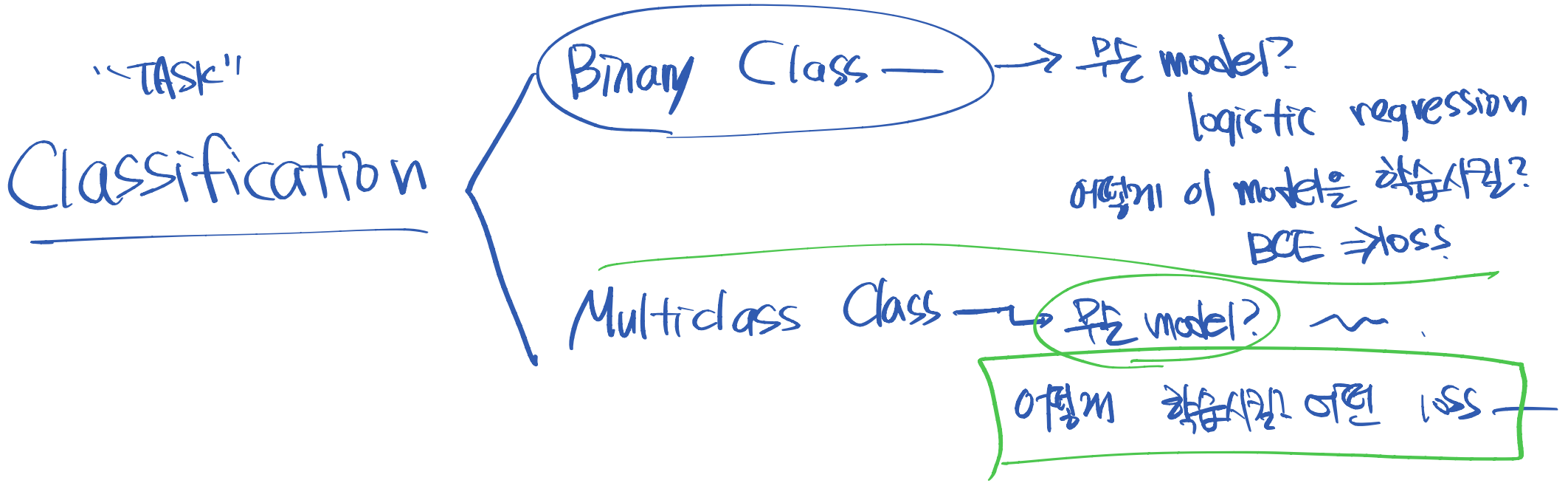
$$b \leftarrow b - \eta \frac{\partial L}{\partial b}$$

- Model: linear score + sigmoid
- Output: probability
- Loss: binary cross-entropy (BCE)
- Learning: gradient descent

Logistic Regression Training



Part 2. Multiclass Classification



Multiclass Classification

- In multiclass classification, each input belongs to one of K classes

$$y \in \{1, 2, \ldots, K\}$$

- Example
  - Handwritten digits ⟶ $0, \ldots, 9$
  - Image classification ⟶ cat / dog / bird / …
  - Disease diagnosis ⟶ Class A / B / C / …

- Can a single sigmoid output represent more than two classes?

$$p = P(y = 1|\mathbf{x})$$

$$P(y = 0|\mathbf{x}) = 1 - p$$

Multiclass Classification

- For 3 classes:

$$C_1, C_2, C_3$$

- One-vs-Rest (OvR):

$$C_1 \text{ vs. others}$$
$$C_2 \text{ vs. others}$$
$$C_3 \text{ vs. others}$$

  - Train one binary classifier for each class
  - Each classifier asks: "Is this class k?"
  - Choose the class with the highest predicted probability

- Another approach is to produce all K scores together

Multiple Scores

- In binary classification, we use one score:

$$z = \mathbf{w}^T\mathbf{x} + b$$

- In multiclass classification, we compute one score for each class:

$$z_1, z_2, \ldots, z_k$$

  - For class k:

$$z_k = \mathbf{w}_k^T\mathbf{x} + b_k$$

  - A larger z_k means that the model favors class k more strongly
  - Example
    - For three classes:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.0 \\ 0.1 \end{bmatrix}$$

Softmax Function

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

- z_k: score for class k
- p_k: predicted probability of class k

- Properties
  - $0 < p_k < 1$
  - $\sum_{k=1}^{K} p_k = 1$
- Softmax converts a set of class scores into a probability distribution

Softmax Function

- Example
  - Given scores

$$\mathbf{z} = [2.0, 1.0, 0.1]$$

  - Exponentiate

$$e^z = [e^{2.0}, e^{1.0}, e^{0.1}] \approx [7.39, 2.72, 1.11]$$

  - Normalize

$$\sum_j e^{z_j} \approx 11.22$$

$$p \approx [0.66, 0.24, 0.10]$$

$$0.66 + 0.24 + 0.10 = 1$$

Prediction

$$\hat{y} = \arg\max_k p_k$$

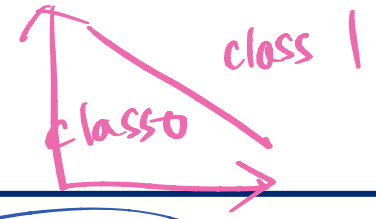
- Pipeline

$$\mathbf{x} \longrightarrow z_1, \ldots, z_K \longrightarrow \text{Softmax} \longrightarrow p_1, \ldots, p_K \longrightarrow \hat{y}$$

- Example

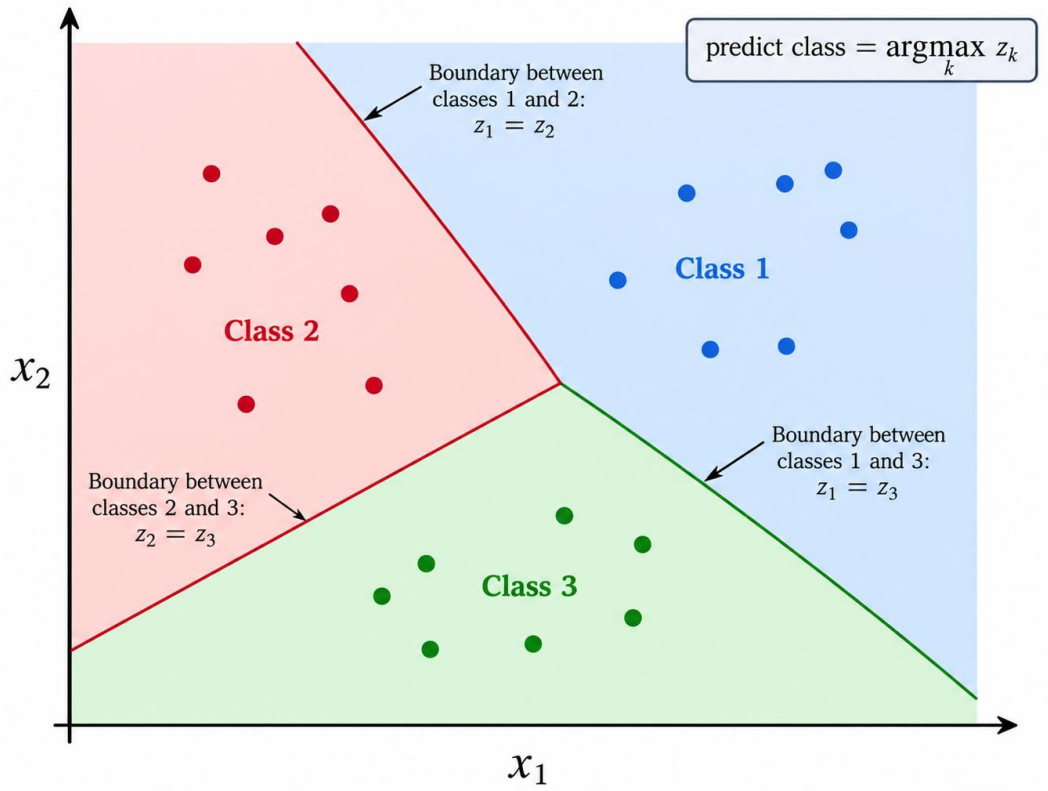
Class	Score $= z$	Probability
Cat	2.0	0.66
Dog	1.0	0.24
Bird	0.1	0.10

Decision Regions



- Each point is assigned to the class with the highest score (or probability)

$$\hat{y} = \arg\max_k z_k = \arg\max_k p_k$$



Korea Aerospace University

Learning Multiclass Classifiers

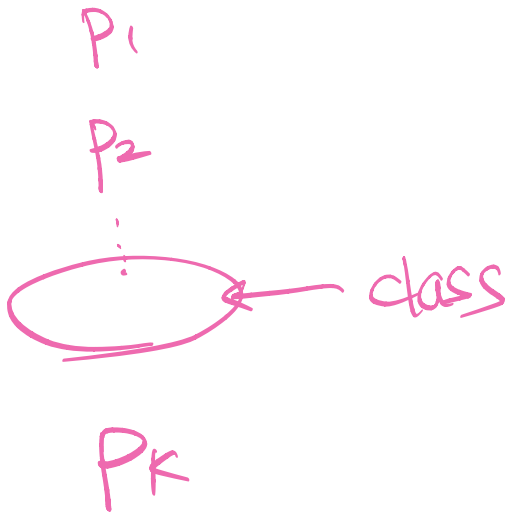
Multiclass Loss

- In binary: Binary Cross Entropy

$$L = -[y \log p + (1 - y) \log(1 - p)]$$

- In multiclass (where the true class is c): Multiclass Cross Entropy

$$L = -\log p_c$$



Multiclass Loss

- Example
  - True class = Cat

cat, dog, human

$$\mathbf{p} = [0.70, 0.20, 0.10]$$

$$L = -\log(0.70) \approx 0.357$$

- Compare
  - Correct & confident

$$p_c = 0.9 \Rightarrow L \approx 0.105$$

  - Low true-class probability

$$p_c = 0.1 \Rightarrow L \approx 2.303$$

(똑똑) (멍청)

	모델1	모델2
cat	0.9	0.2
dog	0.05	0.7
human	0.05	0.1

Loss가 작 / Loss가 커

$L_1 = -\log 0.9 < L_2 = -\log 0.2$

Cross-Entropy

$L = -\log p_c$

- One-hot label:

[1, 0, 0]

$$\mathbf{y} = [y_1, \ldots, y_K]$$

- cat
- dog
- human

개 / 고양이

[1, 0, 0]

- Multiclass cross-entropy:

$$L = -\sum_{k=1}^{K} y_k \log p_k = -\log p_c$$

where c is the true class

$= -(y_1 \log p_1 + y_2 \log p_2 + y_3 \log p_3)$

$= -(\log p_1) = -\log p_1 = -\log p_c$

Cross-Entropy

- Example
  - True class: Dog
  - Predicted probabilities

$$\mathbf{y} = [0, 1, 0]$$

$$p = [0.2, 0.7, 0.1]$$

[Cat, Dog, Bird]

  - Cross-entropy loss

$$L = -[0 \log 0.2 + 1 \log 0.7 + 0 \log 0.1]$$

$$= -\log 0.7 \approx 0.357$$

  - Only the probability of the true class contributes to the loss

Binary vs. Multiclass Classification

	Binary	Multiclass
Number of classes	2	K
Scores	z	$z_1, \ldots, z_K$
Probability Mapping	Sigmoid	Softmax
Probabilities	p $1-p$	$p_1, \ldots, p_K$
Prediction	Threshold	Argmax
Loss	Binary CE	Multiclass CE

- Binary

$$x \longrightarrow z \longrightarrow \sigma(z) \longrightarrow p \longrightarrow \hat{y}$$

- Multiclass

$$x \longrightarrow \mathbf{z} \longrightarrow \text{softmax} \longrightarrow \mathbf{p} \longrightarrow \hat{y}$$

Next Lecture

- Lecture 4: Model Evaluation & Generalization
  - Train/Validation/Test Split
  - Cross-Validation
  - Performance Metrics
  - Overfitting & Underfitting
  - Bias-Variance Tradeoff
  - Regularization
  - Generalization

Korea Aerospace University

Questions?

jiyoonshin@kau.ac.kr